

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА**

М. В. ПКУЛЯК, М. С. ДУТЧАК

**Навчальний посібник
з дисципліни «Web-дизайн»**

для студентів спеціальності
121 «Інженерія програмного забезпечення»

**Івано-Франківськ
2024 р.**

УДК 004.43

П 32

Рекомендовано до друку Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника (протокол № __ від 21 лютого 2024 р.)

Рецензенти:

Ткачук В. М. – кандидат фізико-математичних наук, доцент кафедри інформаційних технологій факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника

Казмерчук А. І. – кандидат фізико-математичних наук, доцент кафедри диференціальних рівнянь і прикладної математики факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника

Пікуляк М. В., Дутчак М.С.

П 32 Навчальний посібник з дисципліни «Web-дизайн» для студентів спеціальності 121 «Інженерія програмного забезпечення. Івано-Франківськ: ПНУ, 2024. 122 с.

У посібнику наведено основні концепції вивчення дисципліни «Web-дизайн», що дозволяє студентам освоїти методи та засоби обробки, конвертації і виведення графічної інформації за допомогою пакетів графічних програм та їх використання для розробки дизайну веб-ресурсів.

Навчальний посібник рекомендовано для студентів спеціальності 121 «Інженерія програмного забезпечення».

УДК 004.43

П 32

© Пікуляк М.В., Дутчак М.С., 2024

© Видавництво Прикарпатського національного університету імені В. Стефаника

ЗМІСТ

Вступ.	4
Лекція № 1	
Основні поняття Web-дизайну.....	5
Лекція № 2	
Основи роботи графічного редактора Photorea.com	18
Лекція № 3	
Основи роботи онлайн-сервісу Figma.com	31
Лекція № 4	
Етапи та приклади створення графічного макету сайту.....	62
Лекція № 5	
Створення веб-макету. Grid модуль в CSS.....	83
Література	122

Вступ

В багатьох джерелах та побуті вважається, що розробниками Web-сайтів є Web-дизайнери. Але не складно показати, що Web-дизайнер це є один з розробників Web-сайтів.

«Web- дизайн (від англ. Web design) згідно з Вікіпедією – галузь веб-розробки й різновидність дизайну, в задачі якого входить проектування веб-інтерфейсів користувачів для сайтів чи веб-додатків. Веб-дизайнери проектують логічну структуру веб-сторінок, знаходять найбільш зручні рішення надання інформації, а також займаються художнім оформленням веб-проекту» .

У результаті поєднання двох галузей людської діяльності гарний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і мати відповідні художні якості.

Це визначення має суттєві недоліки. По-перше, нема чіткого визначення місця Web-дизайнера у розробці та по дальшому супроводженні сайтів. Супроводження для сайтів це такий же важливий етап як і створення. Сайту без інформаційного наповнення не існує, а його дає замовник.

По-друге, логічну структуру сайту, як і його інформаційне наповнення, теж задає замовник.

По-третє, подальше існування сайту, його розвиток, його супроводження теж залежать не тільки від Веб-дизайнера.

І, четверте, сайт може мати певні вразливості (іноді вони мають назву – "дірки") і їх наявність залежить від кваліфікації Веб-дизайнера, а точніше від сумісної роботи Веб-дизайнера та Веб-програміста.

Фактично у розробці сайту приймають чи повинні приймати участь декілька спеціалістів різних фахів:

1. Замовник, який володіє інформацією, що розміщується на сайті. Який вміє структурувати інформацію, виділяти найбільш важливу, зробити її лінгвістичну перевірку чи доручити зробити цю перевірку відповідним фахівцям.

2. Web-дизайнер – треба розглядати як координатор робіт по створенню сайту. Він повинен опікуватися і про подальше існування розробленого сайту.

3. Web-програміст. Його роль збільшується в процесі розвитку Webтехнологій. Якщо на початку існування Web-сайтів його функції зосереджувались на розробці скриптів, які розширювали можливості сайтів. При технологіях, коли генерація сторінок виконується на боці сервера, функції Web-програміста при розробці Web-сайта вважаються основними.

4. Ще одним спеціалістом, який може бути залучений до роботи над Web-сайтом, є Web-художник. Часто без спеціальної освіти, та таланту художника знайти художні рішення не вдається.

5. Якщо сайт потужний, а вміст (контент) збирається з багатьох джерел, керує проектом Менеджер проекту.

Лекція 1. Основи роботи в графічному редакторі Photorea

План лекції

Вступ

1. Встановлення додатку та спільна робота Google Drive і вебверсії додатку Photorea
2. Редагування фотографії онлайн з Photorea
 - 2.1. Розпочати проект
 - 2.2. Зробіть вибір
 - 2.3. Уточнення краю
 - 2.4. Розуміння шарів
 - 2.5. Налаштування різкості зображень
3. Робота із шарами
 - 3.1. Прикрасьте ваші зображення
 - 3.2. Використання коригувальних шарів
 - 3.3. Налаштування фільтрів
4. Експорт зображень

Вступ

Термін Web-дизайн складається з двох частин. Перша частина - Web - це скорочене написання слів World Wide Web - Всесвітня Павутина. Друга частина терміна - слово "дизайн" - походить від англійського слова design, що означає: проектувати, конструювати, планувати, креслити, створювати малюнок. Таким чином, орієнтовний переклад терміна Web-design з англійської мови на українську мову може бути таким: "проектування для Інтернету".

Для уточнення сенсу цього терміна розглянемо існуючі поняття і визначення.

Дизайн в широкому сенсі слова - будь-яке проектування, тобто процес створення нових предметів, інструментів, обладнання. Дизайн у вузькому сенсі слова - це художнє конструювання. Його мета - проектування предметів, в яких форма відповідає їх призначенню.

Web-дизайн - це відносно новий жанр образотворчого мистецтва, заснований на застосуванні специфічних інструментів (програм-редакторів та онлайн-сервісів), спеціальних мов розмітки та програмування (HTML, CSS, JavaScript, Perl, PHP, Python та ін.) і мережевих технологій. Метою Web-дизайну є проектування (створення) об'єктів (Web-сторінок, банерів, аплетів, скриптів, окремих елементів Web-сторінок), що розміщуються в Інтернеті.

1. Встановлення додатку та спільна робота Google Диска і вебверсії додатку Photorea

Photorea — розширений редактор зображень, який може працювати як з растровою, так і з векторною графікою, це безкоштовний аналог графічного редактора Adobe Photoshop. Ви можете використовувати його для простих завдань, таких як зміна розміру зображень, а також для складних завдань, таких як дизайн веб-сторінок, створення ілюстрацій, обробка фотографій тощо.

ГР Photorea дає можливість створювати і працювати із графічними файлами які безпосередньо знаходяться на вашому Google диску. Для цього потрібно, щоб додаток Photorea був підключений до вашого Google диску.

Якщо він уже підключений, то створити новий документ можна клікнувши на Photorea у переліку додатків, який можна подивитися клікнувши на Створити -> Більше.

Якщо його немає у переліку, то Створити -> Більше -> Підключити більше додатків і у полі пошуку ввести Photorea, далі Встановити, після чого він має з'явитися у переліку додатків.

Запустіть Photorea і клікніть на кнопку "Профіль" далі "Увійти" -> "Google" -> "Ваш обліковий запис Google" для того, щоб зв'язати Photorea із вашим обліковим записом Google. Червоний колір кнопки "Профіль" вказує на відсутність входу, а оранжевий підтверджує вхід в обліковий запис Google.

Закрийте вкладку із додатком Photorea і повторно клікніть на Створити -> Більше -> Photorea. Після чого має з'явитися спливаюче вікно "Виберіть обліковий запис, щоб перейти в додаток Photorea", обираєте свій обліковий запис і клікаєте "Дозволити".

Якщо спливаюче вікно не з'явилося, однією із причин може бути налаштування браузера на заборону спливаючих вікон. Для того, щоб її зняти, відкрийте піктограму, яка з'являється в кінці адресного рядка і дозвольте браузеру демонструвати спливаючі вікна або знайдіть встановлення цієї заборони у налаштуваннях браузера.

Для того, щоб перевірити чи створений або відкритий з допомогою додатку Photorea документ зв'язаний з Вашим Google диском і на ньому зберігатимуться зміни, у меню Файл має бути доступний пункт Зберегти (Google Drive). Якщо він не доступний, то збережіть файл з розширенням PSD на ПК, потім завантажте на Google Drive у потрібну папку і відкрийте з допомогою додатку Photorea.

***Увага!** Такий спосіб можете використати для збереження файлів, які не стосуються виконання лабораторних робіт. Для виконання лабораторних робіт файли обов'язково мають створюватися і редагуватися безпосередньо на Google Drive. Якщо у вас з цим виникли труднощі, зверніться до викладача про допомогу, оскільки папки кожної із робіт потрібно створювати безпосередньо на Google Drive у папці дисципліни і виконувати у них всі завдання відповідних робіт, а не шляхом завантаження папок і файлів з локального ПК.*

ФАЙЛИ, ЗАВАНТАЖЕНІ ІЗ ПК ПЕРЕВІРЯТИСЯ НЕ БУДУТЬ!!!

Зверніть увагу, що:

➤ якщо створювати документи через Створити -> Більше -> Photorea, додаток Photorea створює документи не у відкритій папці Google Drive, а в його кореневій папці. Для перенесення файлів у папку відповідної лабораторної роботи, потрібно клікнути на нього правою кнопкою миші, далі "Перемістити в" і вкажете папку куди перенести.

➤ Для того, щоб файл створювався у потрібній папці Вашого Google Drive, його потрібно створювати через Файл -> Створити, а при зберіганні обрати Зберегти більше - PSD to Storage - Google Drive і обрати потрібну папку.

➤ Для відкриття файлу із Google Drive обираєте Файл -> Відкрити більше -> Storage -> Google Drive.

Якщо це виконувати через меню Файл -> Відкрити чи Файл -> Зберегти, то файли відкриватимуться (зберігатимуться) з (на) локального (-му) ПК, так ніби Ви працюєте із десктопну версією програми.

Десктопную версією програми можна встановити на ПК, клікнувши на Більше - Встановити Photopea у веб-версії додатку Photopea (для виконання лабораторних робіт десктопную версією програми не використовуйте).

Якщо Ви маєте деякі навички роботи в Adobe Photoshop, і звикли до його комбінацій клавіш, то дію їх у веб версії потрібно досліджувати, оскільки при роботі у веб-версії Photopea деякі комбінації клавіш відрізняються (а для деяких взагалі немає працюючих аналогів) і якщо Ви будете викликати по звичці комбінації клавіш Adobe Photoshop, то деякі з них будуть виконані браузером, а не Photopea, а в локальній версії вони працюють так як в Adobe Photoshop.

Наприклад, Ctrl+T (вільна трансформація) працює однаково у встановлених версіях Adobe Photoshop і Photopea, а для веб-версії її аналогом є Alt+Ctrl+T. Аналогом Ctrl+N є Alt+Ctrl+N.

Shift+Ctrl+N - створення нового шару, у веб-версії ця комбінація також вказана, але по факту відкриває нове анонімне вікно.

Тому цю особливість веб-версії потрібно досліджувати і звикати до неї.

Комбінації клавіш для роботи у Photopea (як і в Adobe Photoshop) вказані біля назв відповідних пунктів меню та при наведенні мишкою відображаються на всіх інструментах.

Довідник по комбінації клавіш наведений у самій програмі:

Більше - Поєднання клавіш

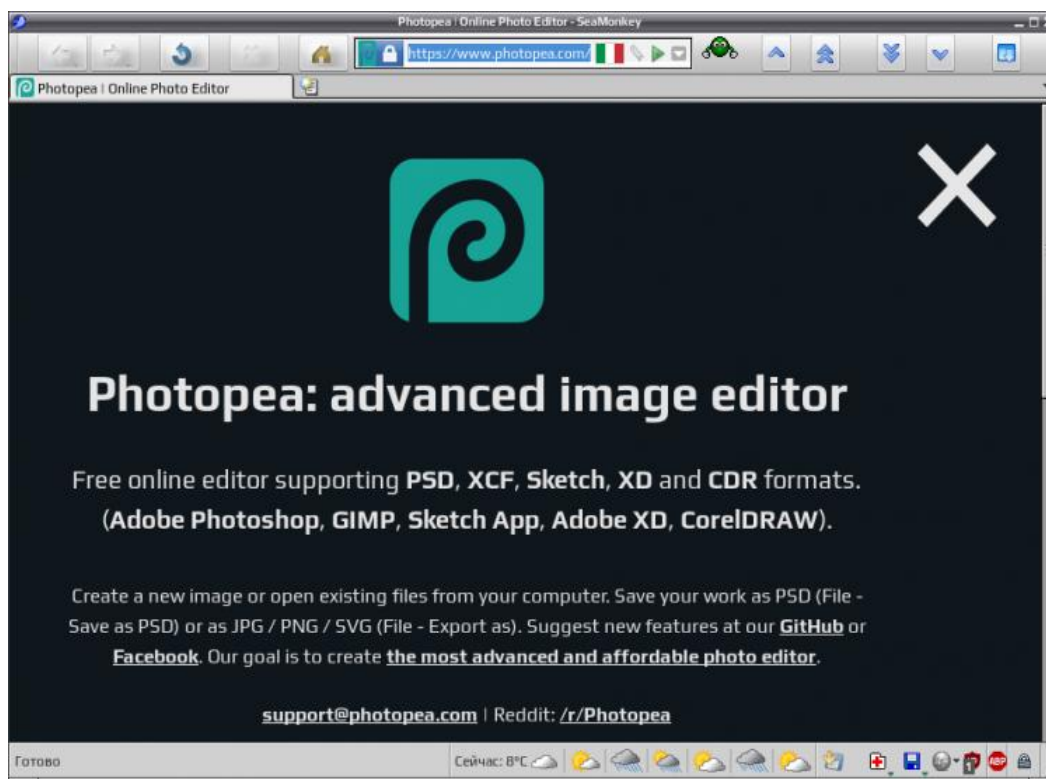
Файл		Вільна трансформація	Alt+Ctrl + T	Об'єднати з нижче	Ctrl + E
Відкрити	Ctrl + O	Налаштування	Ctrl + K	Виділити	
Зберегти	Ctrl + S	Коригування		Усе	Ctrl + A
Зберегти як PSD	Shift+Ctrl + S	Рівні	Ctrl + L	Скасувати виділення	Ctrl + D
Експортувати як	Alt+Shift+Ctrl + S	Криві	Ctrl + M	Інвертувати	Shift+Ctrl + I
Редагувати		Відтінок/Насиченість	Ctrl + U	Перегляд	
Крок уперед	Shift+Ctrl + Z	Інвертувати	Ctrl + I	Збільшити масштаб	Ctrl + +
Крок назад	Ctrl + Z	Шар		Зменшити масштаб	Ctrl + -
Вирізати	Ctrl + X	Новий шар	Shift+Ctrl + N	Лінійки	Ctrl + R
Копіювати	Ctrl + C	Скопіювати на новий шар	Ctrl + J	Напрямні	Ctrl + ;
Вставити	Ctrl + V	Маска відсікання	Alt+Ctrl + G	Сітка	Ctrl + '
Очистити	Delete	Групувати шари	Ctrl + G	поєднання клавіш	?
Заповнити	Alt + Backspace				
Navigation					
Vertical scroll	Wheel	Horizontal scroll	Ctrl + Wheel	Zooming	Alt + Wheel
Quick tools (press to enable, release to disable)					
Переміщення	Ctrl	Рука	Space	Лула	Ctrl + Space

Tools					
Переміщення	V	Латка	J	Прямокутник	U
Artboard Tool	V	Червоні очі	J	Еліпс	U
Вибір прямокутником	M	Пензель	B	Лінія	U
Вибір еліпсом	M	Олівець	B	Параметрична форма	U
Вибір ласо	L	Заміна кольору	B	Довільна фігура	U
Виділення багатокутним ласо	L	Інструмент клонування	S	Рука	H
Вибір магнітним ласо	L	Гумка	E	Лула	Z
Вибір об'єкта	W	Фонова гумка	E		
Швидке виділення	W	Гرادієнт	G	Інше	
Чарівна Паличка	W	Заливка	G	Стандартно: Білий та Чорний	D
Кадр	C	Освітлювач	O	Обміняти Кольори	X
Кадрування перспективи	C	Випалювання	O	Режим швидкої маски	Q
Фрагмент	C	Палець	O	Decrease Brush Size	[
Виділення фрагмента	C	Текст	T	Increase Brush Size]
Піпетка	I	Ручка	P	Decrease Hardness	{
Лінійка	I	Вільна ручка	P	Increase Hardness	}
Точковий пензель відновлення	J	Виділення Контуру	A		
Пензель відновлення	J	Виділення Вузла	A		

Корисні посилання <https://www.photopea.com/learn/>

<https://helpx.adobe.com/ua/photoshop/user-guide.html>

Photopea. Додатковий матеріал



Photopea - онлайн-редактор зображень, призначений для роботи з растровою і векторною графікою.

Photopea - це робота Івана Куцкіра, який народився в Україні, але тепер живе в Чехії. Його веб-сайт дає уявлення про те, чим він займається: алгоритми, трасування променів, 3D-моделювання та інші завдання програмування.

Онлайн-редактор Photopea оптимізований для комфортної роботи на будь-якому пристрої (*комп'ютер, ноутбук, планшет або телефон*), в процесі редагування підключення до Інтернет не потрібно (*автономний режим*).

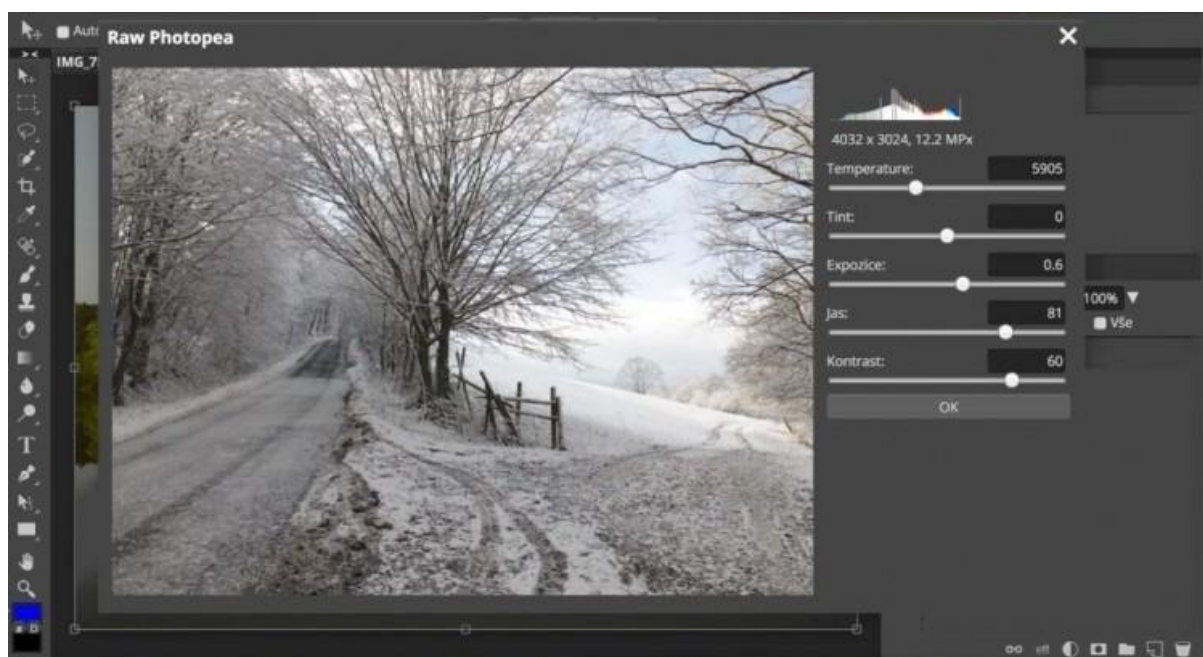
Для онлайн-сервісу Photopea є неофіційний Electron клієнт, який працює аналогічно браузерної версії. Після завантаження клієнт може працювати в автономному режимі, без активного підключення до Інтернет, після закінчення кешування (*кеш знаходиться в каталозі ~/.config/Photopea*).

Інтерфейс сервісу і клієнта виконаний по аналогії з інтерфейсом Adobe Photoshop (*комерційний растровий графічний редактор*), наявний функціонал дозволяє вирішувати прості (*обрізка зображень, зміна розміру і ін*) і складні завдання (*створення веб-сторінок, ілюстрацій, редагування фото тощо*).

Photopea завдяки "Меню швидкого старту" дозволяє швидко створити новий проект (*порожній або за шаблоном*), відкрити збережений (*локально або в мережі*), можна одночасно відкрити кілька проектів, в вкладках.

Основні інструменти знаходяться в невеликій панелі зліва робочої області Photopea, на ній розташовано все необхідне що може знадобиться для редагування зображень. Редактор пропонує велику кількість візуальних ефектів, що застосовуються до окремих зображень або до всього проекту.

Photoreo підтримує редагування шарів (*редагувати шари можна без зміни фактичного вмісту*) і використання фільтрів, пропонує велику кількість шрифтів (*як найбільш популярних, так і екзотичних*), є велика колекція кистей і передбачена можливість імпорту власних.

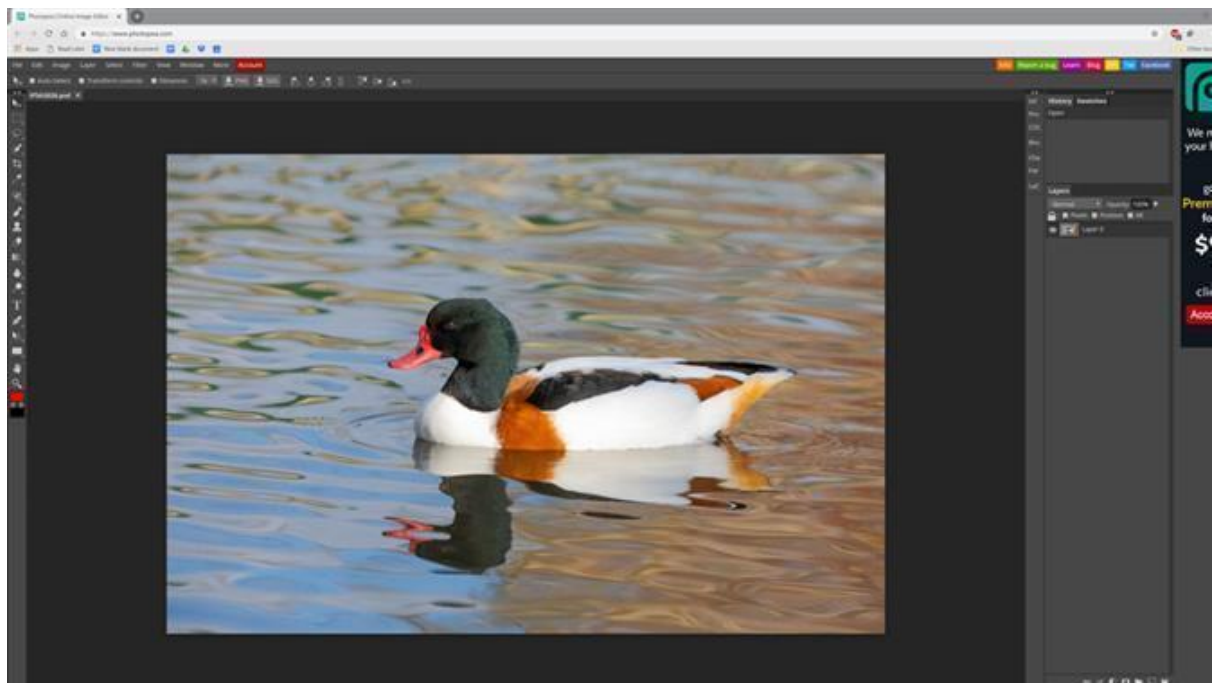


Photoreo пропонує великий набір інструментів для роботи з векторною графікою, підтримується робота (*імпорт і експорт*) з більшістю популярних графічних форматів (*векторних і растрових*), для управління може використовуватися миша і / або клавіатурні поєднання.

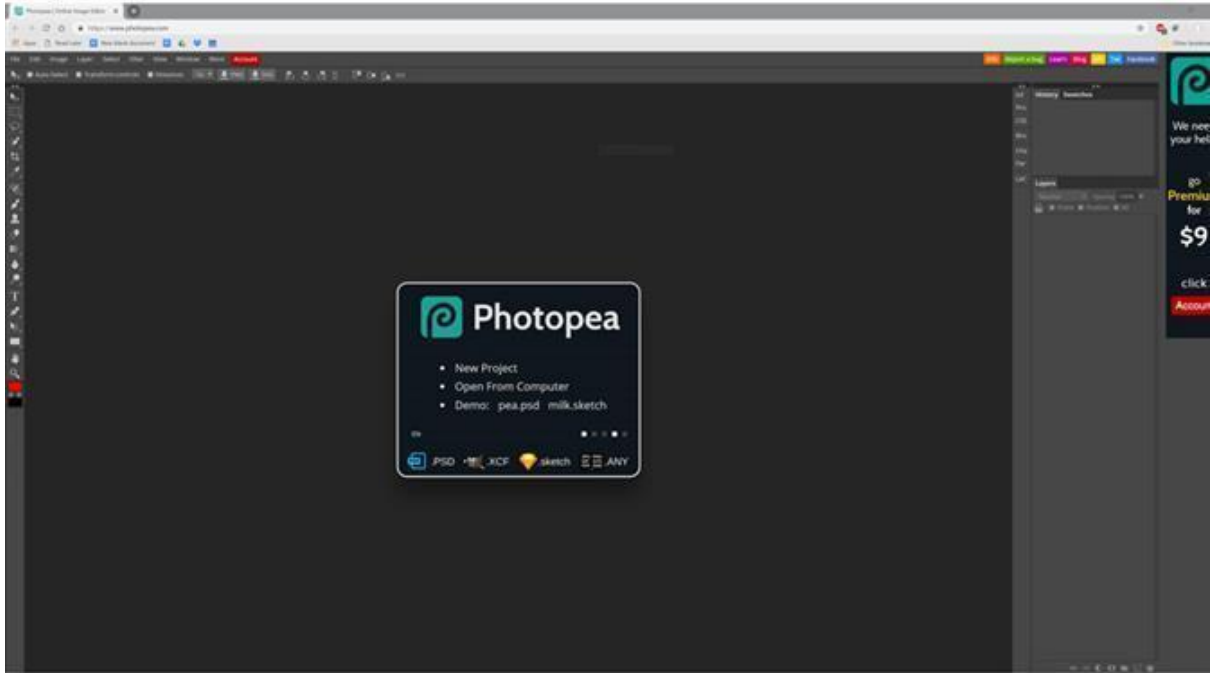


Photopea безкоштовний веб-сервіс (*FREE*), але є і платні послуги, які не впливають на основний функціонал редактора а лише розширюють другорядні функції (наприклад відсутність реклами).

2. РЕДАГУВАННЯ ФОТОГРАФІЇ ОНЛАЙН З РНОТОРЕА

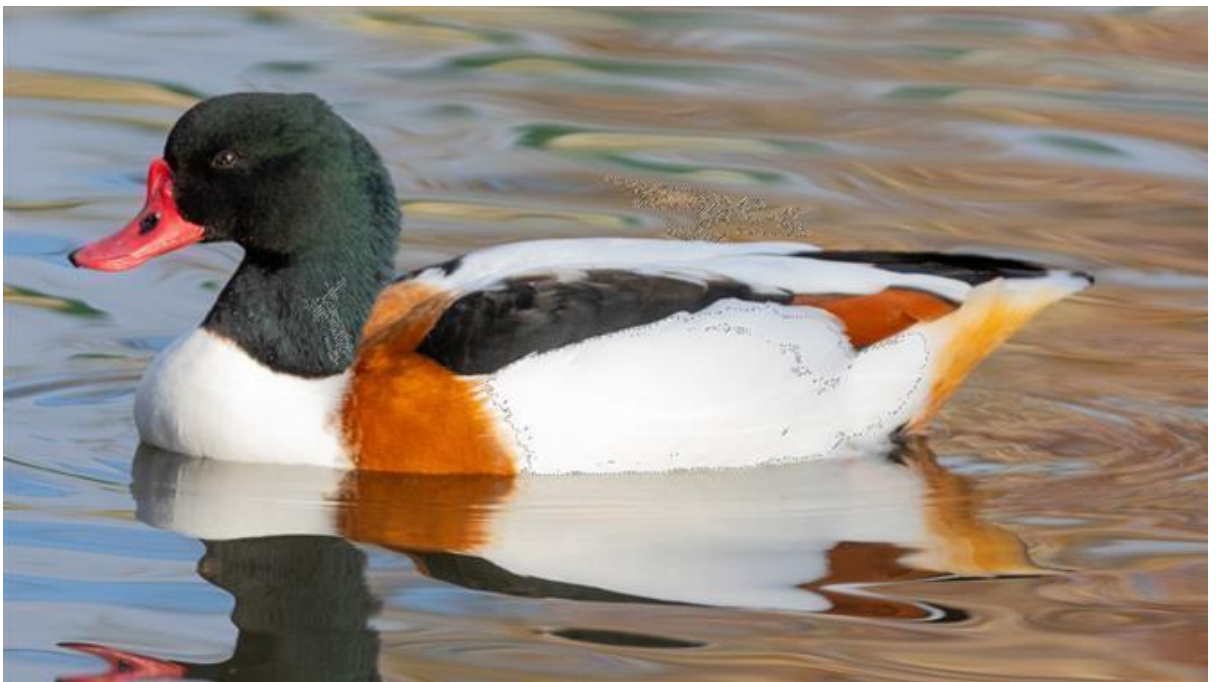


КРОК ЗА КРОКОМ: РЕДАГУЙТЕ ЗОБРАЖЕННЯ ОНЛАЙН З РНОТОРЕА



2.1. Розпочати проект

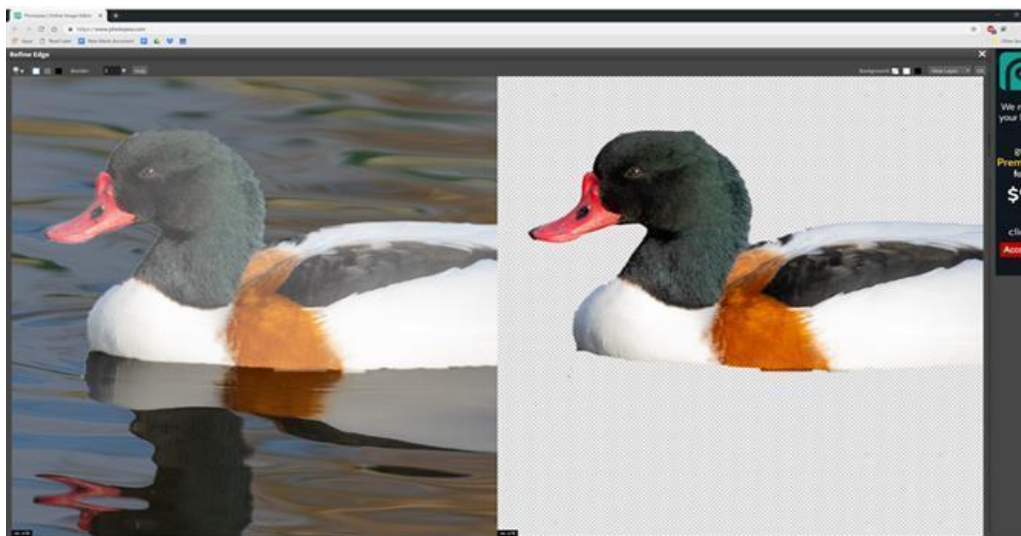
Ви можете почати новий проект в Photopea або завантажити файл зображення з вашого ПК. Додаток сумісний з PSD в Photoshop, а також з файлами JPEG, PNG і іншими форматами графічних файлів. Натисніть «Відкрити з комп'ютера» для роботи з уже наявним файлом або з новим проектом.



2.2. Зробіть вибір

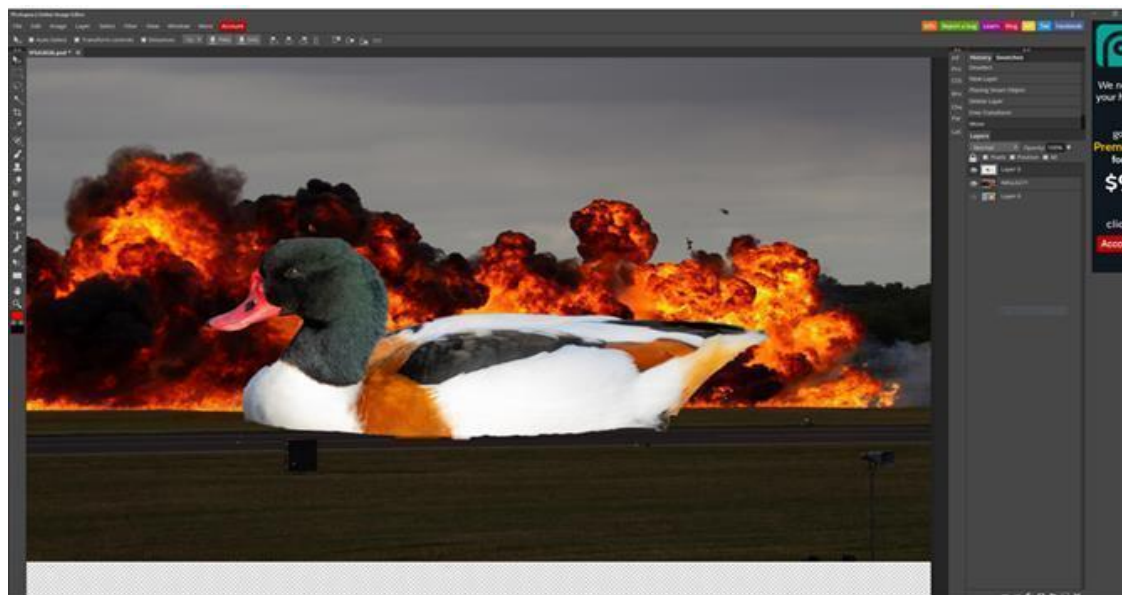
Можливість працювати з однією частиною зображення, не впливаючи на інше, є потужною функцією, і Photopea дозволяє робити вибірки за допомогою різних інструментів,

включаючи чарівну паличку, яка вибирає безперервні області кольору, кисть швидкого вибору, яка дотримується краю, поки ви малюєте виділення, плюс ласо від руки і прямокутні інструменти виділення.



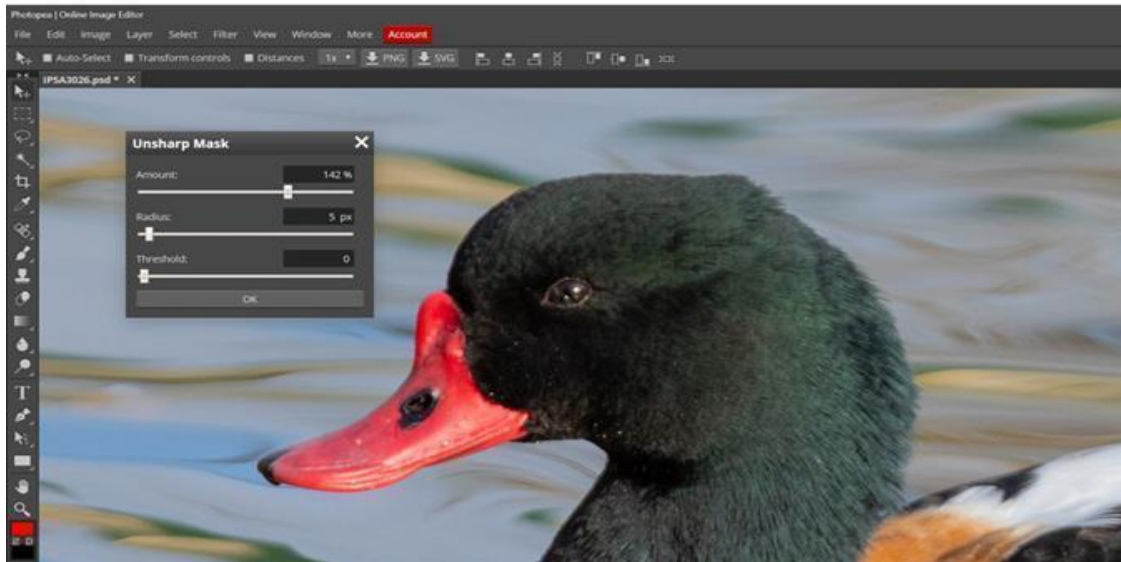
2.3. Уточнення краю

Зробивши вибір, натисніть кнопку «Уточнити край». Відкриється вікно, в якому з одного боку буде показаний ваш оригінал, а з іншого - ваш вибір. Кисті по краях обраного об'єкта (зліва) змінює контур виділення (праворуч). Ви можете змінити розмір кисті і вибрати, малювати білим (додати), чорним (видалити) або сірим (пом'якшити).



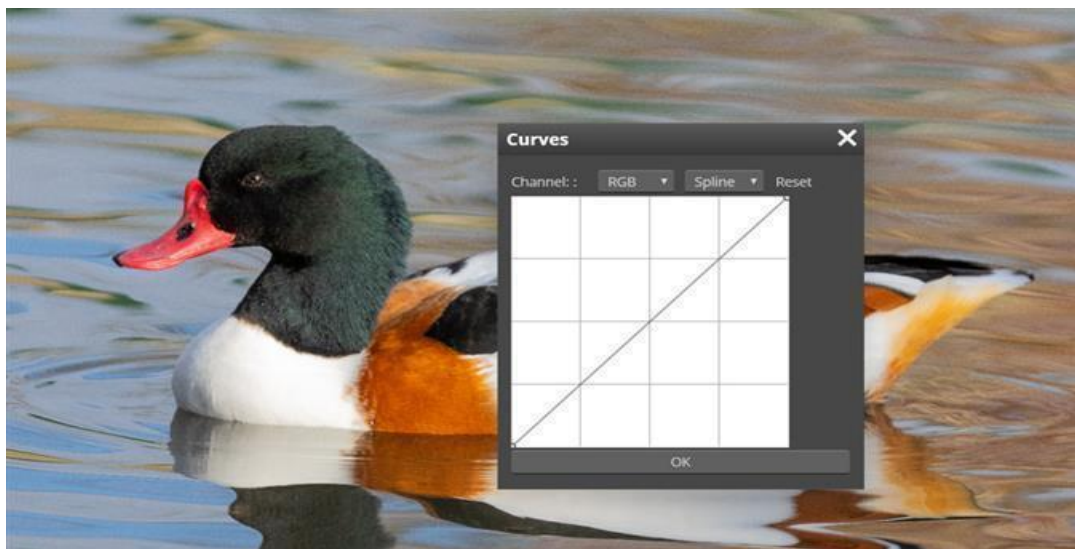
2.4. Розуміння шарів

Після уточнення виділена область копіюється і відображається у вигляді нового шару з прозорим фоном в палітрі шарів. Шари дозволяють вам робити трюки, такі як заміна фону і зміна кольору частин зображення. Щоб вставити новий фон за об'єктом вирізу, використовуйте «Файл» Відкрити і розмістити», щоб відкрити нове зображення і помістити його на свій власний шар.



2.5. Налаштування різкості зображень

Підвищення різкості виконується фільтрами з такими іменами, як «Посилити різкість», але для більшого контролю використовуйте «Фільтр» Посилити різкість > Розумна різкість», який дозволяє контролювати ступінь підвищення різкості за допомогою повзунків, серед інших параметрів.

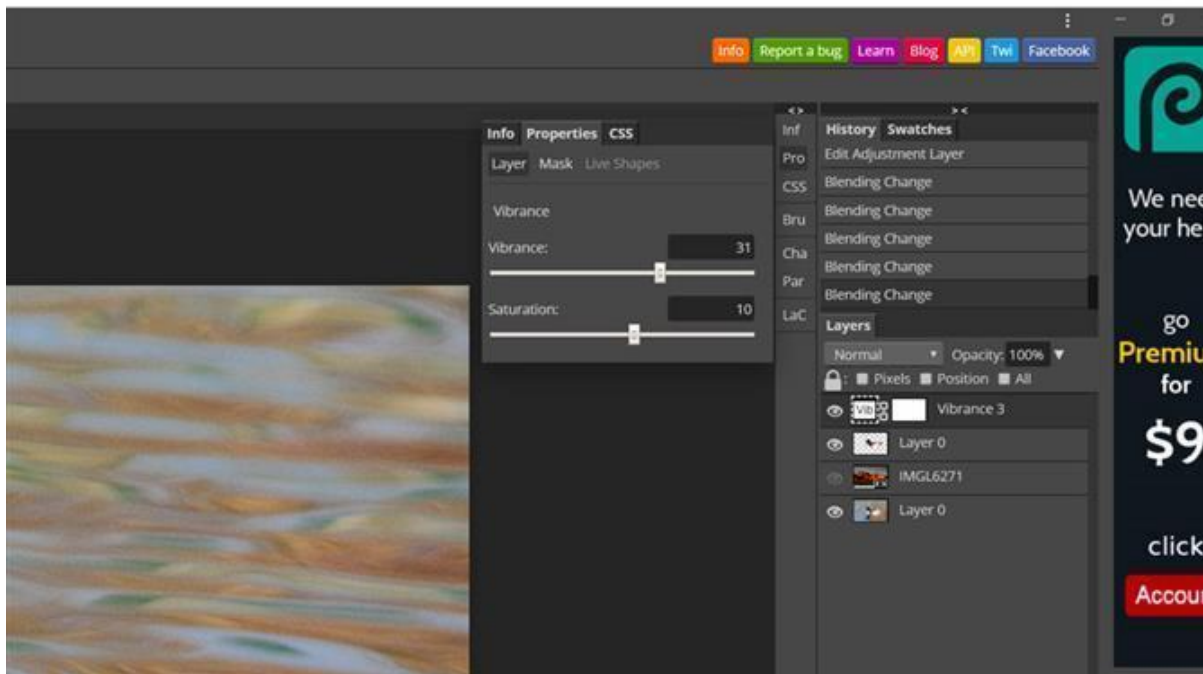


3. Робота із шарами

Уявіть, що ваше зображення розділене на шматочки, які складені так, що ви бачите верхній. Ви також можете побачити тих, хто знаходиться під ним, якщо є діра, або якщо вони налаштовані на змішування.

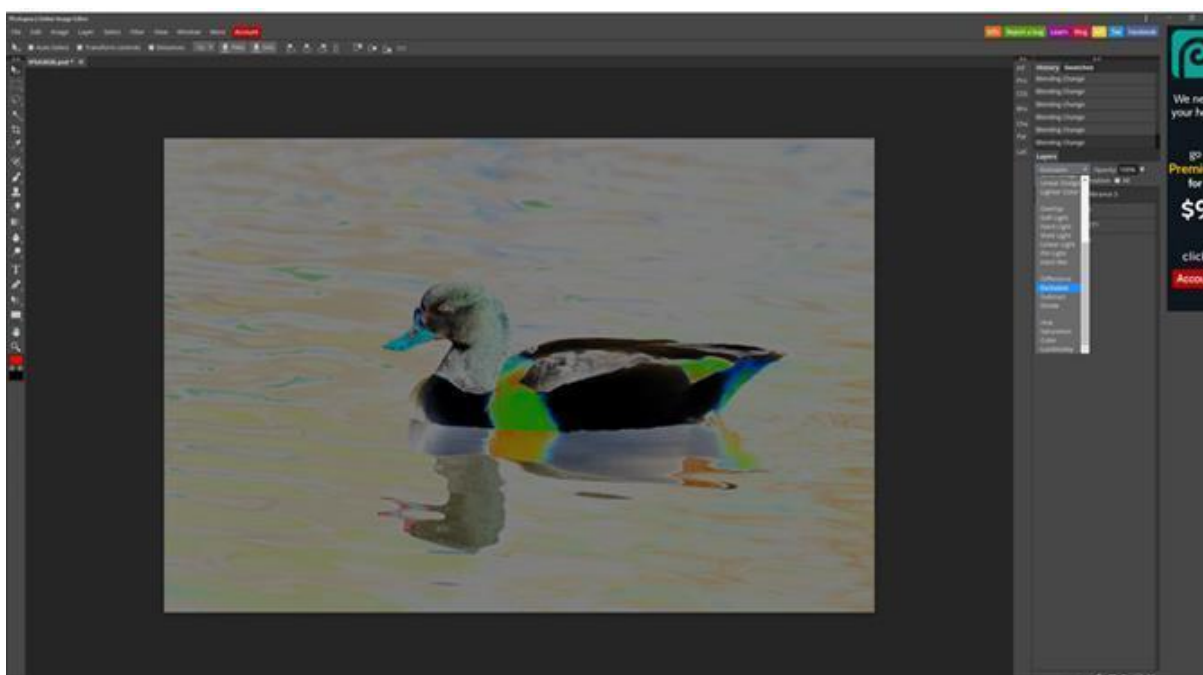
3.1. Прикрасьте ваші зображення

Яскравість вашого зображення обробляється декількома інструментами в «Image» Adjustments» («Зображення» Коригування»), але кращим з них є рівні. Це дозволяє вам змінювати загальну яскравість вашого зображення за допомогою центрального (сірого) повзунка під гистограмою, а також переміщати білі і чорні точки для збільшення контрастності, щоб побачити, що відбувається. Завжди є можливість скасування будь-яких змін, які були зроблені.



3.2. Використання коригувальних шарів

Коригувальні шари змінюють колір і яскравість шарів під ними. Щоб додати один, використовуйте кнопку в нижній частині палітри шарів і виберіть у спливаючому меню. Ви можете перетягнути коригувальні шари вгору і вниз по стеку шарів, щоб змінити ефект.

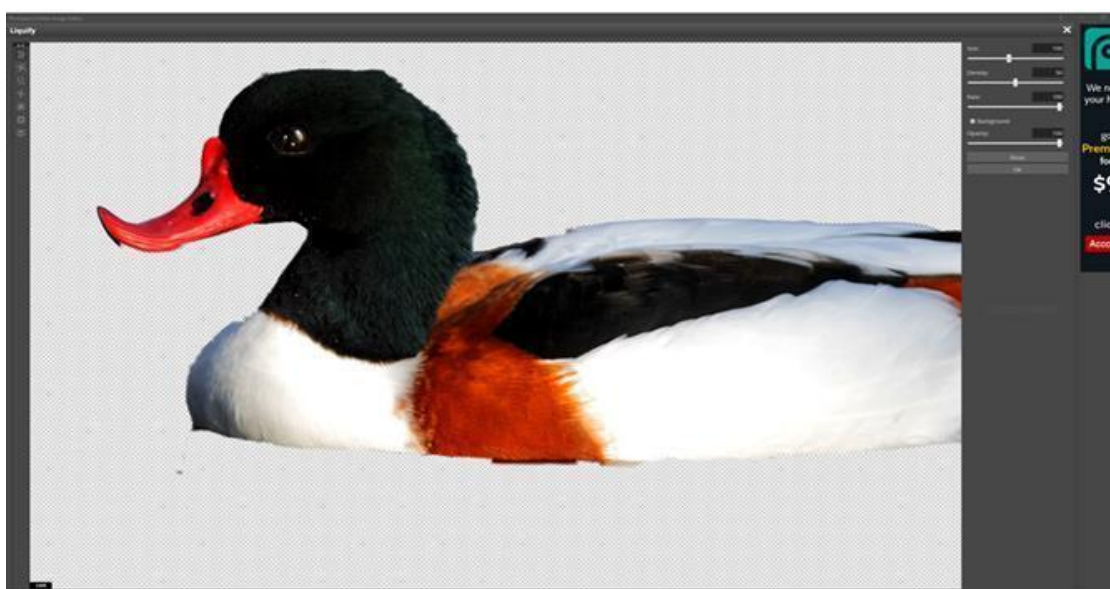


3.3. Налаштування фільтрів

Фільтр - ефект, який застосовується або до цілого зображення, або тільки всередині виділення. Ви відразу побачите ефект фільтра, але деякі поставляються з вікном опцій, щоб спочатку налаштувати їх параметри.

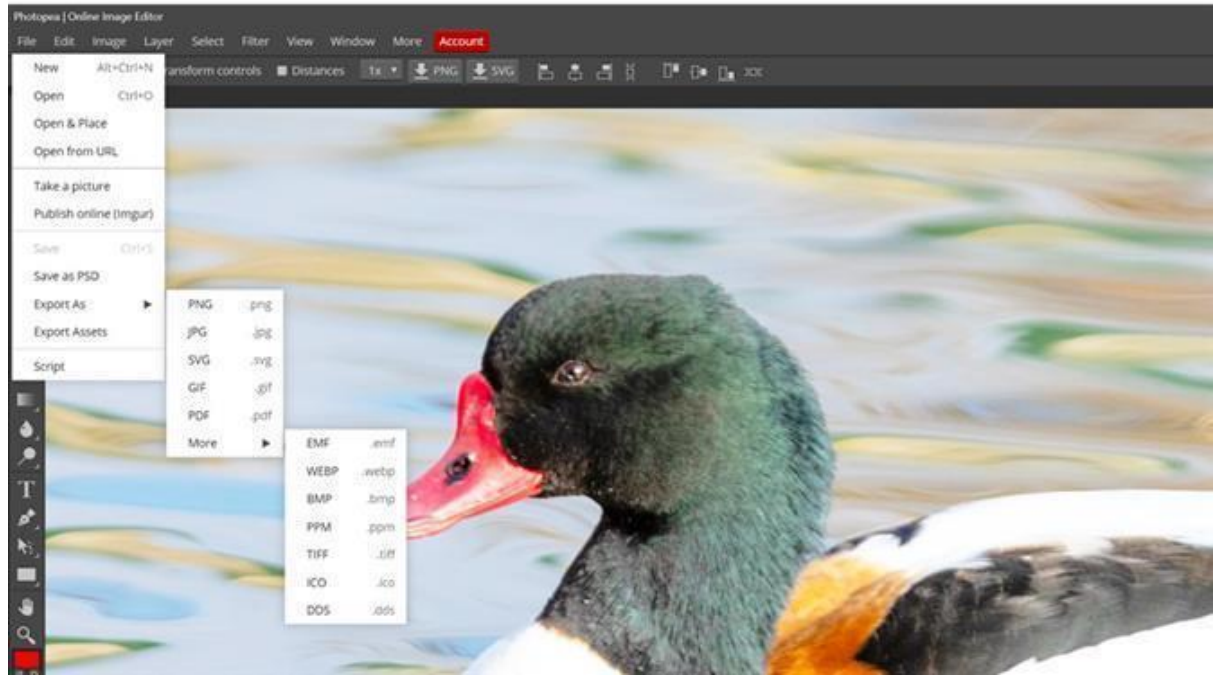
Режими накладення

Змішування змінює спосіб відображення шарів нижче шару, над яким ви працюєте. За замовчуванням використовується значення «100%», що не дозволяє нічого пропустити. Змініть це на Множення, використовуючи спадне меню у верхній частині палітри Шарів, і ви побачите потемніння квітів у міру їх злиття. Екран має протилежний ефект, в той час як деякі з тих, хто знаходиться ближче до нижньої частини списку, можуть мати дуже дивні ефекти.



Більше фільтрів

Якщо вам потрібно розмити, спотворити або іншим чином зіпсувати зображення, в меню «Фільтр» є варіанти. Фільтри застосовуються до всього зображення, якщо тільки у вас не обрана область, коли вони впливають тільки на внутрішню частину контуру виділення. Фільтр «Liquidify» дозволяє розтягувати, формувати і іншим чином спотворювати зображення. Параметр «Останній фільтр» повторно застосовує останній використаний фільтр і його настройки.



4. Експорт зображень

Існують кнопки для експорту обраного в даний момент шару у вигляді PNG або SVG. Для експорту всього зображення перейдіть в меню «Файл». Звідси ви можете зберегти свою роботу у вигляді PSD-файлу, цілих верств або експортувати в форматі JPEG, PNG або багатьох інших популярних форматів. Однак при цьому ваші шари будуть зведені в один, тому переконайтеся, що ви закінчили, перш ніж вибрати цей параметр.

Перетворення Figma в PSD

Figma - популярний інструмент для створення інтерфейсів користувача. Але його неможливо експортувати у формат PSD, і Adobe Photoshop не може відкрити файли FIG. Але ви можете зробити це автоматично, як описано нижче.

FIG до PSD

На даний момент Photopea є найкращим інструментом для перетворення документів Figma FIG у файли PSD Photoshop.

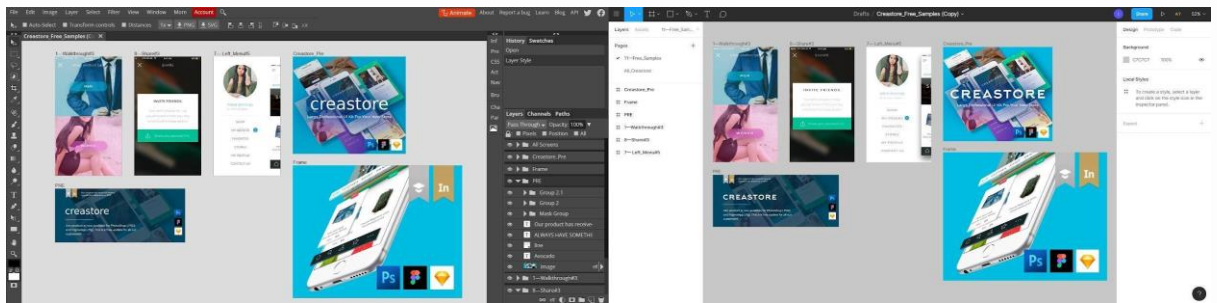
Завантажте файл Figma зі свого пристрою: натисніть **Файл - Відкрити** та знайдіть та виберіть файл Figma (або скористайтеся перетягуванням).

Вміст файлу Figma (усі сторінки та монтажні області) має бути перед вами. Ви можете натиснути **Файл - Зберегти як PSD** і отримати PSD -версію свого файлу :)

Відсутні шрифти

Якщо ви використовуєте будь-які спеціальні шрифти у вашому файлі Figma, завантажте їх у Photorea за допомогою File - Open (і виберіть файли TTF або OTF). Photorea використовуватиме ці шрифти для перемалювання відповідних текстових шарів.

Приклад того самого файлу Figma у Photorea та Figma: структура шару збережена.



Лекція 2. Основні поняття Web-дизайну

Цілі: ознайомити студентів з основними поняттями та завданнями web-дизайну, загальними поняття комп'ютерної графіки та форматами графічних файлів для Web.

План лекції.

1. Поняття web-дизайну
2. Загальні поняття комп'ютерної графіки
3. Формати графічних файлів для Веб

1. Поняття web-дизайну

Розробка серйозного комерційного проекту вимагає значних витрат часу, вкладення коштів і застосування кваліфікованої праці. Найчастіше складні Web-сайти розробляються колективно, так як доводиться вирішувати безліч складних технічних, образотворчих, інформаційних, психологічних та інших завдань.

Основним завданням професійних Web-дизайнерів є залучення якнайбільшої кількості відвідувачів на створений ними сайт. Цьому сприяє нерозривна єдність художнього оформлення сайту і його предметного змісту (єдність форми і змісту). Не останню роль у набутті популярності відіграє вміння "розкрутити" сайт.

Для створення комерційного сайту необхідна професійна майстерність (знання теорії, володіння навичками), вміння виділяти актуальні проблеми і хороший художній смак. При розробці сайту дизайнери використовують закономірності, накопичені в живописі, графіці, скульптурі, архітектурі, художній фотографії, декоративно-прикладному мистецтві, графіті. Однак багато прийомів Web-дизайну є принципово новими. Тому часом говорять про Web-дизайн як про новий жанр мистецтва, який спирається на свої образотворчі закони. Ці закони в даний час створюються нашими сучасниками і проходять практичну апробацію в мережі.

Образотворчими засобами дизайну є: точка, лінія, фактура, текстура, колір, форма, пропорція і ін.

На Web-сторінках присутні не тільки текст і статичні зображення, але і анімація, відео, звуки. Зовнішній вигляд Web-сторінок динамічно змінюється при виконанні користувачем певних дій. Це робить Web-сторінку складним твором мистецтва, створюваним на стику літератури, журналістики, образотворчого мистецтва, кіно, телебачення, радіо, фотомистецтва та IT-технологій. При цьому досвідчений професійний розробник добре уявляє контингент людей, які частіше за інших відвідують його сайти, враховує психологію користувачів, підбирає теми, які найбільш популярні в мережі в даний момент часу. Переважна більшість Web-сторінок є продуктом мультимедіа.

Проектування сторінок або сайтів ускладнюється ще й тією обставиною, що одна і та ж сторінка відображається по-різному різними браузерами (Internet Explorer, Google Chrome, Opera і т.д.) і у вікнах різного розміру. У багатьох випадках браузер самостійно "вирішує" як буде виглядати завантажена сторінка. Змінити макет сторінки може і користувач, наприклад, шляхом відключення опції завантаження малюнків, анімації.

Для створення динамічних Web-сторінок (що володіють елементами штучного інтелекту) завантаження окремих елементів проводиться в залежності від часу доби або за допомогою генератора випадкових чисел. Таким чином, програмується зміна музичних кліпів, малюнків, завантажених анекдотів і афоризмів тощо.

Перелічимо деякі важливі рекомендації щодо створення Web-сайту.

- Доцільно не перевантажувати сторінку великим числом деталей. Це пов'язано з тим, що людина за короткий час здатна сприйняти і запам'ятати не більше 7-8 об'єктів.
- Вважається, що чим менший розмір об'єкта, тим більшою повинна бути його контрастність. Помічено, що чітко виражений фоновий малюнок на Web-сторінці підвищує стомлюваність очей і знижує ефективність сприйняття тексту. Для

виключення строкатості сторінку рекомендується створювати з використанням не більше чотирьох основних кольорів.

- На всіх сторінках одного сайту рекомендується використовувати однакові панелі навігації, розміщені в однакових місцях. Кожна Web-сторінка сайту повинна мати посилання на головну сторінку сайту.

- Користувачі, які “подорожують” по мережі, “проскакують” повз ті Web-сторінки, які неприпустимо довго завантажуються на їх комп'ютери. З цієї причини при проектуванні сайту дизайнерові необхідно вміти заздалегідь оцінити приблизний час завантаження і допустимий обсяг створеної ними сторінки.

- На останній стадії проектування сайту доцільно перевірити працездатність всіх зроблених гіперпосилань. Завершуючи проектування сайту, бажано переглянути його за допомогою різних браузерів.

Наведемо основні визначення у web-дизайні

Web-дизайнер - це фахівець, який займається проектуванням і розробкою web-вузла; створенням його структури, підбором кольорової палітри і розробкою користувальницького інтерфейсу.

Web-майстер - це фахівець, основним завданням якого є підтримка існуючого web-сайту. Підтримка полягає в інформаційному наповненні сторінок сайту і у віртуальній взаємодії з користувачами Інтернету, які відвідують цей сайт.

Web-сайт - це сукупність web-сторінок, об'єднаних між собою єдиною композицією, темою інформаційного змісту, структурою, набором кольорів, і знаходяться на одному або декількох серверах.

Web-сторінка - інформаційний ресурс, доступний в мережі World Wide Web (Всесвітня павутина), який можна переглянути у Web-браузері. Зазвичай, інформація Web-сторінки записана у форматі HTML, XHTML, або рідше Wml (для wap-сторінок).

Web-браузер - це прикладна програма, яка дозволяє подавати запити, отримувати з сервера і відображати web-сторінки.

Статичний web-вузол - це web-сайт, вміст якого фіксований і не змінюється до тих пір, поки web-майстер не замінить дані на сайті.

Динамічний web-вузол - це web-сайт, в якому інформація зберігається в базі даних, а web-сторінки генеруються динамічно у відповідь на запит клієнта.

Сервер - це пристрій, підключений до мережі, на якому встановлене серверне програмне забезпечення і який призначений для управління ресурсами. Якщо він підключений до глобальної мережі інтернету, то на ньому можуть бути розміщені web-документи, які передаються у відповідь на запити клієнтів.

Мережевий робот - це комп'ютерна програма, яка сканує різні web-сайти, витягує ключові слова і зберігає їх в базі даних пошукового сервера.

Пошуковий сервер - це програма, яка організовує пошук за ключовими словами в глобальній мережі Інтернет або на конкретному web-вузлі. У відповідь на запит користувача пошуковий сервер повертає список посилань на документи.

Доменне ім'я - це назва web-сайту, представлена, як правило, у вигляді символів латинського алфавіту.

Система доменних імен (англ. Domain Name System, DNS) — ієрархічна розподілена система перетворення імені хоста (комп'ютера або іншого мережевого пристрою) в IP-адресу.

Протокол Інтернет (IP) - це протокол передачі інформаційних пакетів, який лежить в основі функціонування глобальної мережі Інтернет.

Web-палітра - це набір веб-надійних кольорів. Web-орієнтована палітра гарантує, що кольорове зображення буде відображатися в незмінному вигляді більшістю браузерів, що працюють на різних платформах.

Протокол - це набір правил, що регламентують обмін інформацією між комп'ютерами в мережі. Як приклади мережевих протоколів можна привести IP, HTTP і FTP.

Гіпертекст - це інформаційна структура, яка забезпечує навігацію між web-документами за допомогою гіпертекстових посилань. Фрагменти тексту або зображення, при активації яких відображаються пов'язані з ними документи, називаються гіпертекстовими посиланнями.

2. Загальні поняття комп'ютерної графіки

Будь-яке графічне зображення можна представити як композицію різнобарвних ділянок, тому, однією з основних відомостей про картинку є інформація про колір.

Колір — це властивість видимих предметів, яка безпосередньо сприймається оком. Для розуміння утворення комп'ютерних колірних зображень варто розглянути наступні поняття:

- Роздільність.
- Глибина кольору.
- Колірна модель.

Роздільність

Оскільки поняття роздільності вживають стосовно різних об'єктів варто розрізняти:

- Роздільність зображення.
- Роздільність екрану.
- Роздільність принтеру.

Роздільність зображення

Роздільність зображення вимірюється у кількості пікселів на дюйм PPI (pixel per inch) і задається при створенні зображення в графічному редакторі або при отриманні оцифрованого зображення внаслідок сканування.

Піксель (pixel скорочення від picture element) — це найменший елемент двовимірного цифрового зображення в комп'ютерній графіці.

Піксель є неподільним об'єктом прямокутної, зазвичай квадратної, або заокругленої форми, що зафарбований в певний колір. Комп'ютерне зображення складається з пікселів, які розташовані по рядках і стовпцях. Один піксель може зберігати інформацію лише про один колір. Чим більше пікселів на одиницю площі містить зображення, тим більш детальним воно є. Максимальна деталізація растрового зображення задається під час його створення. Якщо збільшується розмір зображення, то пікселі перетворюються на крупні зерна. Ступінь деталізації при цьому не зростає, оскільки для забезпечення плавного переходу між початковими пікселями просто додаються нові, значення кольорів яких обчислюється на підставі кольорів сусідніх пікселів початкового зображення.

Фізичний розмір зображення визначає розмір зображення по горизонталі та вертикалі. Може вимірюватися як в пікселях так і в одиницях довжини (міліметри, сантиметри, дюйми). Фізичний розмір зображення задається при створенні зображення, а інформація про розміри зберігається разом із файлом.

Якщо зображення готують для перегляду на моніторі, то його розміри варто зазначити в пікселях, щоб знати, яку ширину екрану воно займатиме. Якщо зображення буде згодом надруковане, то розмір краще задавати в традиційних одиницях довжини (міліметри, сантиметри, дюйми).

Роздільність зображення та розмір його файлу є зворотно пов'язаними між собою. При зменшенні роздільності розмір файлу із зображенням збільшується і навпаки.

Роздільність екрану

На екрані комп'ютерного монітора зображення утворюється з екранних пікселів. Екранний піксель є мінімальним елементом монітору, якому притаманні всі властивості відтворення кольорів, які доступні для даного монітору.

Кількість пікселів в одному дюймі екрану називається роздільністю монітору. Вона залежить від розміру екрану та розміру зерна і, зазвичай, складає [152 ppi](#).

Екранні пікселі складаються з тріад (складових червоного, зеленого і синього кольорів, розташованих поряд в певній послідовності). Коли зображення чи веб-сторінка виводиться на екран монітора, всі їх пікселі представляються за допомогою певного числа екранних

пікселів. Роздільність монітора визначає розмір екранного зображення, і її не слід плутати з роздільністю зображення, що характеризує щільність пікселів в зображенні.

Наприклад, розмір фотографії з роздільністю 300 ppi на екрані монітора з роздільністю 152 ppi буде приблизно вдвічі перевищувати реальний розмір, оскільки в кожному дюймі екрану може бути відображено лише 152 з 300 пікселів. При виведенні на монітор з роздільністю 120 ppi те ж саме зображення буде дещо більшим за оригінал, оскільки в цьому випадку в кожному дюймі екрану може міститися 120 з 144 пікселів.

Наступною характеристикою є роздільна здатність екрану і вона залежить від показників відеокарти та поточних налаштувань операційної системи. Вимірюється в кількості пікселів по ширині та висоті екрану. Наприклад, 800x600, 1024x768, 1600x1200, 1920x1020. Роздільну здатність можна легко змінювати під особисті потреби користувача. При низькій роздільній здатності екранні пікселі стають збільшеними, при високій роздільній здатності навпаки, екранні пікселі стають дрібнішими, зображення зменшуються і відповідно збільшується робочий простір екрану.

Максимальну ширину web-сторінки слід вибирати виходячи з максимально можливої роздільної здатності пристрою (в пікселях) по ширині. Наприклад, ширину web-сторінку слід обмежити 800 пікселями (а краще і дещо меншого розміру) для її перегляду на екрані планшета з роздільною здатністю 800x600. При більшій ширині для її перегляду доведеться користуватися смугами прокрутки, що є не прийнятним.

Роздільність принтеру

Це властивість принтеру, що показує, скільки окремих точок може бути надруковано на ділянці певної довжини.

Роздільність принтерів вимірюється в точках на дюйм DPI (dots per inch), визначає якість зображення в заданому розмірі і може бути змінена користувачем. В середньому сучасні струменеві принтери мають роздільність 300-600 dpi, лазерні 600-1200 dpi.

Одиниці вимірювання DPI і PPI часто плутають. По-перше, незважаючи на те, що термін PPI відноситься до роздільності цифрового зображення, яке з'являється на екрані, показник PPI може також впливати на якість зображення, надрукованого на папері. По-друге, деякі професійні сервіси друку вимагають, щоб зображення мали певний рівень DPI для того, щоб вони могли бути коректно надруковані; те, що вони зазвичай мають на увазі, є кількістю PPI (пікселів на дюйм), а не DPI - і таким чином, це тільки додає плутанини.

Якщо ви розділите розмір зображення (загальна кількість пікселів цифрового зображення) на його роздільність, ви можете визначити розмір зображення, відображеного на екрані або надрукованого на папері в дюймах (см). Приклад: зображення 3000x2000 пікселів, роздруковане з роздільністю 300 dpi матиме розмір відбитка (ширина) 10 дюймів (3000 пікселів / 300 dpi)

Пікселі і глибина кольору

Для більшого розуміння відображення на моніторі чи аркуші графічного файлу уявимо собі певне зображення і нанесемо на нього прямокутну сітку з маленькими квадратними комірками. Кожній комірці сітки відповідає один піксель, який в ній знаходиться.

Тобто, для опису зображення потрібно його розкласти на множину елементів — пікселів і визначити спосіб опису одного пікселя, щоб описати все зображення як сукупність описів окремих пікселів. Цей опис зберігається у спеціальних графічних форматах, яких налічується більше десятка.

Глибиною кольору називають кількість біт, що відведено для кодування кольору одного пікселя.

- 1 біт ($2^1 = 2$ кольори). Бінарний колір, найчастіше чорний та білий.
- 1 байт (8-біт) ($2^8 = 256$ кольорів). Індексована палітра кольорів, відтінки сірого.
- 2 байти (16-біт) ($2^{16} = 65\,536$ кольорів). Режим High Color.
- 3 байти (24-біт) ($2^{24} = 16,5$ мільйонів кольорів). Режим True Color.

Очевидно, чим більшою є глибина кольору, тим більшим буде об'єм вихідного файлу.

Колірні моделі математичного опису кольору

Для опису кольору пікселів використовують універсальні колірні моделі, де застосовується точний опис кольору в стандартизованих цифрових і математичних виразах.

Існує багато різних моделей опису кольору, але всі вони належать до одного з трьох типів:

- Адитивні. Засновані на складанні випромінювань окремих зон спектру світла і застосовуються для середовищ, що випромінюють світло (екран, сканування).

- Субтрактивні. Засновані на поглинанні окремих зон спектру світла при відбиванні або пропусканні світла і застосовуються у середовищах, що поглинають світло (забарвлені поверхні, барвники).

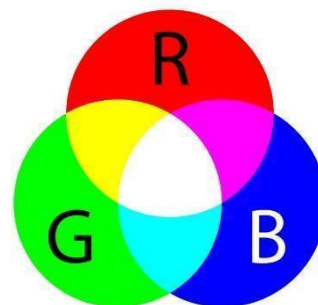
- Психологічні. Засновані на сприйнятті кольорів людиною і пов'язані з особливостями його зорової системи.

Найпоширенішими моделями є:

- RGB — адитивна колірна модель.
- CMYK — субтрактивна колірна модель.
- HSB та Lab — психологічні колірні моделі.

Модель RGB

Модель RGB є найбільш простою для розуміння і застосовується в пристроях, що випромінюють світло або колір утворюється електронним шляхом: у моніторах, телевизорах, проекторах, сканерах тощо.



Колір в моделі RGB представлено сумою яскравостей трьох базових кольорів — червоного (Red), зеленого (Green) і синього (Blue). Назву моделі утворено з перших літер англійських назв цих кольорів.

Для опису кольору в моделі RGB застосовують 3 байти (24 біти=3x8 біт), по одному байту на кожен колірну складову.

Яскравість (інтенсивність) кожного базового кольору може приймати 256 (2^8) дискретних значень від 0 до 255. Змішування кольорів в різних пропорціях, варіюючи яскравість кожної складової, складає $256 \times 256 \times 256 = 16\,777\,216$ кольорів.

Кожний колір можна представити в закодованому вигляді, що містить значення яскравості трьох колірних складових.

Десяткове і шістнадцяткове представлення кольору

Десяткове представлення — це трійка десяткових чисел, що розділені комами. Перше число відповідає яскравості червоної складової, друге — зеленої, а третє — синьої.

Наприклад, червоний колір: (255,0,0); зелений колір: (0,255,0), синій колір: (0,0,255).

Код кольору в шістнадцятковому представленні має вид 0xXXXXXX. Префікс 0x вказує на те, що в коді використано власне шістнадцяткове число. Після префіксу зазначено шість шістнадцяткових цифр (0, 1, 2 ..., 9, A, B, C, D, E, F).

Перші дві цифри — шістнадцяткове число, що представляє яскравість червоної складової, друга і третя пари відповідають яскравості зеленої та синьої складових.

Наприклад, червоний колір: 0xFF0000; зелений: 0x00FF00; синій: 0x0000FF.

Якщо всі складові мають максимальну яскравість (255,255,255 — в десятковому представленні; 0xFFFFFF — в шістнадцятковому представленні), виходить білий колір. Мінімальна яскравість (0, 0, 0 або 0x000000) відповідає чорному кольору.

Змішування базових кольорів

- Червоний і зелений — за максимальної яскравості утворюють жовтий колір.

Жовтий колір: 255,255,0 або 0xFFFF00

- Зелений і синій — за максимальної яскравості надають блакитний колір.

Блакитний колір: 0,255,255 або 0x00FFFF

- Червоний і синій — за максимальної яскравості надають фіолетовий колір.

Фіолетовий колір: 255,0,255 або 0xFF00FF

Слід також відзначити, що в записі шістнадцяткового значення кольору можна використовувати як великі, так і маленькі латинські літери, наприклад, запис #00FF00 рівнозначна запису #00ff00.

Нижче представлена таблиця 16 стандартних кольорів разом з їх шістнадцятковий

кодами.

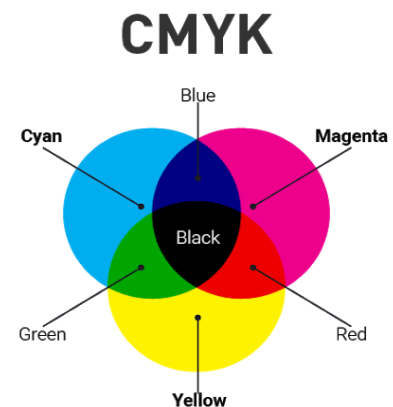
Колір	Код	Колір	Код
black (чорний)	# 000000	silver (срібний)	# C0C0C0
maroon (темно-бордовий)	# 800000	red (червоний)	# FF0000
green (зелений)	# 008000	lime (вапно)	# 00FF00
olive (оливковий)	# 808000	yellow (жовтий)	# FFFF00
navy (темно-синій)	# 000080	blue (синій)	# 0000FF
purple (фіолетовий)	# 800080	fuchsia (фуксія)	# FF00FF
teal (синьо-зелений)	# 008080	aqua (аква)	# 00FFFF
gray (сірий)	# 808080	white (білий)	# FFFFFFFF

Той факт, що розробник Web-сторінки нічого не знає про комп'ютер, на якому документ буде проглядатися, накладає додаткові обмеження на використання RGB-моделі. На деяких моніторах неможливо відобразити все розмаїття відтінків. Браузери в цьому випадку скорочують використовувану кількість кольорів, перепризначаючи їх під власні палітри. Використання розробниками сайтів **Web-палітри** є в деякій мірі гарантією того, що документ буде виглядати однаково на різних дисплеях. Кольори, RGB-компоненти яких входять в Web-палітру, називають **Web-надійними** кольорами. Коди Web-надійних кольорів можна знайти як на онлайн-ресурсах (наприклад тут: <https://htmlcolorcodes.com/>), так і в програмі графічного редактора, клікнувши на палітру кольорів, далі поставити галочку біля “Тільки веб-кольори” і вибрати потрібний колір

Модель CMYK

Модель RGB добре підходить для моделювання кольорів, що утворюються електронним шляхом, але не є доречною у разі моделювання кольорів для друкованих зображень.

Модель CMYK описує змішування фарб, які використовують для друкованих зображень. В цій моделі теж використовуються три базові кольори: блакитний (Cyan), фіолетовий (Magenta) і жовтий (Yellow). Крім того, застосовується чорний колір (black). (Зазвичай, абревіатуру CMYK вимовляють, як «цмік» або «смік»).



Кожен з трьох базових кольорів моделі CMYK утворюється в результаті вилучення з білого кольору одного з базових кольорів моделі RGB. Так, наприклад, блакитний (Cyan) утворюється внаслідок вилучення червоного з білого, а жовтий (Yellow) — вилученням синього.

- Cyan = RGB - R = GB = (0, 255, 255);
- Magenta = RGB - G = RB = (255, 0, 255);
- Yellow = RGB - B = RG = (255, 255, 0).

Оскільки базові кольори CMYK утворюються внаслідок вилучення з білого кольору базових кольорів RGB, їх називають додатковими, бо вони доповнюють основні кольори до білого, а модель CMYK субтрактивною (та, що віднімає).

Базові кольори моделі CMYK є достатньо яскравими кольорами і не цілком підходять для відтворення темних кольорів. Так, при їх змішуванні на практиці утворюється не справжній чорний, а брудно-коричневий колір. Тому, колірна модель CMYK містить ще і чистий чорний колір, який використовується для створення темних відтінків, а також для друкування чорних елементів зображення.

Представлення кольору

Колір в CMYK описується четвіркою десяткових чисел, кожна з яких визначає відсоток фарби даного кольору. Наприклад, для отримання кольору «хакі» слід змішати 30%

блакитної фарби, 45% фіолетової фарби, 80% жовтої фарби і 5% чорної. Позначити такий вибір можна в два способи:

(30,45,80,5) або C30M45Y80K5.

В друкарнях колірні зображення друкують в кілька прийомів. Накладаючи на папір почергово блакитний, фіолетовий, жовтий і чорний слої, отримують повноколірну ілюстрацію. Тому, готове зображення, що створено на комп'ютері, перед друком розділяють на чотири складові одноколірні зображення. Цей процес називається кольороподілом. Сучасні графічні редактори мають засоби для виконання цієї операції.

Модель CMYK використовують для підготовки друкарських зображень. Тут, світло не випромінюється, а поглинається. Чим більше фарби покладено на папір, тим більше світла вона поглинає і менше відбиває. Поєднання трьох основних фарб поглинає майже все падаюче світло, і з боку зображення виглядає майже чорним. На відміну від моделі RGB збільшення кількості фарби приводить не до збільшення візуальної яскравості, а навпаки, до її зменшення.

Моделі HSB і HLS

Модель HSB засновано на трьох параметрах:

- H — відтінок або тон (Hue).
- S — насиченість (Saturation).
- B — яскравість (Brightness).

Модель HSB краще за моделі RGB і CMYK, відповідає поняттю кольору, яке використовують професійні художники. Дійсно, в них, зазвичай, є кілька основних фарб, а решта утворюються внаслідок додавання до них білої або чорної фарб. Таким чином, потрібні кольори — це певна модифікація основних кольорів: висвітлення або затемнення.

В певних графічних редакторах, наприклад, в Adobe FreeHand, використовується модель HLS (Hue, Lightness, Saturation), яка є схожою до HSB. У моделі HLS, на відміну від HSB, замість яскравості застосовано параметр L — освітленість (Lightness). Зменшення освітленості наближає колір до чорного, а збільшення — до білого. Чистий спектральний колір утворюється при освітленості 50%.

Моделі HSB і HLS є апаратно незалежними і не орієнтовані на жоден технічний пристрій.

Модель Lab

Модель Lab засновано на трьох параметрах:

- L — яскравість (Luminosity)
- Два колірні параметри — a і b. Параметр a містить кольори від темно-зеленого через сірий до яскраво-рожевого. Параметр b містить кольори від світло-синього через сірий до яскраво-жовтого.

Поняття яскравості в моделях Lab і HSB не є тотожними. Як і в RGB, змішування кольорів зі шкал a і b дозволяє отримати яскравіші кольори. Зменшити яскравість кольору можна за рахунок параметра яскравості L.

Модель Lab є апаратно незалежною, її колірний діапазон покриває діапазони RGB і CMYK. Графічний редактор Adobe Photoshop при переході від режиму RGB до CMYK використовує Lab як проміжний етап.

Колірний обхват

Величезна кількість різних кольорів, які сприймає людина, може бути представлена на екрані монітора і на папері. Проте, монітор не може відтворити в точності всі кольори, які присутні в природі. Наприклад, він погано відтворює чисті блакитний і жовтий кольори. Частина кольорів, що відображаються монітором, можна надрукувати, проте, при друкуванні погано передаються кольори, що мають дуже низьку щільність. Тому, для моделей існує важлива характеристика - колірний обхват або діапазон (Gamut) колірних моделей.

Найбільшим колірний обхват має модель Lab, в ній можна представити практично всі природні кольори, які здатна сприйняти людина. Власне, з цією метою вона і створювалася.

Запис у файл

У заголовок файлу записуються відомості про колірну модель, розміри зображення, глибину кольору та певні інші відомості. Після заголовку у файл записуються один за одним

коди кольорів (або параметрів колірної моделі) окремих пікселів, зліва направо і зверху вниз. Така структура файлу є досить узагальненою і реалізована у форматі BMP.

3. Формати графічних файлів для Веб

У підготовці зображень для веб-сторінок варто бути обізнаним в популярних графічних форматах та їх властивостях. Комп'ютерні графічні файли можна поділити на дві великі гілки: растрову і векторну.

- Векторні файли представляють математичний опис об'єктів відносно точки відліку координат. Наприклад, для того, щоб відобразити пряму потрібно вказати координати двох точок, які об'єднуються за коротшим шляхом, для дуги чи кола задається радіус тощо. Таким чином, векторне зображення є набором геометричних примітивів. Складність при переведенні чи перенесенні даних з одного векторного формату до іншого полягає у використанні в програмах різних алгоритмів, різних математичних формул для побудови векторних примітивів та опису растрових об'єктів.

- Растровий файл влаштовано простіше. Він представляє прямокутну матрицю (bitmap), що поділена на пікселі. Растрові формати різняться між собою здатністю містити додаткову інформацію: різні колірні моделі, вектори, Альфа-канали, слої різних типів, інтерліньяж, анімацію, можливості стиснення тощо.

Растрові веб-формати

GIF (CompuServe Graphics Interchange Format)

<https://www.w3.org/Graphics/GIF/spec-gif89a.txt>

Апаратно незалежний формат GIF розроблено в 1987 році (GIF87a) фірмою CompuServe для передачі растрових зображень по мережах. У 1989-му формат модифіковано (GIF89a), до нього додано підтримку прозорості та анімації.

GIF використовує LZW-компресію, що дозволяє добре стискати файли, в яких багато ділянок з однорідним заповненням (логотипи, написи, схеми).

Метод стиснення LZW (Lempel-Ziv-Welch) розроблено в 1978 році ізраїльтянами Лемпелом і Зівом, і доопрацьовано пізніше в США. Принцип стиснення полягає у пошуку однакових послідовностей - фраз у всьому файлі. Виявлені послідовності зберігаються в таблиці, їм присвоюються короткі маркери - ключі. Так, якщо в зображенні є набори з рожевого, оранжевого і зеленого пікселів, що повторюються 50 разів, LZW виявляє цей набір, присвоює йому окреме число (наприклад, 7) і зберігає ці дані 50 разів у вигляді числа 7.

Метод LZW ефективно діє на ділянках однорідних, вільних від шуму кольорів, добре стискає довільні графічні дані, але процес кодування і розпаковування відбувається відносно повільно.

У GIF можна застосовувати прозорі ділянки, вони лишаються прозорими в браузерях та інших програмах і через них просвічується фоновий колір. Прозорість забезпечується за рахунок додаткового Alpha-каналу, що зберігається разом з файлом.

Файл GIF спроможні містити кілька растрових картинок, які браузері підвантажують одну за іншою із вказаною у файлі частотою. Так досягається ілюзія руху (GIF-анімація).

Особливості

- Кількість кольорів в зображенні може бути від 2 до 256, але це можуть бути будь-які кольори з 24-бітової палітри.

- Файл у форматі GIF може містити лише 100% прозорі ділянки. Якщо використовується відмінний від білого кольору фон, він буде просвічуватися крізь прозорі ділянки в зображенні.

- GIF підтримує покадрову зміну зображень, що робить формат популярним для створення банерів і простої анімації.

- Використовує вільний від втрат метод стиснення.

Область застосування

Текст, логотипи, ілюстрації з чіткими краями, анімовані малюнки, зображення з прозорими ділянками, банери.

JPEG (Joint Photographic Experts Group)

<https://jpeg.org/>

JPEG це є назва формату та алгоритму стиснення, який засновано не на пошуку однакових елементів, як у LZW, а на різниці між пікселями.

Кодування даних відбувається в кілька етапів. Спочатку графічні дані конвертуються в колірний простір типу LAB, відкидається половина або три чверті інформації про колір (в залежності від реалізації алгоритму).

Далі аналізуються блоки 8x8 пікселів. Для кожного блоку формується набір чисел. Перші кілька чисел представляють колір блоку в цілому, в той час, як подальші числа відображають дрібні деталі.

На наступному етапі, залежно від обраного рівня якості, відкидається певна частина чисел, що представляють дрібні деталі. На останньому етапі використовується кодування за методом Хафмана для ефективного стиснення кінцевих даних. Відновлення даних відбувається в зворотному порядку.

Метод стиснення Хафмана (Huffman) розроблено в 1952 році і використовується як складова частина в ряді інших схем стиснення, в тому числі і у LZW. В методі Хафмана аналізується набір символів для визначення частоти кожного символу. Для символів, що найчастіше зустрічаються, використовується позначення у вигляді мінімальної можливої кількості бітів. Наприклад, найчастіше в англійських текстах зустрічається буква "e". Використовуючи кодування Хафмана можна представити літеру "e" лише двома бітами (1 і 0), замість вісьмох бітів, необхідних для представлення букви "e" в кодуванні ASCII.

Таким чином, чим вище рівень компресії, тим більше даних відкидається, тим нижчою є якість. Використовуючи JPEG можна отримати файл в 2-500 разів менше, ніж BMP.

Формат JPEG є апаратно незалежним. JPEG відтворює спектр кольорів TrueColor (224).

JPEG краще стискає растрові картинки фотографічної якості, ніж логотипи або схеми. Краще і з меншими втратами стискаються великі зображення для Веб з високою роздільністю (200-300 ppi і більше), ніж з низькою (72-150 ppi), оскільки в кожному квадраті 8x8 пікселів переході виходять м'якшими, за рахунок того, що таких квадратів є більше.

Даний формат називають стисненням з втратами, оскільки алгоритм JPEG вибірково відкидає дані. Не бажано зберігати у JPEG-форматі будь-які зображення, де важливими є дрібні нюанси кольорів, оскільки під час стиснення відбувається відкидання колірної інформації. У JPEG слід зберігати лише кінцевий варіант роботи, оскільки кожне повторне збереження призводить до нових втрат (відкидання) даних і початкове зображення може бути вкрай зіпсованим.

Формат JPEG не підтримує прозорість і при збереженні зображення з прозорими ділянками, вони зафарбовуються в певний колір.

Особливості

- В зображенні може бути понад 16 мільйонів кольорів, що цілком достатньо для збереження фотографічних зображень.

- Основною характеристикою формату є якість, яка визначає кінцевий розмір файлу. Слід пам'ятати, що формат застосовує стиснення з втратами. Чим вище стиснення, тим менше якість і навпаки.

- Підтримка технології прогресивного JPEG. Спочатку у вікні перегляду з'являється версія зображення з низькою роздільністю, яке при повному завантаженні поступово набуває початкового вигляду.

Область застосування

Використовується переважно для фотографій. Не є доцільним для зображень з прозорими ділянками, великими одноколірними ділянками.

PNG (Portable Network Graphics)

<http://www.libpng.org/pub/png/>

PNG – Інтернет формат, який долає обмеження GIF. Використовує стиснення без втрат Deflate, подібне до LZW. Стиснуті індексовані файли PNG, зазвичай, є меншими за аналогічні GIF.

Глибина кольору може бути будь-якою, до 48 біт. Використовується двохвимірний interlacing (не лише рядків, але і стовпців), що, подібно до GIF, дещо збільшує розмір файлу.

На відміну від GIF, де застосовується 100% прозорість, PNG підтримує також напівпрозорі пікселі (в діапазоні прозорості від 0 до 99%) за рахунок Альфа-каналу з 256 градаціями сірого.

У файл формату PNG записується інформація про гамму. Гамма є певним числом, що характеризує залежність яскравості світіння екрану монітора від напруги на електродах кінескопа. Це число обчислюється з файлу і дозволяє вводити поправку яскравості при відображенні. Воно потрібне для однакового відображення інформації незалежно від апаратної платформи комп'ютера. PNG підтримується у всіх сучасних браузерях.

PNG-8

PNG-8 — формат подібний до GIF і має покращений формат стиснення даних.

Особливості

- Використовує 8-бітову палітру (256 кольорів) в зображенні, за що і отримав в своїй назві цифру вісім. При цьому можна вибирати, скільки кольорів буде задіяно у файлі — від 2 до 256.

- На відміну від GIF, не відображає анімацію.

Область застосування

Текст, логотипи, ілюстрації з чіткими краями, зображення з градієнтною прозорістю.

PNG-24

PNG-24 — формат, аналогічний до PNG-8, але використовує 24-бітову палітру кольору. Подібно до формату JPEG, зберігає яскравість і відтінки кольорів у фотографіях. Подібно до форматів GIF і PNG-8, зберігає деталі зображення та прозорість.

Особливості

- Використовує понад 16 млн. кольорів, тому, застосовується для повнокольорових зображень.

- Підтримує багаторівневу прозорість, це дозволяє створювати плавний перехід від прозорої області зображення до колірної, так званий градієнт.

- Алгоритм стиснення зберігає всі кольори і пікселі в зображенні незмінними. Якщо порівнювати з іншими форматами, то в PNG-24 кінцевий об'єм графічного файлу виходить найбільшим.

Область застосування

Фотографії, малюнки, що містять прозорі ділянки, малюнки з великою кількістю кольорів і чіткими краями зображень.

Векторні веб-формати

SVG. Масштабована векторна графіка

SVG (Scalable Vector Graphics) - це тип векторних файлів, що описують зображення в форматі XML. Формат з'явився в 2001, однак популярність серед веб-розробників він отримав нещодавно, після впровадження підтримки у сучасні браузери. Формат є відкритим стандартом, на відміну від більшості інших форматів, SVG не є чияюсь власністю.

Файл із зображенням в цьому форматі є звичайним текстовим файлом, який можна відкрити в блокноті і відредагувати. У цьому форматі можна описати не тільки статичну, а й динамічну картинку (анімація), змішати створені вектори з растровою картинкою. Завдяки тому, що кожна фігура для браузера є елементом DOM, за допомогою JavaScript можна описувати досить складні сценарії, взаємодіяти з користувачем.

Розмір об'єктів SVG є меншим за розмір растрових зображень, а самі зображення не втрачають якості при масштабуванні. На відміну від растрових форматів можна взаємодіяти із зображеннями у форматі SVG - за допомогою CSS можна змінювати параметри графіки: колір, прозорість або межі, а за допомогою JavaScript - анімувати зображення.

У SVG є маса функцій, які роблять цей формат рекомендованим для Вебу, особливо якщо SVG використовується для простих зображень типу логотипів, карт, іконок, маркерів.

Переваги формату SVG

- SVG часто важать менше за растрові зображення.
- Формат масштабується, що забезпечує чіткість за будь-якої роздільності екрану.
- SVG-код можна помістити в HTML і налаштувати через CSS.

- SVG-зображення можна анімувати, в тому числі окремі частини, як за допомогою CSS, так і JS.

Втім занадто складні SVG-зображення збільшують розмір файлу. SVG не застосовний до фотографій, тут краще підійдуть формати JPG та WebP.

Анімаційні веб-формати

APNG. Анімований PNG

aPNG (Animated Portable Network Graphics) запропоновано в 2004 році компанією Mozilla. Новий стандарт базується на форматі PNG, додано можливість анімації та покращено алгоритми стиснення. Втім компанія, що підтримує формат PNG не визнала новий стандарт, і aPNG не пішов в реліз. Основна причина відмови звучала так «PNG - це формат для нерухомих зображень».

До 2008 року не було жодних спроб широкого впровадження aPNG. В 2008 році, компанія Mozilla вносить його в свій браузер Firefox, пізніше таку підтримку додано в браузер Opera. З приходом HTML5, ситуація почала змінюватися. Google, Twitter, Facebook і інші популярні сервіси, стали один за одним відмовлятися від застарілих технологій, і переходити на нові, паралельно, задаючи стандарти розробки.

WebP

<https://developers.google.com/speed/webp/>

Сучасний формат зображень з відкритим вихідним кодом, розроблений компанією Google спеціально для Інтернету, є відгалуженням відеоформату WebM.

Подібно до APNG, він також підтримує 24-бітні зображення, 8-розрядну прозорість. У ньому є стиснення з втратами та без втрат, що у деяких випадках дозволяє досягти невеликого розміру файлів із забезпеченням високої якості, що робить його універсальним. Формат поєднує переваги JPG і PNG без збільшення розміру файлу. Сьогодні YouTube використовує перетворення мініатюр для відео в формат WebP.

Поки формат підтримується лише браузером Chrome, і має певні недоліки щодо відображення дрібних деталей. Для підтримки в інших браузерах існують способи обходу обмежень (додаткові плагіни або конфігурація), але вони не сприяють використанню формату повсюдно.

FLIF Free Lossless формат зображення

<http://flif.info/>

FLIF - це новий та інноваційний формат зображення, що підтримує анімацію з альфа-прозорістю, прогресивне завантаження (що дозволяє відобразити зображення в меншій якості, поки він довантажується), а також інші можливості у порівнянні з різними форматами. Формат поки не підтримується всіма браузерами, але розробляються JavaScript-плагіни, що дозволяють використовувати FLIF у будь-якому браузері.

Інші формати графічних файлів

PSD (Adobe Photoshop Document)

Внутрішній формат популярного растрового редактора Photoshop останнім часом підтримується великою кількістю програм. Він дозволяє записувати зображення з багатьма шарами, їх масками, додатковими Альфа-каналами і каналами базових кольорів, контурами та іншою інформацією.

Для стиснення застосовують метод RLE (Run Length Encoding), кодування із змінною довжиною рядка. Дія методу полягає в пошуку однакових пікселів в одному рядку. Якщо в рядку, наприклад, є 3 пікселя білого кольору, 21 - чорного, 14 - білого, то застосування RLE надає можливість не запам'ятовувати кожен з них (38 пікселів), а записати як 3 білих, 21 чорний і 14 білих в першому рядку.

Подібно до методу LZW, алгоритм RLE добре працює з штучними і ілюстрованими картинками і гірше з фотографіями. У випадку, якщо фотографія має багато дрібних деталей, RLE може навіть збільшити розмір файлу.

TIFF (Tagged Image File Format)

Апаратно незалежний формат TIFF, на сьогоднішній день є одним з найпоширеніших і надійніших, його підтримують практично всі програми на PC і Macintosh, що пов'язані з

графікою. TIFF є кращим вибором при імпорті растрової графіки у векторні програми та видавничі системи. Йому доступно весь діапазон кольірних моделей від монохромної до RGB, CMYK і додаткових кольорів Pantone. TIFF може зберігати контури, Альфа-канали та інші додаткові дані.

TIFF має два різновиди: для Macintosh і PC. Це пов'язано з тим, що процесори Motorola читають і записують числа зліва направо, а процесори Intel - навпаки. Сучасні програми можуть без проблем використовувати обидва варіанти формату. Зазвичай, дані у форматі TIFF не стискаються, але може бути використано LZW-стиснення.

BMP (Windows Device Independent Bitmap)

Рідний формат Windows, який підтримується всіма графічними редакторами, що працюють під управлінням цієї операційної системи. Застосовується, в основному, для збереження растрових зображень, що призначені для використання в Windows. Може зберігати як індексовані (до 256 кольорів), так і RGB-кольори (понад 16 млн. відтінків). Можливе застосування стиснення за алгоритмом RLE.

PDF (Portable Document Format)

PDF запропоновано фірмою Adobe, як платформи-незалежний формат для створення електронної документації, презентацій, передачі верстки та графіки через мережі.

Односторінкові файли PDF відмінної якості може створювати Photoshop. Багатосторінкові PDF можуть створювати програми Adobe Acrobat, PageMaker і програми пакету MS Office.

PDF спочатку проектувався як компактний формат для електронної документації. Тому, всі дані в ньому можуть стискатися, причому, до різного типу інформації застосовуються різні, найбільш прийнятні для них типи стиснення: JPEG, RLE, CCITT, ZIP. Програма Acrobat дозволяє розставляти гіперпосилання, заповнені поля, додавати у файл PDF відео чи звук, інші дії.

Файл PDF може бути оптимізованим. З нього видаляються елементи, що повторюються, встановлюється посторінковий порядок завантаження сторінок через Веб, з пріоритетом спочатку для тексту, потім для графіки. Якщо елементів, що повторюються, немає, файл, після оптимізації, як правило, дещо збільшується.

PDF найчастіше використовується для передачі по мережах графіки і зверстаного тексту в компактному виді. Він може зберігати всю інформацію для пристрою виведення, яка була в початковому файлі.

WMF (Windows Metafile)

Векторний формат WMF є рідним форматом Windows і використовує його графічну мову. Призначений для передачі векторних даних через буфер обміну (Clipboard). Розпізнається практично всіма програмами Windows, так чи інакше пов'язаними з векторною графікою. Користуватися форматом WMF варто лише у випадках передачі «чистих» векторів. WMF спотворює колір, не зберігає певні параметри, які присвоюються об'єктам в різних векторних редакторах, не містить растрові об'єкти, не розпізнається багатьма програмами на Macintosh.

AI (Adobe Illustrator Document)

Adobe Illustrator – популярний графічний редактор від Adobe. Векторний формат Illustrator можна безпосередньо відкрити у Photoshop, його підтримують майже всі програми Macintosh і Windows так чи інакше пов'язані з векторною графікою і графікою взагалі.

Формат Illustrator є найкращим посередником при передачі векторів з однієї програми в іншу, з PC на Macintosh і навпаки. Втілені або пов'язані з документом растрові файли при обміні через формат Illustrator втрачаються.

CRD (CorelDraw Document)

Векторний формат, що має незаперечне лідерство на платформі PC. Багато програм на PC (FreeHand, Illustrator, PageMaker тощо) можуть імпортувати файли CorelDraw.

У файлах застосовується окрема компресія для векторів і растрів, можуть втілюватися шрифти, файли CorelDraw мають величезне робоче поле 45x45 метрів (цей параметр є важливим для зовнішньої реклами), підтримується багатосторінковість.

Також доступна можливість створення зображень на веб-сторінці програмним шляхом, зокрема з допомогою написання коду засобами CSS, SVG, Canvas та JS-анімації.

Питання для закріплення.

- 1) Що таке « web-дизайн»?
- 2) Дайте визначення поняттю « web-сторінка»
- 3) Назвіть допустимий обсяг web-сторінки.
- 4) Назвіть основні рекомендації при створенні web.

Лекція 3. Етапи створення графічного макету сайту

План лекції

1. Планування.
2. Розробка графічної концепції.
3. Вибір колірної схеми та художнього стилю.
4. Графічний макет.
 - a. Структура сторінки.
 - b. Модульна сітка.
 - c. Позиціювання і ширина сторінки.
5. Юзабіліті.
 - a. Продуманий інтерактив (навігація, кнопки, посилання тощо).
 - b. Типографіка.
6. Впорядкування вихідних файлів шаблону.

Вступ

Веб-дизайн - це наука, де існують загальноприйняті норми для розробки дружнього веб-інтерфейсу, незалежно від подальшого призначення сайту - візитка, корпоративний, промо, Інтернет магазин тощо. Дотримання цих норм і правил буде запорукою злагодженої роботи решти учасників веб-проекту: верстальників, програмістів, оптимізаторів і популярності ресурсу серед Інтернет-спільноти.

1. Планування

Робота дизайнера розпочинається із ознайомлення з технічним завданням (ТЗ, бріфом) за яким потрібно скласти фінальний дизайн сайту. На папері чи ПК створюється схема або макет (бажано кілька), де буде враховано специфіку ресурсу, меню, тексти, розташування блоків, посилання, ілюстрації тощо. Коли дизайнеру буде зрозуміла структурна й логічна картина майбутнього сайту, тоді можна переходити до реалізації дизайну.

Технічне завдання або Бріф (від англ. brief) - коротка описова форма погоджувального порядку між замовником проекту і його виконавцем. По суті, Бріф на сайт - опитувальник клієнта.

Основні функції ТЗ:

- Організаційна
- Інформаційна
- Юридична

ТЗ затверджується замовником і підписується з двох сторін. Складання ТЗ займає ~ 1/5 частина часу від усіх робіт. Роботи зі складання Бріфа (ТЗ) є окремо оплачуваними.

Основні пункти ТЗ, які важливі для веб-дизайнера:

1. Загальні відомості
 - 1.1 Офіційне повне та скорочене найменування організації Замовника
 - 1.2 Контакти Замовника: поштова адреса, номер телефону, електронна пошта, сайт, реквізити.
 - 1.3 Призначення і цілі створення Сайту
 - Чим займається компанія?
 - Мета створення сайту?

- Які функції сайт повинен виконувати?
- Яка інформація повинна бути опублікована?

Правильно, конкретно поставлена мета дозволяє зробити сайт ефективним інструментом для замовника, мінімізувати витрати на створення сайту.

Приклади формулювання мети

- Публікація інформації про компанію, її послуги.
- Здійснювати віддалену демонстрацію товару для його просування.
- Здійснювати інформаційну та сервісну підтримку клієнтів і партнерів.
- Покращення іміджу компанії, ін.

1.3 Опис цільової аудиторії наприклад:

- Сайт навчального закладу: студенти, абітурієнти.
- Сайт інтернет-магазину жіночого одягу: жінки, переважно до 40 років.

1.4 Перелік нормативних правових актів і локальних документів, які повинні враховуватися при розробці сайту.

2. Вимоги до сайту в цілому

2.1 Загальні вимоги до оформлення і верстки сторінок, наприклад:

- сайт повинен володіти адаптивним веб-дизайном, забезпечувати відображення вмісту на різних пристроях, підключених до мережі Інтернет і динамічно підлаштовуватися під розміри вікна: (перераховуються всі розміри);

- верстка сайту повинна бути фіксованою, ширина всіх сторінок сайту повинна становить 1024 px.

Верстка веб-сторінок - процес формування веб-сторінок в редакторі вихідного коду, а також результат цього процесу (веб-сторінки).

Види верстки (по ширині сторінки):

- Фіксована - встановлюється загальна ширина макету рівна певній величині.
- Гумова - при зміні розмірів вікна відбувається пропорційне зменшення/збільшення ширини елементів веб-сторінки, щоб вони вписалися в нову ширину.
- Адаптивна - елементи можуть приховуватися, змінювати місце розташування чи замінюватися іншими.

Приклади розмірів екранів для адаптивної верстки:

Laptop L: ширина <=1440 px;

Laptop: ширина <=1024 px;

Tablet: ширина <=768 px;

Mobil L: ширина <=425 px;

Mobil M: ширина <=375 px;

Mobil S: ширина <=320 px;

Вказані одиниці ширини - це т. з. Breakpoints (контрольні точки, точки зупинки) - це тригери ширини, які визначають вибір адаптивних макетів у залежності від розмірів пристроїв або областей виводу.

Для розробки адаптивного веб-макету можна використати медіа-запити або один із найпопулярніших CSS фреймворків [Bootstrap](#).

Медіа-запити - це правила CSS, які дозволяють умовно застосовувати стилі на основі набору параметрів пристрою виводу.

Приклад медіа запиту:

```
@media all and (max-width:728px) {
```

<тут задаємо розташування і розміри блоків для перегляду сайту на екранах, ширина яких не перевищує 728 пікселів>

```
}
```

```
@media all and (max-width:1024px) {
```


<тут задаємо розташування і розміри блоків для перегляду сайту на екранах, ширина яких не перевищує 1024 пікселів>

}

@media all and (min-width:1024px) {

<тут задаємо розташування і розміри блоків для перегляду сайту на екранах, ширина яких перевищує 1024 пікселів>

}

і т.д.

Доступні контрольні точки Bootstrap.

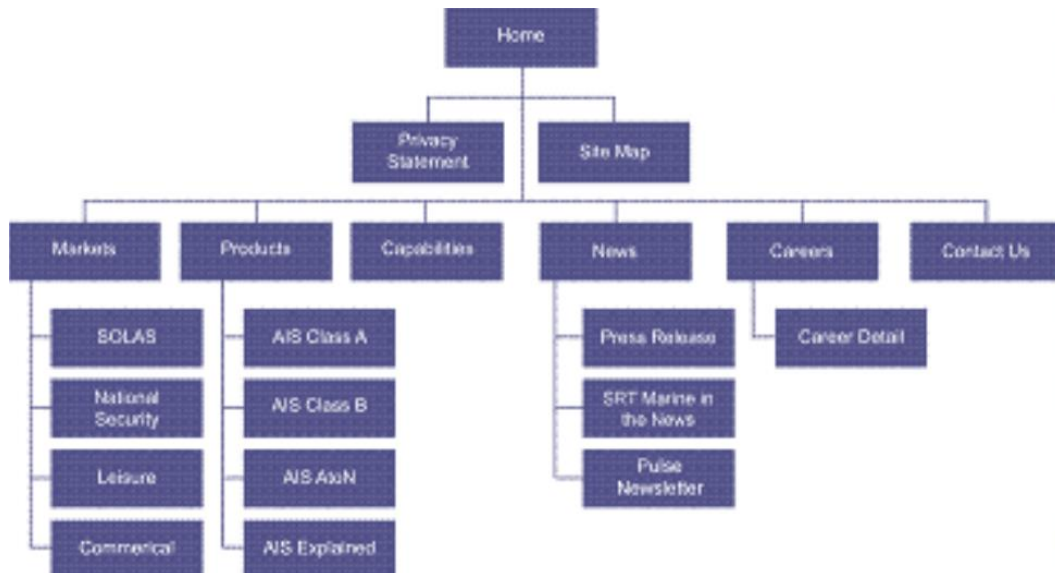
Bootstrap 5.0. за замовчуванням включає в себе шість контрольних точок, іноді їх ще називають рівнями сіток, для швидкої адаптації сторінки. Ці контрольні точки можна налаштувати.

Контрольні точки	Інфікс класу	Розміри
Дуже малий-смафони	Немає	<576px
Малий планшети	sm	≥576px
Середній ноутбуки	md	≥768px
Великий десктопи	lg	≥992px
Дуже великий телевізори	xl	≥1200 px
Надзвичайно великий великі екрани	xxl	≥1400px

Детальніше про створення веб-макетів буде розглянуто в наступному лекційному матеріалі.

2.2 Структура сайту, наприклад:

- Головна сторінка;
- Про нас;
- Послуги;
- Друк
- Ксерокопії
- Брошування
- Контакти та схема проїзду



3. Вимоги до графічного дизайну сайту

3.1 Опис наявного фірмового стилю замовника

3.2 Опис вимог до стилю дизайну, колірної гама (якщо вона не розроблена в фірмовому стилі замовника), до оформлення шрифтів.

Перш ніж почати роботу, потрібно запросити у замовника необхідні для роботи матеріали. Сюди відносяться:

- фірмовий стиль (логотип, фірмові кольори і шрифти, керівництво по фірмовому стилю, фірмовий (корпоративний) герой і т.д.);
- фотоілюстративний матеріал, тобто якісні фотографії, що показують робочий процес (якщо такі є у замовника), фотографії співробітників (якщо передбачається створення сторінки про співробітників) тощо.

Підбір шрифту

Для макета сайту вам, як правило, знадобиться пара шрифтів - один для заголовків, інший для основного тексту. Як максимум можна підібрати третій, якщо дуже потрібно. Що потрібно пам'ятати, коли вибираєш шрифт:

Авторські права. Шрифт потрібно або купити, або знайти той, який можна використовувати безкоштовно. Безкоштовні шрифти шукайте на Google Fonts , Font Space , 1001 free fonts.

Потрібні знаки і накреслення. Іноді у шрифту немає курсиву або напівжирного варіанти, або відсутні спеціальні знаки типу знаки валют чи не підтримують кирилицю.

Фірмовий стиль - це набір постійних кольорних, графічних, словесних, друкарських, дизайнерських елементів, що забезпечують візуальну і смислову єдність товарів (послуг), всієї вихідної від фірми інформації, її внутрішнього і зовнішнього оформлення.

Він потрібен для того, щоб зробити компанію більш презентабельною перед партнерами і клієнтами, а також підвищити впізнаваність фірми і рівень довіри. А це означає, що його створенню потрібно приділити особливу увагу, щоб в результаті сформувати необхідний для побудови і розвитку бізнесу інструмент. Фірмовий стиль дозволяє з ряду графічних знаків, форм, а також кольорів і відтінків сформувати цілісний образ бренду, який буде зберігати свою стилістичну єдність і органічний характер в інформаційному, медійному та фізичному просторі. У пакет послуг входить розробка: логотипу, візитки, бланк, конверта, папки.

Створення унікального фірмового стилю — це досить витратна процедура, однак вона того варта. Головними перевагами якісного фірмового стилю можна назвати:

- *Якісний фірмовий стиль — це «обличчя» вашого іміджу. Завдяки вдало підібраним кольорам і шрифтам ви можете підвищити не тільки впізнаваність свого бренду, але і лояльність*

цільової аудиторії, що, в кінцевому рахунку, призведе до підвищення продажів вашого товару або послуги;

- *Неповторний фірмовий стиль — це перший фактор, який допомагає вам виділитися з загальної маси конкурентів. З цього випливає, що чим краще стиль — тим краще «самопочуття» компанії;*
- *Фірмовий стиль потрібен абсолютно всім організаціям. Існує думка, що фірмовий стиль — це модна «заморочка», необхідна тільки великим корпораціям, але це не так;*
- *Завдяки фірмовому стилю можна виробити стійкий асоціативний ряд у споживачів. Наприклад, мікс червоного і білого кольорів — це Coca-Cola, з'єднання жовтого, білого і червоного — це McDonalds.*

Основні елементи фірмового стилю

1. Товарний знак

Товарний знак – це фірмове позначення, яке допомагає відрізнити товари / послуги / продукти одного підприємства від таких же товарів / послуг / продуктів конкурентів. Воно повинно бути офіційно зареєстровано в установленому порядку на юридичному рівні. Товарний знак може бути словесним, образотворчим, комбінованим і т.д.

2. Логотип



Досить часто логотип плутають з емблемою фірми. Потрібно розуміти, що логотип – це оригінальне зображення словесної написи скороченого або ж повної назви фірми. Іншими словами – це слово або словосполучення, написане в індивідуальному стилі фірмовим шрифтом.

3. Фірмовий блок

Це поєднання товарного знака та повного найменування підприємства (+ логотип). У нашому випадку сам логотип виступає фірмовим блоком. У приклад можна привести компанію Adidas:

Ця ж фірма часто використовує автономно логотип – три смужки, і товарний знак – окремий напис.

4. Гасло

Це якась фраза, яка передає клієнтам головну мету вашої компанії. Крім того, це може бути оригінальний девіз підприємства.



Під час складання гасла або девізу, дуже важливо врахувати своє позиціонування на ринку.

- *Досить часто компанії використовують якісь мотиваційні фрази в якості гасла і застосовують їх в тих же фірмових блоках. Ось так, наприклад, як це робить компанія Nike: Just do it*

Основні елементи:

- Логотип
- Фірмові кольори
- Фірмові шрифти
- Корпоративний герой
- Рекламне гасло (слоган)



Логотип



Фірмові кольори

Plast Lisovsky Medium

**АБВГГДЕЄЖЗИІЙКАМ
НОПРСТУФХЧШЩЬЮЯ
абвггдеежзиікмнопрстуфхчшщьюя
1234567890**



Фірмові шрифти

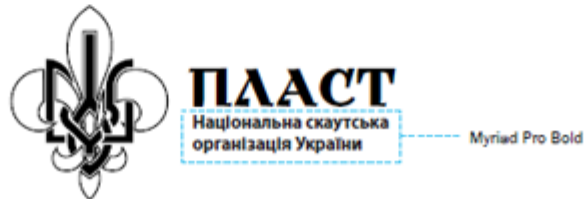
Допоміжний шрифт

Myriad Pro Bold

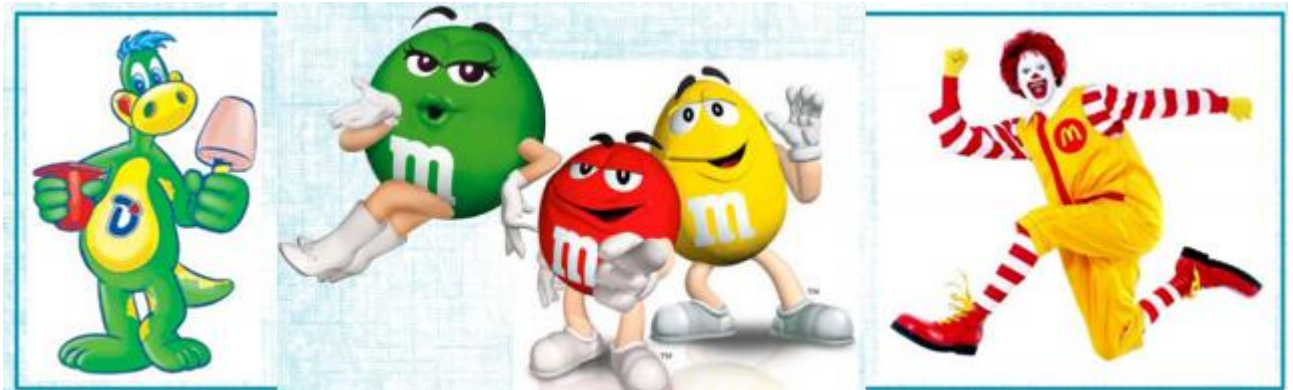
АБВГГДЕЄЖЗИІЙКЛМНОПРСТУФХЧШЩЬЮЯ
абвггдеежзіїклмнопрстуфхчшщьюя
1234567890

Myriad Pro Regular

АБВГГДЕЄЖЗИІЙКЛМНОПРСТУФХЧШЩЬЮЯ
абвггдеежзіїклмнопрстуфхчшщьюя
1234567890



Фірмові шрифти



Корпоративний герой

Корпоративний герой (реklamний персонаж) - особливий елемент фірмового стилю, який використовується для додавання марці людських рис, близький певній цільовій аудиторії. Корпоративні герої своїм виглядом і дією не тільки привертають увагу до марки, а й змушують споживача співпереживати їй, розглядати марку, як своє «друге я», частину своєї системи цінностей. Імідж корпоративного героя повинен відповідати потребам вибраної цільової аудиторії.



Слоган - це і візуальний, і аудіо образ компанії.

Брендбук (Brandbook) - видання, що містить в собі інформацію про фірмовий стиль компанії і правила його використання. Брендбук дозволяє **правильно використовувати фірмовий стиль не фахівцями.**

Розмір брендбука компанії може змінюватись — залежно від кількості розділів.

Ключовими можна вважати такі складові брендбука:

- Детальний опис позиціонування чи платформа бренду. Вона включає місію та цінності компанії, унікальну торгову пропозицію, емоційні та раціональні переваги.
- Портрет споживача – опис цільової аудиторії бренду.
- Стратегія просування бренду.
- Назва бренду: його значення, написання та вимова — якщо вона складна та неочевидна.
- Слоган або дескриптор, якщо є.
- Фірмовий знак, його допустимі та неприпустимі варіанти.
- Охоронне поле логотипу – мінімальна відстань від фірмового знака до інших елементів на макеті, необхідна для збереження читання.
- Фірмові шрифти та способи їх поєднання.
- Фірмові кольори та способи їх поєднання.

Гайдлайн чи брендбук?

Розробка фірмового стилю потребує досліджень, зусиль та часу. Щоб перетворити її результат на зручний практичний інструмент, складають гайдлайн. Це книга правил з використання елементів фірмового стилю.

Гайдлайн визначає (опціонально):

- розміри логотипу, правила його використання на різних носіях, прийнятні кольори;
- назви фірмових шрифтів, де та як їх можна застосовувати;
- назви всіх корпоративних кольорів, правила їх поєднання;
- як виглядає фірмовий блок: поєднання логотипу, слогану, поштової адреси, банківських реквізитів, контактів;
- якими мають бути банери в соціальних мережах, корпоративна поліграфія та рекламна продукція (макети та мокапи можуть додаватись).

Розумний сучасний тренд – мати гайдлайн у цифровому вигляді. Тоді ви миттєво можете надати його будь-якому новому підряднику.

В чому відмінність від брендбука? В останньому міститься значно більше інформації: не тільки про візуальні характеристики та іншу айдентику (тональність комунікацій, персонаж), а ще й про місію, легенду й цінності бренду, корпоративну етику, стратегії маркетингу тощо.

Вимоги до логотипу:

масштабованість - повинен добре виглядати в різних масштабах: і на придорожньому банері, і на авторучці;

кольоро-незалежність - повинен добре виглядати в кольоровому і чорно-білому варіанті;
незалежність від якості відтворення – повинен добре виглядати на екрані комп'ютера, в поліграфії, на бланку факсу;

компактність - повинен бути компактним, тобто вписуються в будь-який контекст;

впізнаваність і оригінальність;

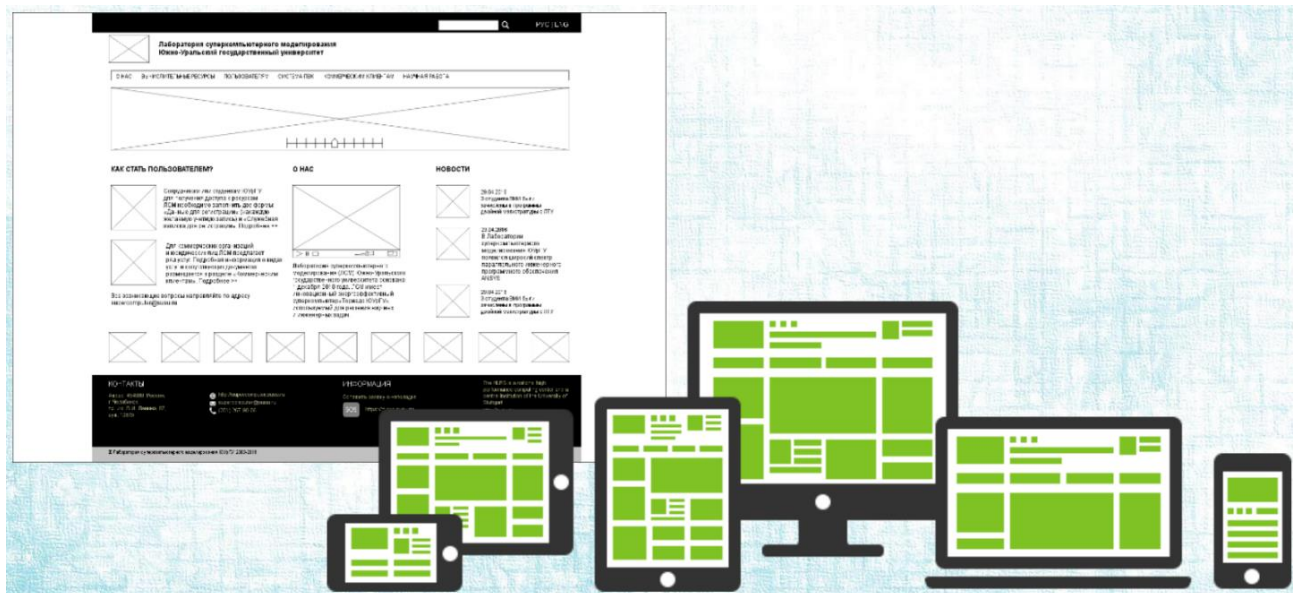
прихований сенс - «Фігура-фон» - спосіб досягнення прихованого сенсу;

ілюстративність - смислове навантаження знаку логотипу повинне співвідноситися з реальним об'єктом, якому цей знак належить, водночас не повинен бути буквальним.

Застосування двох і більше шрифтів можливо тільки в чисто текстових логотипах. У логотипах неприпустимі ні фотографіка, ні текстури, ні «скульптурна» тривимірність в силу

того, що складні текстири важко переносяться на папір і не витримують колірної редуції (редукція - процес або дія, що призводить до зменшення, послаблення або спрощення чогонебудь) .

3.3 Вимоги до оформлення макету: на папері, в текстовому чи графічному редакторі, веб-макету.



4. Графік робіт

5. Огляд сайтів за тематикою і сайтів конкурентів

Опису тільки самого замовником часто недостатньо, так як він, як правило, не має досвіду розробки сайту. Можливо, що в галузі, до якої належить компанія, є свої правила оформлення сайтів.

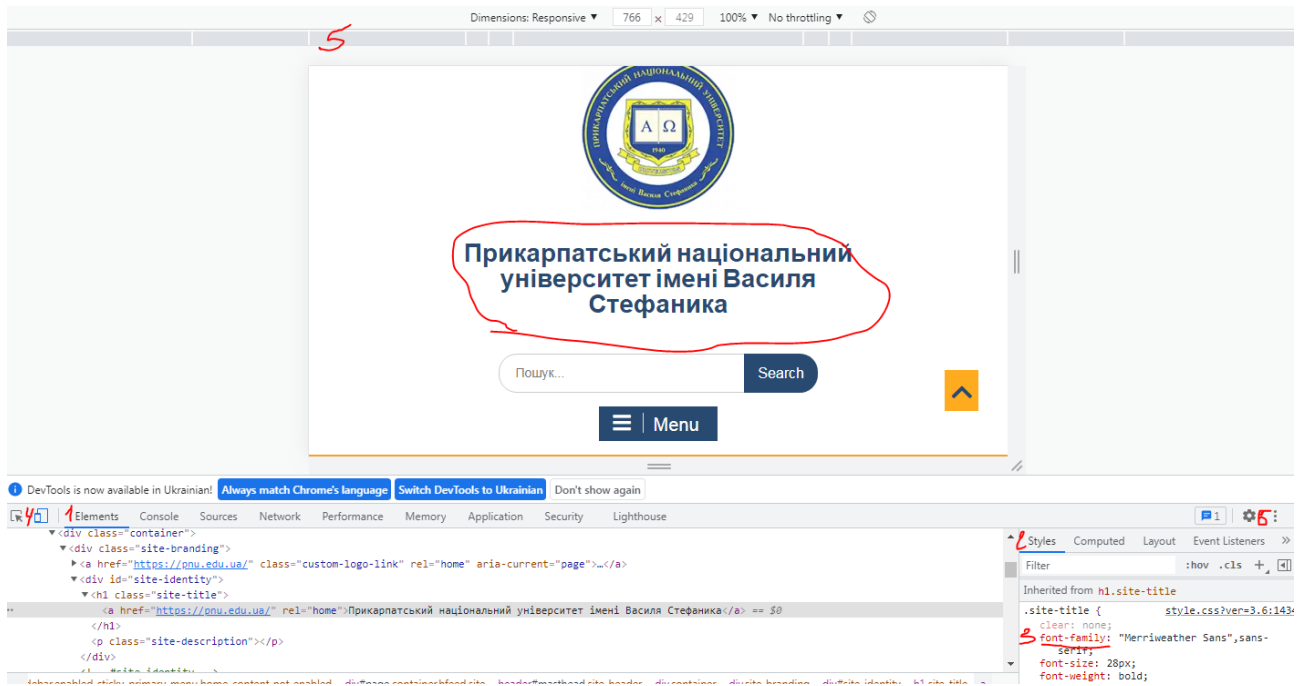
Потрібно:

- з'ясувати у Замовника його конкурентів.
- оцінити тип верстки сайтів, структуру, стиль дизайну, використовувані шрифти і колірні рішення, тощо.

Для дослідження можна скористатися панеллю DevTools браузера Google Chrome. Її можна викликати клікнувши на пункт "Перевірити" контекстного меню вікна браузера або F12. Для зміни розташування панелі DevTools, в її правому верхньому куті клікніть на вертикальні три крапки (6).

Вибравши елемент на веб-сторінці, у закладці Style (2) вкладки "Elements" (1) можна подивитися застосовані стилі, в тому числі назви шрифтів (властивість: font-family (3)).

Клікнувши на Toggle device toolbar (4) можна дослідити відображення елементів веб-сторінки на різних пристроях виведення: вверху над веб сторінкою появиться смуга із поділками для різних пристроїв виводу і розмірів екрану в пікселях (5). Клікніть на них, подивіться як змінюється розташування елементів на сторінці.



2. Розробка графічної концепції

Візуальне оформлення веб-сторінок за суттю подібне до поліграфічного дизайну для друкованого видання. Дизайн сайту є художньою «рамкою» для веб-контенту, але критерії якості сайту не завжди збігаються з критеріями якості поліграфічної продукції.

Дизайн сайту має суміщати художнє оформлення з ергономічністю. Проект повинен бути приємним на вигляд, легким для подальшої реалізації і зручним для використання. Вдалий дизайн повинен врахувати тематику і зміст ресурсу, вдало підкреслювати фірмовий стиль, і добре відтворювати загальну концепцію проекту.

Дизайн сайтів будується за принципом «зручно, стильно, просто», оскільки метою є створення ефективного ресурсу, який буде корисний і для власника, і для відвідувача. Досвідчений дизайнер повинен знаходити правильний баланс між красою і користю.

Функціональні стилі сайтів визначаються умовами використання сайтів.

Основні стилі:

- Мінімалізм
- Інформаційний дизайн
- Американський бізнес-стиль
- Промостиль

Основні риси **мінімалізму**:

- відмова від графіки (крім логотипу і дрібних значків);
- шрифтовий лаконізм (графіка і будь-які несистемні шрифти не використовуються);
- текстова навігація, використання «хлібних крихт» — навігаційний ланцюжок на сайті, який відображає шлях до сторінки, на якій знаходиться користувач, спочатку вони застосовувалися в операційних системах, щоб показувати маршрут до кореневої папки, але вже давно перекочували на веб-сайти;

- ретельна робота над контентом;
- чітка модульна сітка. Ні у кого не повинно виникнути питання: «А чому це розташоване саме тут?» чи «Яким чином це зверстано?».

Мета: надати інформацію, дизайн не для підвищення продажів.

Застосування стилю мінімалізм:

- ✓ сайти, де настільки багато текстової інформації, що для графіки не залишається місця.

Це онлайн - бібліотеки, каталоги;

✓ Інтернет-аукціони, біржі праці, онлайн-торги, тобто сайти, де не можна витратити час на завантаження «важких» сторінок.

✓ Оформлення окремих сторінок сайту, що містять графіки, таблиці і т. п. інформацію, що вимагає прозорості подачі, логічності і відсутності відволікаючого декору.

Інформаційний дизайн заснований на мінімалізмі, але не висуває таких жорстких вимог і завдяки цьому більш гнучкий і рухливий.

Основні риси **американського бізнес-стилю**:

- Фіксована ширина
- Стандартне розташування елементів: логотип в лівому верхньому кутку, меню зліва, тощо
- Звичні образи . Фотоматеріал, який частіше використовується в американському бізнес-стилі, не повинен вражати бурхливим креативом. Якщо сайт продає автомобіль, то в дизайні буде використана фотографія автомобіля в класичному ракурсі і найбільш прийнятному для потенційного покупця кольорі.

- Банерна реклама

Мета: продаж чого-небудь

Промостиль

- Дуже багато графіки . У промостилі не прийнято економити кілобайти і фарбу для пікселів.

- Використання яскравих образів і метафор . В результаті на сайт приємно дивитися, просто як на картину

- Графіка первинна, контент вторинний

- Переважно рекламний зміст сайту

- Промостиль передбачає щось таке, що відключило б мозок користувача сайту і надихнуло його на покупку товару або послуги

- Загальне «багатство» і «глянцевість»

- Відсутність сторонньої реклами

3. Вибір колірної схеми та художнього стилю.

Від вибору колірної гамми безпосередньо залежить його успішність і позитивне сприйняття відвідувачами.

При проектуванні сайту дизайнерові доводиться звертати увагу на правильний вибір кольору шрифту і кольору фону. *Рекомендується віддавати перевагу синім, зеленим чи чорним буквам на білому тлі, чорним на жовтому, білим на синьому. Помічено, що різні кольори по різному впливають на людину: червоний колір збуджує, а синій колір заспокоює.*

Дані психологічних досліджень говорять про те, що людське око сприймає червоний, помаранчевий, жовтий кольори і їх відтінки як теплі. Синій, фіолетовий кольори і їх відтінки сприймаються людиною як холодні. Це пов'язано, мабуть, з тим, що сонце і вогонь містять червоні і жовті кольори, а сніг, лід - блакитні і сині кольори.

Як правило, теплі тони на малюнку мають властивість "наближатися", а холодні - "віддалятися". Іншими словами, предмети, намальовані теплими тонами, будуть здаватися розташованими ближче предметів, зображених холодними тонами.

Фахівцями художньої фотографії встановлено, що око глядача притягується насамперед до світлих тонів знімка і до тих місць, де є найвищий тональний контраст.

Особливості впливу кольору на людину вивчає наука колористика.

Колірна схема має обиратися на підставі:

- Особливостей фірмового стилю. Оформлення сайту повинно відповідати його тематиці та фірмовому стилю компанії. Для випадку офіційного корпоративного сайту якоїсь установи не варто робити його строкатим і застосовувати дуже яскраві кольори, він повинен бути строгим, лаконічним. І навпаки, якщо сайт присвячено організації свят, то тьмяне оформлення навряд чи надихне потенційного клієнта.

- Фізіологічних і психологічних особливостей цільової аудиторії. Комфорт для відвідувача створює умови для позитивного сприйняття інформації. Кольори сайту не мають бути дратівливими і не напружувати очі. Потрібно забезпечити достатній контраст між фоном та

звичайним текстом, і не використовувати велику кількість кольорів. Вдало підібрані кольори задають бажаний настрій і сприяють запам'ятовуванню веб-ресурсу.

Колірна палітра

Колір в дизайні є доволі суб'єктивною категорією. В однієї людини він може викликати одні відчуття, а в іншої – абсолютно протилежні. Іноді сприйняття залежить від особистих переваг, іноді – від культурних традицій оточення. Елементарна зміна відтінку або насиченості кольору може спровокувати абсолютно несподівані емоції, тому, слід ретельно підбирати колірну палітру.

Найбільш поширеними кольорами у веб-дизайні є:

•Синій (блакитний) - один з найпопулярніших кольорів, що використовується в Інтернеті. Відноситься до категорії «безпечних» кольорів, який подобається більшості відвідувачів. Синій колір асоціюється з такими поняттями, як мир, спокій, надійність, довіра, чесність, чистота, ясність. Проте, ясно-блакитний колір може символізувати депресію.

•Зелений - завжди асоціюється з природою. Символізує оточення, здоров'я, долю, молодість. В деяких випадках може асоціюватися з недовір'ям і небезпекою.

•Жовтий - колір радості, щастя, сонця, дружби. Серед негативних асоціацій можна виділити боязкість і хворобу. Не варто використовувати дуже темний жовтий колір.

•Оранжевий - символізує теплоту і енергію. Хороший колір для акцентування уваги.

•Червоний - небезпека, кров, любов і тепло. Червоний колір добре привертає увагу.

•Фіолетовий - таємничий, творчий колір. Зазвичай, використовується для надання сайту особливої витонченості.

•Рожевий - найбільш жіночний колір. Асоціюється з дітьми і дитинством.

•Чорний - символізує силу, темноту і зло. Часто використовується як фоновий колір.

•Білий - чистий, добрий колір. Може асоціюватися з байдужістю і зимою.

Колірні комбінації

Фактори, які забезпечують гармонію :

- ретельно підібрані комбінації кольорів

- контраст по яскравості, насиченості і відтінку.

Відтінок (червоний, синій, жовтий і т.д.).

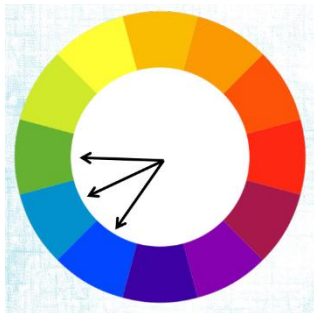
Яскравість кольору залежить від ступеня присутності в кольорі чорного або білого.

Насиченість кольору - показник, що визначає, наскільки колір "чистий", тобто містить або не містить домішки сірого.

Білий, чорний і сірий кольори називаються ахроматичними (тобто ті, що НЕ містять в собі колірних тонів (відтінків)), всі інші - хроматичними.

Ахроматичні відтінки поєднуються з будь-яким хроматичним. Якщо розбавити основний хроматичний колір білим або чорним, то отримаємо четвертий колір.

Монохромна колірна композиція будується на одному хроматичному кольорі і різних варіантах його яскравості.



Аналогова гармонія - це три відтінки, що йдуть один за одним на



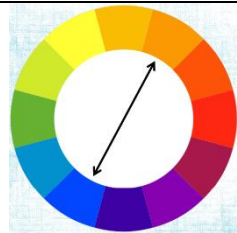
колірному колі.

Традиційна аналогова гармонія – містить недостатньо кольорового контрасту, необхідного для реалізації ефективного дизайну сайту.



Змінена гармонія з підвищеним рівнем кольоровості

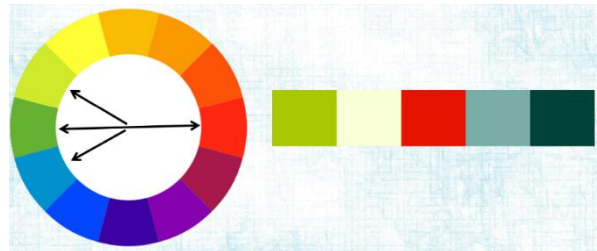




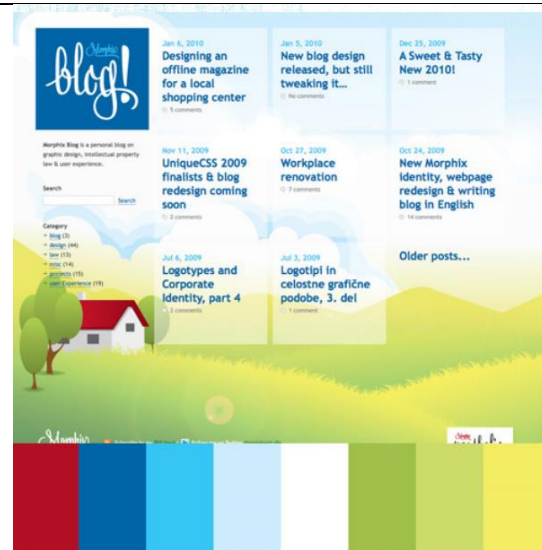
У
комплементарної
гармонії
знаходяться будь-які
два кольори,
розташовані один
навпроти одного на колірному колі.

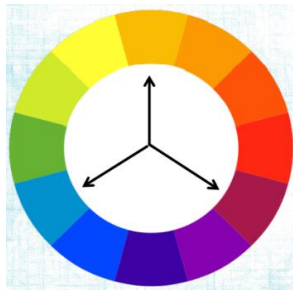


Якщо застосовувати тільки кольори,
розташовані на протилежних сторонах
з одним рівнем кольоровості або / і
рівнем яскравості, то можна отримати
дуже не привабливий і навіть
відлякуючий результат. Краще
унікати цього: або залишати простий
білий фон між цими кольорами, або
додавати між ними ще один
перехідний колір



Аналого-комплементарна гармонія -
синтез аналогової та комплементарної
гармонії. Суть її в тому, що до набору
кольорів з аналогової гармонії
приєднується колір, протилежний
середньому з аналогового набору

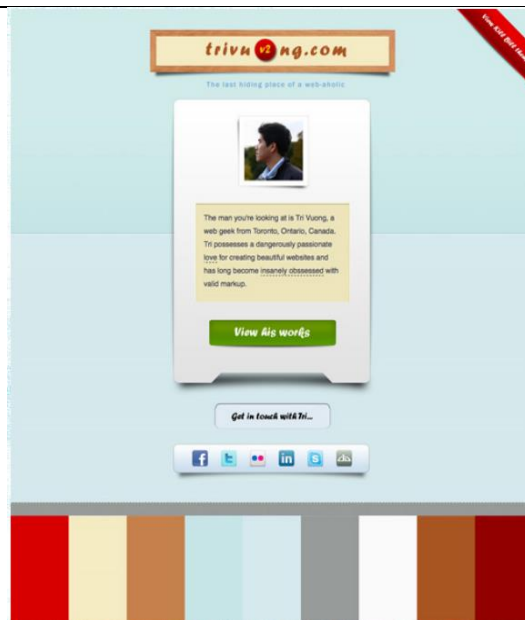




Триадна гармонія
Триадні кольорні
схеми створюються з
кольорів, рівномірно
розташованих на
колі кольорів. Вона є
однією з
найрізноманітніших.



Використовуючи більш блідий або темний
варіант кольору в триадній схемі, разом з
двома відтінками / освітленнями двох інших
кольорів, можна домогтися окремого
кольору, який буде грати роль нейтрального
кольору в схемі



Спеціально для дизайнерів створено багато онлайн інструментів, які дозволяють правильно підібрати кольори для дизайну сайту, ось деякі безкоштовні

- [Adobe Color](#)
- [ColorCode](#)
- [ColrD](#)
- [Cohensive Colors](#)
- [ColorHunter](#)
- [Tilda color](#)

4. Створення графічного макету Макет сторінки

Макет (Лейаут, Layout, Прототип) дозволяє створити цілісну картину елементів сторінки, єдність та відчуття неподільного образного ряду. Макет впорядковує структуру сторінки. Головна функція макету – визначити розташування всіх функціональних елементів сайту на сторінці.

Макет – візуальний скелет сторінки, він не містить ніяких декоративних елементів.

На етапі макету вирішуються наступні завдання:

- що головне, а що другорядне і як це показати;
- логічно розташовуються елементи сайту;
- чи зручно буде користувачеві.

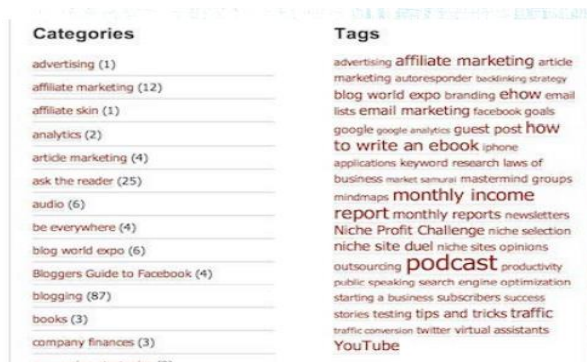
Макет включає:

- структуру сайту
- сітку сайту

Структура сайту

Блогова структура сайту. Особливість - в динаміці оновлення. У блогової структурі сайту весь контент представлений у вигляді статей з простим посторінковим виводом (пагінатором, Paginator).

Статті об'єднуються по категоріям, в залежності від контенту. Одна стаття може розташовуватися в



декількох категоріях. Категорії - динамічні структури, вони формуються в залежності від наявних статей.

Книжкова структура сайту: Статичні розділи підрозділами.



Книжкова структура сайту

Подання структури сайту у вигляді дерева – це найпростіший спосіб візуалізації структури сайту. Список сторінок в такому вигляді - є картою сайту.

Приклад деревоподібної структури:

Головна сторінка

- Новини
 - Новини за поточний місяць
 - Архів
- Про компанію
 - Історія
 - Керівництво
 - Досягнення і нагороди
- Продукція
 - Пили
 - Сокири
 - Молотки
- Контакти
- Відгуки покупців
 - Залишити відгук
 - Переглянути відгуки
- Карта сайту

Подання структури сайту у вигляді ментальної карти (mind-map).

«Mind map» (англ. карта думок) – структура у вигляді блоків, що демонструє зв'язок між елементами. Цей термін має безліч синонімів: асоціативна карта, діаграма зв'язків, карта знань, ментальна карта, інтелект-карта, карта думок, карта розуму.

Головні правила створення ментальної карти.

Для того, щоб діаграма зв'язків була максимально ефективною, необхідно керуватися декількома правилами:

- У центрі розташовується головна мета (проблема, ідея).
- Від центру виходять гілки з основними передбачуваними відповідями на питання, вирішення проблеми або розвиток ідеї.
- Кожна гілка теж може гілкуватися. Таким чином все вибудовується від головного до другорядного.

Замість фраз записуються тільки ключові слова, що несуть основне смислове навантаження.



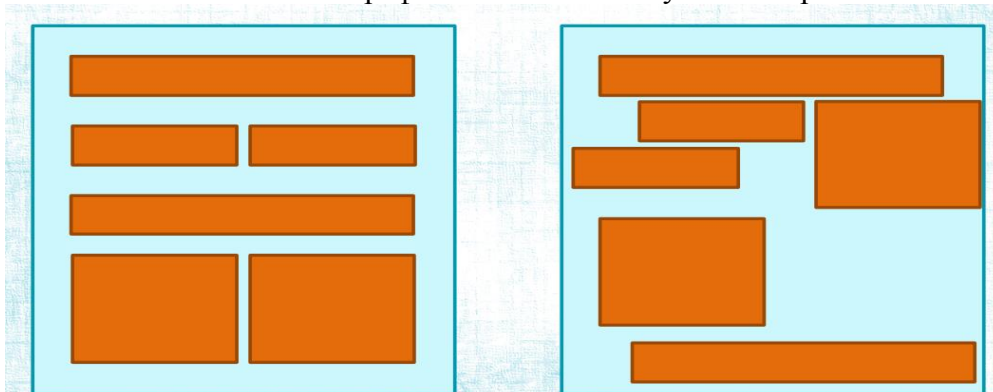
Приклад ментальної карти.

ТОП 10 програм для побудови mind-map

Дубльований контент - ситуація, коли на різних сторінках сайту зустрічається один і той же контент (текст або будь-яка інша інформація). Від дублів сторінок потрібно позбавлятися на самих ранніх стадіях створення веб сайту.

Сітка в веб-дизайні

Веб-сторінка фактично розглядається як набір прямокутних блоків, які складаються в певному порядку. Завдання макетування полягає у гармонійному розташуванні різноманітних текстових та графічних блоків майбутньої сторінки.



Основні елементи веб-сторінок:

- Логотип
- Назва сайту
- Меню
- Контент
- Інформація про розробників і власників сайту
- Контакти
- Лічильник відвідування

Елементи головної сторінки:

- Welcome-текст

- Анонси новин
- Короткий перелік послуг компанії або 3-4 найпопулярніші товари.
- Додаткові елементи: голосування, підписка на розсилку, календарі, блоки з контактною інформацією та ін.

Список елементів веб-сторінки формується на етапі ТЗ / брифа.

За сталими традиціями інформація розподіляється наступним чином:

В верхній частині сторінки (шапка, хедер, header) містяться: Логотип. Заголовок. Слоган. Телефон. Вибір мовної версії. Меню навігації (по сайту).

В середній (основній) частині сторінки: Меню навігації по тематичних розділах сайту. Основна інформація. Реклама, банери.

В нижній частині сторінки (підвал, футер, footer): Додаткове меню навігації (по сайту, по розділах сайту). Копірайти. Адреси, телефони. Лічильники і банери.

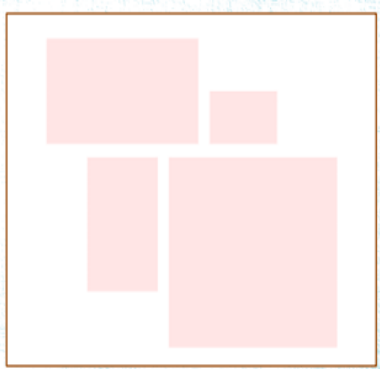
Сітка (Grid) вносить логічну структуру в сторінку за рахунок вирівнювання елементів. Сітка є набором невидимих ліній, вздовж яких розташовуються елементи сторінки. Це полегшує розміщення даних в документі, забезпечує візуальний зв'язок між окремими блоками і зберігає цілісність дизайну при переході від однієї сторінки до іншої. Використання сітки значно спрощує як проектування дизайну, так і подальшу верстку макету сайту, оскільки всі матеріали розбиваються на окремі блоки, які вирівнюються по невидимих напрямних лініях. Такі блоки хоч і взаємозв'язані між собою, але дозволяють незалежне оформлення елементів.

Види сіток для веб-дизайну:

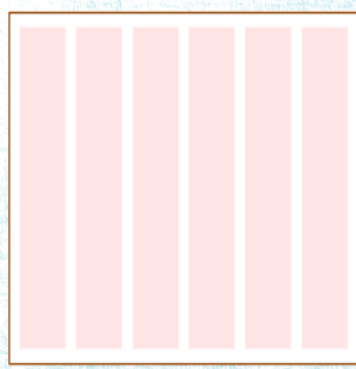
блочна сітка – розмітка площі по блоках;

колоночна сітка - в своїй структурі має колонки;

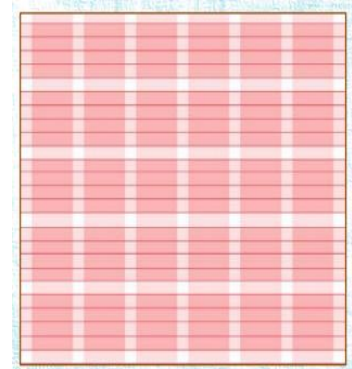
модульна сітка - складається з пересічних прямих, які утворюють модулі.



Блочна сітка



Колоночна сітка



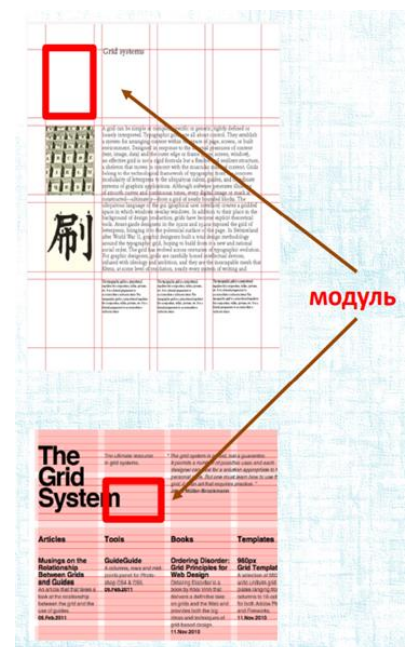
Модульна сітка

Модульна сітка

Модульна сітка являє собою таблицю з осередків, де один з них взятий за одиницю виміру (модуль), а інші рівні або кратні йому.

Модуль лежить в основі композиції сторінки (друкованої і веб). Модульна сітка визначає, де будуть знаходитися основні елементи сторінки.

Побудову модульної сітки починаємо з визначення розмірів робочої області.



Потім потрібно зробити шрифтову сітку. Для цього вибираємо висоту рядка, яка повинна бути єдиною для всього макета. В елементах, які мають кегль відмінний від кегля основного тексту, міжрядковий пробіл повинен бути кратним нашій висоті рядка.

Таким чином, ми отримуємо основу нашої майбутньої сітки, на яку буде спиратися весь текст.

Тепер нам потрібно вирішити, якої ширини буде наш модуль. Тут ми можемо відштовхнутися від ширини будь-якого постійного елемента. Також потрібно визначити, яким буде відстань між модулями. Вона повинна бути рівною мінімум одній висоті рядки з шрифтової сітки.

Важливо правильно вибрати розміри модуля, адже якщо вони будуть занадто великими, то пропаде гнучкість сітки, а якщо ж вони будуть дрібними, то в сітці не буде сенсу.

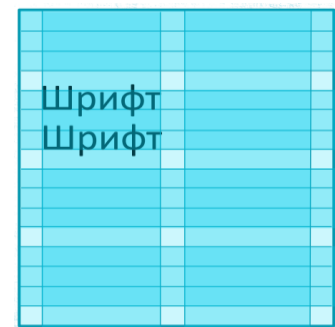
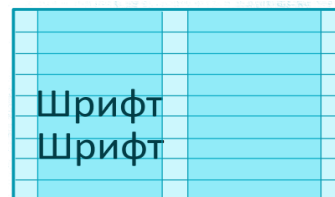
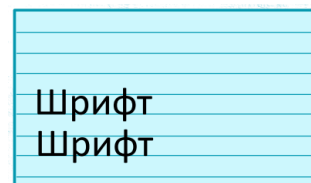
Горизонтальне членування зробити досить просто: його висота повинна бути кратна висоті рядка.

Далі залишилося тільки об'єднати модулі в регіони і створити композицію.

Існують багато онлайн-сервісів, які допомагають розрахувати та створити сітку сайту, наприклад [цей](#). Також існує так звана стандартна 12-ти колоночна bootstrap-сітка. Її опції для адаптивної верстки сайтів з допомогою найпопулярнішого CSS фреймворка [Bootstrap](#) можна подивитись [тут](#).

Варіанти сітки [Bootstrap 5.0](#) для різних пристроїв виводу.

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					



Gutter width	1.5rem (.75rem on left and right), 1 rem=16 px
Custom gutters	Yes
Nestable	Yes
Column ordering	Yes

Маючи конкретні розміри, сітку можна створити з допомогою макету напрямних ліній. У Photopea: “Вікно-Більше-Guide Guy”; у Photoshop: “Перегляд-Новий макет напрямних”, і вказати потрібну кількість колонок, рядків, їх ширину, відступи, тощо. Ctrl + ; - відобразити/приховати напрямні. Модульну сітку можна зробити один раз, а потім використовувати її (їх, якщо сіток кілька для різних пристроїв виводу) для різних проектів. Часто можна зустріти сайти, для яких важко або взагалі неможливо визначити застосовану сітку. Такі проекти виглядають креативнішими і цікавішими, але водночас і складнішими для верстки. Використання CSS фреймворка [Bootstrap](#) для їх верстки часто є неможливим. Останнім часом все популярнішою стає верстка з допомогою [CSS Grid](#) (двовимірна система сіток в CSS), оскільки вона є більш гнучкою і дозволяє верстати сайти з нестандартною розміткою. В мережі Інтернет можна знайти багато матеріалів, чому для створення веб-макету CSS Grid є кращий ніж Bootstrap.

Перш ніж почати створення макету, необхідно визначити кількість принципово різних сторінок у майбутньому сайті:

- **1 варіант:** одна сітка для головної і внутрішніх сторінок. Часто робиться на новинних сайтах і великих порталах.

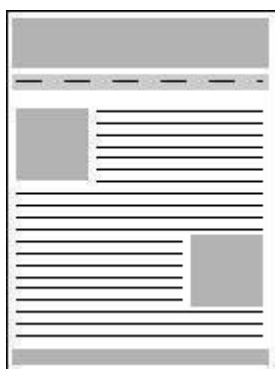
- **2 варіант:** сітка внутрішніх сторінок – спрощений варіант сітки головної сторінки. Тоді створення макету для сайту зводиться до планування головної сторінки і єдиного шаблону, за яким буде створено всі внутрішні сторінки. Всі сторінки майбутнього сайту повинні бути виконані в єдиному стилі, хоча компоновка текстової області на головній сторінці може бути дещо складнішою, ніж на внутрішніх. Для такого сайту створюють, як правило, два-три макети.

- **3 варіант:** для кожної сторінки своя сітка. Потрібно, якщо на різних сторінках розташовується різнорідна інформація, наприклад, форум, фотогалерея, чат та ін. Тоді, звісно потрібно створення кількох макетів для одного сайту, причому стилістично схожих між собою.

Кількість сіток для адаптивного дизайну сайту: к-сть сіток * к-сть різних пристроїв.

- 1) Одна сітка
- 2) Дві сітки
- 3) Сітка для кожної сторінки

- 1) Монітор
 - 2) Планшет з альбомною орієнтацією
 - 3) Планшет з книжковою орієнтацією
 - 4) Смартфон
- тощо



Зазвичай, при формуванні структури сторінки дані візуально розташовуються по колонках, тому вживають терміни одно, двох, трьохколонна і т.д. структури.

Одноколонна структура.

Структура в одну колонку найчастіше зустрічається в академічному дизайні, при фіксований ширині сторінки і публікації великого об'єму тексту. Академічний дизайн характеризується мінімалізмом оформлення і навіть аскетизмом. Основна увага приділяється до змістовної частини, а не до дизайну. Переважно

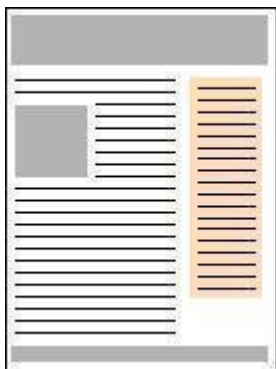
академічний дизайн зустрічається в науковому середовищі чи державницьких інституціях, наприклад <https://zakon.rada.gov.ua/laws/show/2145-19#Text>

У типовій схемі одноколонної структури, як правило, спостерігається чотири основні блоки:

- Заголовок сторінки.
- Набір посилань на інші сторінки сайту (навігація).
- Власне сам текст (контент).
- Контактна інформація в нижній частині сторінки.

Якщо висота сторінки є достатньо великою, то навігацію дублюють внизу або роблять посилання «Догори», що пересуває сторінку на початок. Ілюстрації на сторінці застосовують за змістом тексту, і, зазвичай, текст обтікає їх за контуром.

Двохколонна структура



Це один з найпоширеніших варіантів при використанні на сайтах. За такою структурою, зазвичай, лишаються незмінними верхній та нижній блоки сторінки, а основний (середній) блок поділяється на дві колонки, що відводяться для:

- Основного тексту.
- Навігації чи іншої корисної інформації.

Принципового значення не має, зліва чи справа розташовується колонка з навігацією, популярними є обидва варіанти.

Двохколонна структура стала стандартом де-факто для інформаційних сайтів завдяки своїй зручності. Дійсно, все «під рукою»: і текст і посилання. Двохколонні сітки є зручними для створення і не вимагають особливих знань з верстки веб-сторінок.

Трьохколонна структура



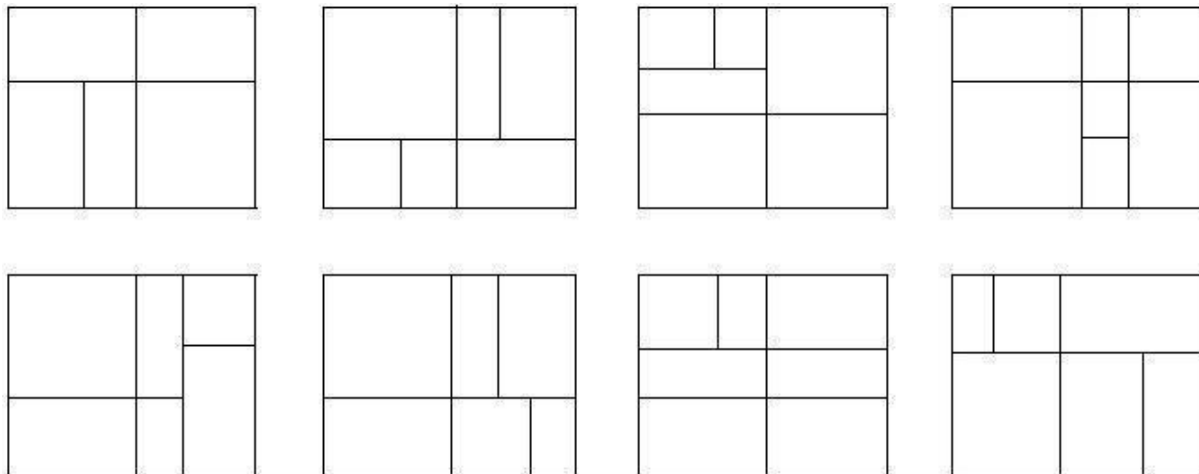
Такі структури часто застосовуються на головних сторінках сайтів, де потрібно одночасно показати відвідувачу багато можливостей, які присутні на даному сайті.

Також трьохколонна структура використовується і на внутрішніх сторінках, якщо двох колонок бракує для розміщення різної інформації. Одна з колонок містить навігацію, друга, найширша під основний текст, а у третій колонці розміщують рекламу, посилання, поради тощо.

Трьохколонна структура забезпечує більше простору для дизайну, адже в деяких місцях можна об'єднувати колонки, розбивати матеріал на окремі фрагменти і візуально відокремлювати один блок від іншого.

Макет при цьому може вийти достатньо складним, але результат того вартий. Якщо представити основні блоки сторінки у вигляді прямокутників, то отримаємо наочну модульну сітку, з якої відразу видно як складено сторінку.

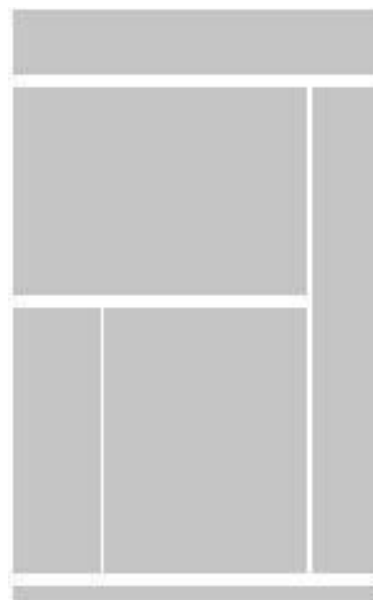
Звісно, що можна застосувати інші види модульних сіток, в тому числі і з більшою кількістю колонок. Хоча вони виглядають складними, сайти на їх основі будуть оригінальними і незвичними.



Модульна сітка не є єдиним варіантом при створенні структури сторінки. Можна скористатися перспективою, хаотичністю або іншою структурою, головне, щоб вона була естетичною і привабливою.

Відмова від загальноприйнятих правил структурування є характерною для дизайнерів, які прагнуть новизни, хочуть вразити уяву відвідувачів і, можливо, закласти нові модні течії. Але, навіть вони роблять для замовника сайти на основі формальних сіток, оскільки так простіше додавати інформацію і вносити різноманітність до дизайну макету.

Для прикладу розглянемо головну сторінку певного сайту.



Кожен блок цієї сторінки чітко відокремлено від інших за допомогою вільного простору, рамки або роздільника.

Тут важко розібратися, скільки колонок містить макет, оскільки колонки можуть об'єднуватися, а також містити не лише суцільний текст, але і графічні вставки.

Графіка на web-сторінці ділиться на виставкову графіку, фони і візуал.

Виставкова графіка - це зображення, які розміщуються на сайті не з якоюсь службовою або декоративною метою, а заради них же самих. Наприклад, фотографії для фотогалереї, скріншоти сторінок сайту для портфоліо і т.п.

Фони застосовні як до web-сторінки в цілому, так і до її окремих елементів (меню, шапки сайту і т.д.). Використання фонових зображень значно скорочує час завантаження сторінки, оскільки дозволяє зображенням невеликого розміру заповнити всю необхідну площу певного блоку за рахунок його повторення.

Візуал - щось середнє між «Декоративною плямою», тематичної ілюстрацією і емблемою. Візуал на головній сторінці:

- ілюструє основну сферу діяльності фірми, метафорично відображає її назву або рекламне гасло
- ілюструє текст - останні новини, прес-релізи, огляди і т. п.

Джерела візуалу: фотографії, мальована графіка.

«Нелегальні» методи отримання візуалу: взяти картинку з іншого сайту, в пошукових системах, відсканувати з журналу, «вкрадені» платні кліпарті і т.д.

«Легальні» методи отримання візуалу: безкоштовні кліпарты, послуги фотографа або ілюстратора, самостійне створення фотографій, ілюстрацій, колажів.

Кліпарты - це фотографії чи ілюстрації, які знаходять в спеціальних сховищах (фотобанках) і використовуються безкоштовно чи ні.

Інфографіка – це графічний спосіб подачі інформації, даних і знань.



Макет (прототип)

Отже, після закінчення формування брифу, можна починати створювати блочну структуру макета сайту, тобто прототип. Це схематичний малюнок, який показує розташування основних елементів: хедер, футер, блоки контенту на сторінці, орієнтовне співвідношення їх розмірів. Єдиного стандарту щодо формування прототипу немає, його попередню версію можна зобразити як на листку паперу, так і відразу в графічному редакторі із врахуванням необхідної сітки. Також може бути спочатку створений орієнтовний веб-макет з допомогою засобів HTML і CSS, а на його основі вже розробляти ескіз чи фінальний варіант дизайну сайту, наприклад у форматі PSD.



Макет (прототип) сайту на різних пристроях

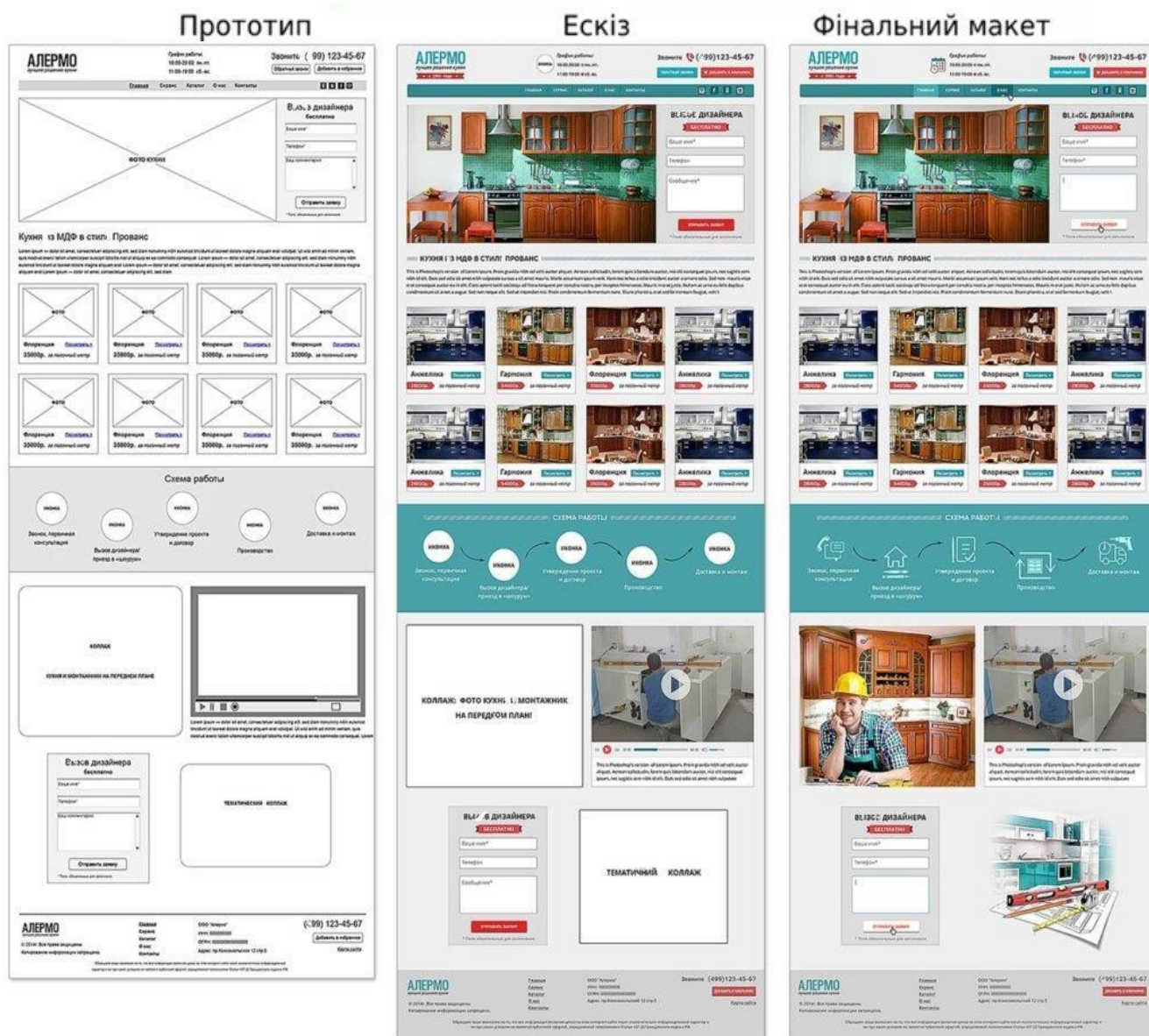
Ескіз web-сторінки

Ескіз web-сторінки створюється на базі брифа/ТЗ, фірмового стилю, сітки сайту і прототипу web-сторінки. Ескіз створюється в графічному редакторі, він являє собою проміжний варіант дизайну між прототипом і фінальною версією шаблону дизайну веб-сторінки. По структурі і наповненню він максимально наближений до фінальної версії дизайну, але не містить складних декоративних елементів.

Шаблон (template, фінальний макет, графічний шаблон) web-сторінки

Шаблон web-сторінки створюється на базі брифа/ТЗ, фірмового стилю, модульної сітки, лейауту і ескізу веб-сторінки. Шаблон сайту - це його реалістичний прототип, створений в графічній програмі на кшталт Photoshop, Photorea, figma.com. Він виглядає як готовий сайт: з усіма блоками тексту і зображень, кнопками, фоновими малюнками і так далі. Єдина відмінність в тому, що в шаблоні може не бути функціонального наповнення: наприклад, сторінка з товарами заповнена шаблонами описів. “Lorem” + Tab у VS Code дозволяє

створити текстовий контент. [Тут](#) можете переглянути і проаналізувати готові PSD-шаблони (templates). Детальніше про створення шаблонів буде розглянуто в наступному лекційному матеріалі.



На практиці, межа між цими поняттями дуже розмита, для виконавця і замовника вони можуть мати різний сенс. Тому, якщо ви використовуєте ці слова у ТЗ, просто домовтеся про їхній зміст і переконайтеся, що всі їх розуміють однаково.

Позиціонування сторінки у вікні браузера

Веб-сайт є електронним аналогом друкованого видання. Звісно, що інформація в будь-якому складному документі має бути впорядкованою, і тут впливає основна відмінність між веб-сторінкою і сторінкою в журналі. Будь-яка поліграфія на зразок буклету, листівки або брошури друкується на листах встановленого розміру і в межах одного тиражу має лише незначні або непомітні оку відмінності.

Особливістю веб-документу є відображення в браузерах на численних гаджетах користувачів. Зрозуміло, що це надвелика кількість різноманітних екранів, операційних систем з різними налаштуваннями і браузерів. Отже, дизайнеру слід враховувати, що одна і та ж сторінка сайту буде по-різному відображатися в кожного користувача.

Для однакового відображення сторінки потрібно взяти до уваги багато різних неоднозначних чинників, що є доволі складною задачею. Тому, завдання створення веб-

сторінок формулюється так: **сформувати документ, який би з невеликими відмінностями коректно відображався на основних платформах і в популярних браузерях.**

Існують певні особливості, які варто враховувати при верстанні сторінки.

Ширина документа

Визначення ширини майбутньої сторінки є доволі важливою і непростою задачею, бо ширина вікна браузера користувачів заздалегідь є невідомою і може мінятися в найширших межах. Ширина вікна браузера залежить від роздільної здатності екрану, довжини його діагоналі та інших чинників. З урахуванням цієї особливості дизайнери застосовують кілька варіантів вибору ширини сторінки та позиціонування у вікні браузера.

Фіксований дизайн

«Фіксованим» називають дизайн, коли ширина інформативної частини сторінки є визначеною величиною в пікселях і не залежить від роздільної здатності монітора. Найчастіше при розробці фіксованого дизайну орієнтуються на найменшу роздільну здатність сучасних моніторів. Самі перші сайти орієнтувалися на роздільну здатність 640x480, згодом 800x600, але тепер переважна частина моніторів мають екрани з мінімальною роздільною здатністю 1024x768 пікселів, хоча, звичайно існують екрани із значно більшими показниками.

Для фіксованого дизайну важливим є вибір позиціонування сторінки. Вона може бути притиснутою до лівого краю або розташовуватися по центру вікна. Сторінка, що вирівняна по центру виглядає більш традиційно і класично. Несиметрична сторінка, що виходить в результаті її зсуву в ліву частину, додає сайту оригінальності і незвичності.

Переваги. Простота верстання документа. Після оптимізації сторінки для відображення з роздільною здатністю 1024x768 точок, можна бути впевненим, що при зміні екранних налаштувань елементи дизайну не будуть зсунутими. Даний варіант компоновки сайту в більшості випадків дозволяє ідентично відображати сторінки в різних браузерах.

Недоліки. Важко передбачити екранні налаштування користувача. При відображенні документа на комп'ютері зі значно вищою екранною роздільною здатністю з країв екрану або з правого його боку залишається помітне порожнє поле. Але більшою вадю є коли роздільна здатність екрану є меншою за ширину сайту. В цьому випадку, внизу вікна з'являється горизонтальна прокрутка, що є **неприпустимим** з точки зору професійного дизайну.

Гумовий (еластичний) дизайн

«Гумовим» називають дизайн, коли сторінка розгортається на весь екран по ширині незалежно від встановленої роздільної здатності монітора. Наприклад, і при роздільній здатності 1024*748 і при 1600*1200 сторінка займатиме всю ширину екрану.

Цей вид макету використовує в якості одиниць вимірювання відсотки. Загальна робоча ширина вікна браузера 100%, і колонки макету в сумі не повинні її перевищувати, тому, для зручності, як правило, скрізь застосовують запис у відсотках. При зміні розмірів вікна відбувається перерозподіл вмісту сторінки відповідно до нової ширини.

Переваги.

- Ефективно використовується вся площа екрану. Сторінка розтягується по всій ширині екрану, не лишається невикористаних порожніх полів.

- Дуже вдалий вибір для сайту, що містить багато інформації.

Недоліки.

- Складність верстання сторінки, доводиться враховувати багато чинників і знати певні прийоми верстки.

- Потребує ретельного тестування для адекватного відображення при різних роздільних здатностях і в різних браузерах.

- Спостерігається ефект зсування різних блоків сторінки. При надто високій роздільній здатності абзац тексту може перетворитися на один довгий рядок, що є незручним для сприйняття інформації.

Комбінований дизайн

Існує так званий «псевдо-гумовий» дизайн: ширина сторінки є фіксованою, але візуально сторінка якби заповнює весь екран. Такий ефект досягається за допомогою спеціальних дизайнерських прийомів - продовження верхніх і нижніх частин сторінки в обидві сторони,

використання фонового зображення, та багато чого, що робить перехід від інформаційної частини сторінки до незаповненої менш різким і помітним. Але, ці прийоми не міняють суті фіксованого дизайну.

Адаптивний дизайн – розмір, відображення і розміщення блоків адаптується під роздільну здатність пристрою виводу. Блоки можуть приховуватися, змінювати місце розташування чи замінюватися іншими. Сучасні методи каскадних таблиць стилів CSS дозволяють встановити максимальну та/або мінімальну ширину екрану для відображення того чи іншого макету сторінки. Наприклад, одна і та сама сторінка на екранах різної роздільної здатності буде відображатися з різною модульною сіткою.

Який дизайн застосувати

Отже, для того щоб обрати доцільний варіант, варто проаналізувати об'єм інформації, який передбачається розміщувати на кожній сторінці та вимоги до зовнішнього вигляду цієї інформації.

Якщо інформації багато і принципово важливою є саме інформація, а не вигляд її представлення, тоді, вибирається гнучкий дизайн сторінки. Як правило, такий тип дизайну підходить для інформаційних порталів, новинних і подібних до них сайтів. Для таких сайтів важливо розмістити на екрані якомога більше інформації, а естетика є другорядною справою.

Якщо ж сайт є більш презентаційним і має порівняно менше інформації або для дизайнера важливим є точне розташування основних блоків за його задумом, тоді, перевага віддається адаптивному дизайну сторінки. Найчастіше на основі адаптивного дизайну створюють сайти для презентації компаній чи приватних осіб. Тут, зазвичай, текст щедро розбавлено графічними блоками, що ілюструють даний фрагмент тексту.

Зустрічаються верстки, коли один і той самий сайт має різні доменні адреси для його перегляду на різних пристроїв виводу. Наприклад <https://www.facebook.com/> і <https://m.facebook.com/>

Висота документа

Історично склалося, що перегортання довгого документа на комп'ютері відбувається зверху вниз. Для перегортання призначено вертикальні смуги прокрутки, клавіатурні комбінації, скролінг мишки. А ось переміщення по горизонталі відбувається не так зручно, тому, горизонтальної смуги прокрутки бути не повинно. Отже, сторінка повинна вписуватися у вікно браузера за шириною, а висота може змінюватися в дуже широкому діапазоні.

Але, чим більше на сторінці інформації, і, відповідно, висота документа більшою, тим складніше знаходити потрібні дані. Тому, текст структурують, розбивають на блоки і кожному з них надають заголовок, щоб користувач вільно орієнтувався на сторінці.

Також слід врахувати, що об'єми текстів на сторінках сайту можуть достатньо сильно різнитися, і відповідно, буде мінятися і висота сторінки. Тому, слід заздалегідь потурбуватися про те, щоб макет відображався без помилок за різних значень висоти.

Особливості макету для подальшої верстки

- Врахування масштабованості: як буде виглядати сторінка при різних розмірах вікна.
- Врахування подальшого розвитку сайту в майбутньому, наприклад, збільшення пунктів меню або зовнішньої реклами.
- Можливість реалізації дизайнерських задумів при верстанні: складне фонове зображення, накладання блоків, випадне меню, тіні, заокругленості, анімація тощо.

5. Юзабіліті

Юзабіліті - це сукупність зручності, дружності та ефективності використання. При розробці дизайну потрібно врахувати, щоб користувачеві було комфортно на сайті, щоб він не відчував труднощів з навігацією та інтерфейсом, все повинно бути чітко, зрозуміло, логічно і інформативно. Основні принципи юзабіліті регламентують, що сайт повинен бути простим у навчанні, ефективним у використанні, вільно запам'ятовувався для повернення постійних відвідувачів і задовольняв їх інтерес.

Навігація по сайту

Важливою складовою сторінки є зручна і зрозуміла для користувача навігація. В ідеалі користувач, потрапивши на будь-яку із сторінок сайту, повинен відразу зорієнтуватися, де він знаходиться, і куди йому рухатися далі.

Система навігації повинна бути єдиною для всіх сторінок сайту. Вдала навігація є передбачуваною і дозволяє користувачам відчувати себе впевнено при переміщенні по сторінках сайту. Їм не потрібно нічого вивчати або запам'ятовувати, у продуманому дизайні все осмислено і впорядковано і не доводиться сумніватися, де шукати елементи сайту.

Значна зміна елементів навігації при переході від однієї сторінки сайту до наступної дезорієнтує користувача і відволікає його увагу.

Посилання на головну сторінку

Посилання до головної сторінки має бути присутнім на всіх сторінках сайту. Загальноприйнятим правилом є додавання до логотипу сайту посилання на головну сторінку.

Меню (навігаційні панелі)

Меню є основним функціональним елементом сайту, що складається з набору посилань (пунктів меню) на різні сторінки. Бажано мати на кожній сторінці навігаційну панель, що містить посилання на основні розділи сайту (і, можливо, на підрозділи поточного розділу).

Найбільш поширеними є два варіанти розміщення навігаційної панелі: горизонтальним рядком у верхній частині сторінки та вертикально в лівій чи правій колонці. Часто можна зустріти і комбінацію цих способів, коли для організації навігаційної панелі використовується і бічна колонка, і верхній рядок. Логічним є розміщення посилання на важливі розділи у верхньому меню.

Головне меню сайту містить посилання на основні розділи сайту. Пункти головного меню повинні бути максимально короткими, в ідеалі – однослівними. Число пунктів в головному меню, їх назва і порядок не повинні мінятися при перегортанні сторінок. Добре було б певним чином виділити пункт, що відповідає відкритій сторінці. Це спрощує орієнтацію користувача у сайті, оскільки він бачить однотипне меню на всіх сторінках і одночасно розуміє, де він в даний момент знаходиться.

Якщо сторінка є достатньо об'ємною і для її перегляду потрібне перегортання, тоді головне меню бажано продублювати наприкінці сторінки. Додатково, там можна розташувати навігаційне посилання «догори» на початок сторінки.

Додаткове меню сайту, як правило, має ієрархічну структуру і буває 3-х видів:

- Випадаюче меню. Автоматично з'являється при наведенні вказівника на пункт меню.
- Меню, що розгортається. При натисненні на пункті меню розкривається перелік.
- Розкрите меню. Статичне меню, де відразу відображено всі пункти, часто має деревовидну структуру.

Меню ієрархії документів

Досить зручним для відвідувачів сайту є зазначення місця сторінки в структурі сайту. Це можна реалізувати рядком у верхній частині сторінки приблизно такого вигляду:

Головна -> Розділ -> Підрозділ -> Назва поточної сторінки

При цьому всі старші пункти ланцюжка оформлюються як посилання. Особливо корисною така підказка буде у випадку, якщо сайт має більше трьох рівнів вкладеності.

Меню послідовної навігації

Якщо сторінка містить великий об'єм тексту (наприклад, книгу), краще роздібнити цей об'єм на менші логічні частини. Тоді, після, а іноді і перед текстом виводиться меню послідовної навігації приблизно такого вигляду:

попередня сторінка * зміст * наступна сторінка

Іноді замість послідовної навігації застосовують пряму навігацію:

Частина 1 | Частина 2 | Частина 3 | Частина 4 | Частина 5 | Частина 6

Гіперпосилання

Гіперпосилання — це зв'язок між сторінками або файлами. При натисненні на гіперпосилання в браузері відкривається або запускається вказаний в ньому об'єкт. Переважно гіперпосилання вказує на іншу сторінку, але за його допомогою можна

відобразити збільшений малюнок, запустити мультимедійний файл, завантажити файл чи програму, запустити поштовий клієнт для відправлення листа за вказаною адресою електронної пошти.

За об'єктом, на який посилаються, розрізняють:

- Зовнішнє гіперпосилання вказує на об'єкт, що розташований поза даним сайтом.
- Внутрішнє гіперпосилання вказує на об'єкт, що розташований на даному сайту.
- Непрацююче (зіпсоване) гіперпосилання вказує на недоступний або неіснуючий об'єкт.

За виконанням гіперпосилання бувають:

•Текстовим гіперпосиланням називається слово або фрагмент тексту, якому призначено адресу іншого об'єкта. За замовчуванням, текстові гіперпосилання є підкресленими і відображаються іншим кольором.

•Графічні гіперпосилання (кнопки) – це картинка, якій призначено адресу іншого об'єкта.

При наведенні вказівника мишки на текстове чи графічне гіперпосилання, воно змінює свій вигляд. За замовченням - це іконка руки, яка є стандартною і звичною для всіх користувачів Інтернету. Іноді, заради новизни чи в плані експерименту дизайнери можуть змінити стандартне зображення на інше і збентежити відвідувачів сайту. Чи варто наголошувати, що подібних «експериментів» слід уникати.

Інтерактивність гіперпосилань

Часто до гіперпосилань застосовують елементи інтерактивності. Вони відображають реакцію гіперпосилання на різні події, найчастіше на наведення користувачем вказівника мишки та на натиснення (вибір).

Для текстових гіперпосилань – це зміна кольору, підкреслення, розміру. Дизайнери мають можливість вибрати чотири кольори, для відображення різних станів гіперпосилань:

- Гіперпосилання, що досі не використовувалося.
- Гіперпосилання, на яке наведено вказівник мишки.
- Активне гіперпосилання в момент натискання на нього.

•Переглянуте гіперпосилання, за яким вже було здійснено перехід. Такий колір браузер може зберігати відповідно до терміну зберігання журналу переглянутих сторінок, за замовчуванням цей термін складає 20 днів.

Для графічних гіперпосилань – це заміна зображення елемента іншим зображенням, з певним ефектом (зсув, підсвічування, тінь тощо).

Можна змінювати вигляд кнопки після завантаження сторінки, при наведенні на неї вказівника мишки, при натисненні вказівником мишки. Відповідно, для такої інтерактивної кнопки дизайнер має підготувати відповідно 3 різні картини.

При створенні макета дизайну сайту дизайнер повинен спланувати поведінку передбачених інтерактивних елементів на дію користувача. Наприклад, при створенні посилання важливо вказати як посилання буде виглядати в різних станах (звичайному стані, стан при наведеному вказівнику мишки, стан після відвідування сторінки). Зазвичай, посилання помічають підкресленням, іншим кольором, або товщиною. Всі елементи навігації, посилання, меню, форми тощо необхідно продумувати, щоб відвідувач розумів, що здійснює якусь дію.

Типографіка

Під юзабіліті для відвідувача, зазвичай, розуміють зручність інтерфейсів, навігації, дизайну та інших візуальних об'єктів на сайті. Але лівова частка контенту, який потрібен користувачу - це текст: опис товарів, способи замовлення послуги, порівняння марок тощо. Багато користувачів не просто шукають в Інтернеті, а порівнюють, читають аналітичні матеріали та статті на сайтах, тому для них потрібно забезпечити максимальний комфорт і читабельність.

Нечитабельний текст на сайті розчаровує користувача, який скоріше проігнорує і текст і сайт. Дрібний шрифт, надвеликі текстові блоки та відсутність банальних підзаголовків відлякують значний сегмент цільової аудиторії.

Основні параметри розміщення тексту та типові помилки:

Шрифт.

Існує два типи шрифтів: із зарубками (serif) і без зарубок (sans serif).

Зарубки можуть бути скругленими в місці сполучення з основним штрихом ([англ. adnate serif](#)), а можуть поєднуватися під кутом і без закруглення.

В Інтернеті використовується переважно шрифт без зарубок, шрифти класу serif підходять краще для друкованої продукції. Це пояснюється тим, що з екрану їх читати складніше. Стандартними і популярними для основного тексту є шрифти Verdana, Arial та Tahoma, в західному сегменті можна зустріти шрифт Georgia із зарубками. Розміщення на сайті тексту в стандартному шрифті Times New Roman виглядає примітивно і нагадує саморобні безкоштовні сайти.

На сайті краще використовувати один-два шрифти в різних накресленнях (напівжирний, курсив). Більша кількість шрифтів виглядає неприємно, змушуючи користувача підсвідомо нервувати.

Колір шрифту. Кольори фону і основного тексту повинні бути контрастними. Чим контрастніше текст, тим він краще сприймається. Для того, щоб бути впевненим у достатній контрастності кольорів, потрібно створити дублікат макету сторінки і перетворити зображення у відтінки сірого, якщо текст є читабельним, тоді контрастність обраних кольорів є достатньою.

В ідеалі на сайті повинен бути темний текст на світлому фоні, або навпаки. Не варто використовувати синій колір для шрифту, так як він традиційно використовується для позначення посилань. Найгіршими поєднання вважають червоний чи синій на чорному тлі та блакитний чи світло-сірий на білому.

На сайті не слід використовувати більше чотирьох різних кольорів тексту. Не слід також забувати, що користувачі надають однакове значення елементам, які візуально виглядають однаково.

Розмір шрифту. Розмір шрифту для основного тексту повинен бути в межах від 12 до 16 пікселів, великий шрифт дратує, дрібний є нечитабельним. Для заголовків таких обмежень немає, і якщо дизайн допускає – шрифт може бути доволі великого розміру

Інтерліньяж (відстань між рядками) у веб-тексті повинен бути в півтора рази більшим, ніж у друкованих виданнях, це полегшує перегляд і читання тексту.

Абзаци. Тут немає якихось жорстких обмежень за розміром і кількістю, але завжди хочеться зробити текст не просто інформативним, але і красивим.

Відносно довжини абзацу, яка буде оптимальною для швидкості читання, існує припущення, що людське око в певний момент часу може сфокусуватися приблизно на 7,5 см рядка тексту, вважається, що довжина в 9 або 10 слів збільшує швидкість читання і сприйняття матеріалу. За висотою бажано робити абзаци приблизно однакового розміру, по 4-7 рядків. Занадто великі абзаци погано сприймаються, занадто короткі виглядають «брудно», тому самотні рядки варто об'єднувати до одного абзацу.

Вирівнювання тексту за лівим краєм збільшує швидкість читання, оскільки рівний лівий край допомагає знайти початок нового рядка. Вирівнювання тексту за шириною може надати вкрай небажані великі відстані між словами, тому використовується рідко.

Підзаголовки. Тексти довше за 1,5-2 тисяч знаків потрібно ділити на підзаголовки. Зазвичай, користувач не читає тексти, а переглядає у пошуках потрібної інформації. Підзаголовки економлять час і дозволяють швидше знайти те, що потрібно.

Шрифтові виділення. В арсеналі дизайнера є три види виділення основних думок тексту: напівжирне написання, курсив і прописні літери. Напівжирне виділення використовується для позначення головного слова абзацу, його тематики, курсив традиційно позначає головну думку автора, а прописні букви використовуються для резюмування і залучення уваги. Не рекомендується використовувати великі літери занадто часто - це на мові шрифту позначає крик і дратує відвідувача.

Виділення посилань. Доцільним вибором будуть традиційні кольори: сині посилання, фіолетові «пройдені». Якщо існуючий дизайн сайту не передбачає цього, тоді слід пам'ятати правило: посилання повинні бути виділені в інший колір, використані посилання повинні відрізнятися за кольором від активних, текст посилання слід підкреслювати.

Не варто підкреслювати текст, який не є посиланням - це вводить користувача в оману. Людина заходить на сайт, бачить щось схоже на посилання (те, що повинно бути посиланням, судячи з його досвіду), клікає і ... не отримує нічого. У мережі занадто велика конкуренція, щоб розчаровувати і ображати користувача такими дрібницями.

6. Порядок у вихідному макеті (шаблоні)

Одним із лідерів серед редакторів, що використовують веб-дизайнери є Adobe PhotoShop (його безкоштовним аналогом є Photorea, який з точки зору веб-дизайну і веб-розробки має ряд переваг над Adobe PhotoShop: перегляд CSS-властивостей, автоматичне завантаження використаних у шаблоні шрифтів, наявність вебверсії і т.д.). Правильно складений шаблон повинен бути збережений у форматі psd із чіткою ієрархією слоїв та груп слоїв. Вихідний макет очищають від елементів, які не використовуються, всім шарам дають інтуїтивно зрозумілі назви, для текстових слоїв роблять растрові копії. Важливим є наявність напрямних ліній по модульній сітці макету.

Такий порядок дозволяє швидко орієнтуватися в макеті, легко виділити, замінити чи змінити потрібний елемент.

Поради з розробки ефективного дизайну

1. Схемний погляд відвідувачів проходить по веб-сторінці за формою написання літери «F». Відвідувачі спочатку дивляться на ліву верхню частину сторінки, а згодом переводять погляд нижче і правіше. Інтерфейс більшості програм дотримується цього правила. Варто розташовувати важливі елементи сторінки в цих ділянках: заголовки, підзаголовки, виділений текст, щоб спонукати відвідувачів рухатися сайтом.

2. Меню у верхній частині сайту є найефективнішим. В ідеалі потрібно прагнути, щоб відвідувачі проглядали якомога більше сторінок на сайті, тому, важливо надати їм зручне меню для навігації по всіх розділах сайту.

3. Відвідувачі помічають заголовки на сторінках в першу чергу. Заголовки повинні бути добре помітними при першому погляді на сторінку та не повинні закриватися елементами дизайну.

4. Відвідувачі не люблять великі блоки тексту. Дослідження підтверджують, що більшість відвідувачів в Інтернеті уникають великих блоків текстової інформації, незважаючи на її оригінальність і корисність. Тому, важливо розбивати текст на невеликі параграфи, використовувати марковані чи нумеровані списки.

5. Використання вільних ділянок на сайті. Сайти з перенасиченою інформацією часто ігноруються користувачами. Варто використовувати просте оформлення і більше відкритого простору для того, щоб очі відвідувачів не перевтомлювалися.

6. Читачі часто переглядають нижні частини сторінок. Тому, це можна вигідно використати і розмістити там хмару тегів, додаткове меню, карту доїзду.

7. Незвичні ефекти і нестандартні шрифти ігноруються. Підсвідомо відвідувачі приймають такі елементи за рекламу. Згідно досліджень, відвідувачам важко відшукати потрібну інформацію в яскравому тексті, і вони його ігнорують. Слід не перенавантажувати дизайн сайту різними незвичайними ефектами.

8. Реклама у верхній і лівій ділянках сайту є найбільш ефективною. Реклама має бути ненастирливою і не порушувати дизайн.

Модні тенденції

Веб-дизайн подібно до дизайну одягу є властивими модні тенденції. Думки про те, що таке хороший дизайн, постійно змінюються. Над цим працюють лабораторії великих компаній, які проводять дослідження поведінки відвідувачів сайтів, аналізують їх особливості, звички та прагнення. Природно, результати цих досліджень негайно використовуються професійними веб-студіями при розробках сайтів для своїх клієнтів.

Формуються нові стандарти оформлення сайтів і розміщення на них інформації. З'являються цілі науково-практичні напрями - наприклад, «Інформаційна архітектура сайту». Тому те, що сьогодні здається правильним і сучасним, на завтра може бути вже не актуальним і застарілим. І такий стан справ є неминучим, бо технології Інтернету розвиваються дуже стрімко.

Аудиторія відвідувачів також зростає і Інтернет-середовище повинно ставати для них більш комфортним, зручним та інформаційно насиченим.

Лекція 4. Створення графічного макету сайту

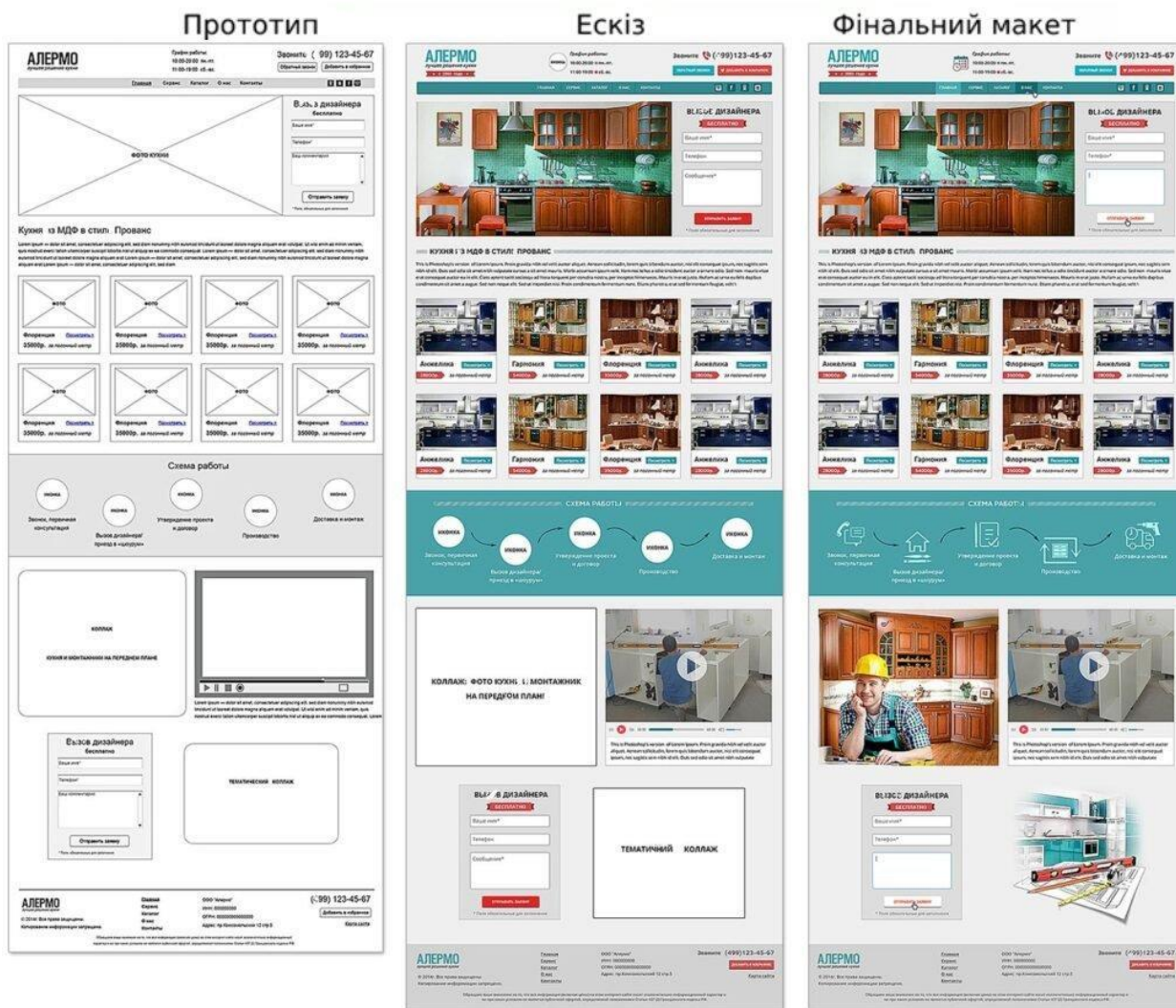
Зміст

- 1. 62
- 2. 7
- 3. 70
- 4. 80

1. Основні етапи створення макету сайту та навщо вони потрібні

Макет – це модель об'єкта в зменшеному масштабі або в натуральну величину, позбавлена, як правило, функціональності представленого об'єкта.

Макет сайту - це його реалістична модель, на якій показано розташування всіх елементів: блоки, функціональні кнопки і т. д. З його допомогою замовник може внести правки і детальніше висловити свої побажання. Умовний поділ процесу створення макету сайту на етапи визначається за складністю структури макету, рівнем візуалізації та інтерактивності. Відповідно до цього поділу, основними етапами створення макету сайту є створення прототипу, ескізу, графічного макету (фінального макету, графічного шаблону) та веб-макету (або веб- шаблону). Веб-макет - це зверстаний графічний макет. Часто в мережі Інтернет фінальний варіант обох видів макету називають просто шаблоном.



Різновиди макету сайту

Макети сайтів ділять на:

- статичні — «знімок» сторінки, де видно функції та основні блоки, але взаємодіяти з ними не можна: прототип, ескіз та графічний шаблон.
- динамічні — інтерактивна модель, макет веб-сайту з розміщеними елементами: веб-шаблон;

Статичний прототип корисний на початкових стадіях розробки сайту, коли ви продумуєте ідею, створюєте структуру та наповнення сайту. Важливо, що ви витрачаєте час на візуалізацію вашої ідеї в цілому, а не на створення окремих елементів.

Створення динамічних прототипів сайту допомагає визначитися, як будуть працювати посилання, наскільки зручно розташовані функціональні елементи, чи відповідає структура сайту потребам аудиторії. Інтерактивний прототип сайту вимагає більше зусиль і часу, але дає наглядніший результат.

Щоб створити макет сайту, рухайтесь від статичної моделі до динамічної. Етапи створення макету відрізняються за ступенем опрацювання, деталізації та інтерактивності. Якщо час обмежений і замовник вимагає відразу візуалізувати результат, можна пропустити кілька кроків і зробити динамічний макет.

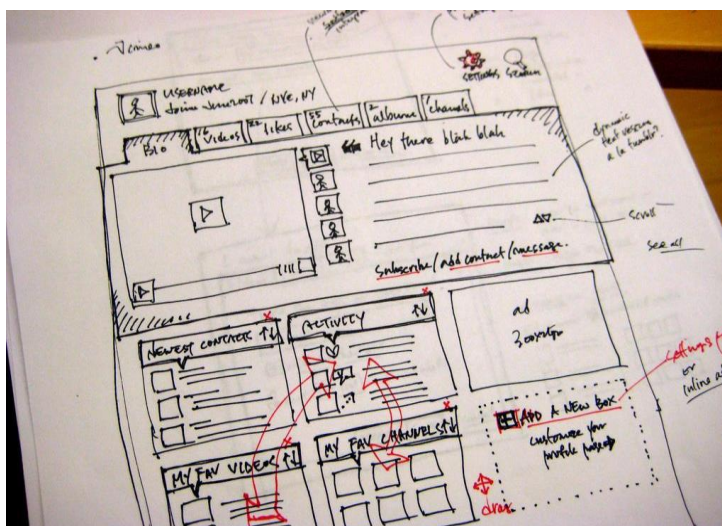
Прототип (Prototype)

Перш, ніж побудувати корабель або створити автомобіль, для них роблять креслення та макет. Вони допомагають візуалізувати кінцевий результат, не вкладаючи відразу багато грошей в його розробку. Прототип сайту дозволяє зробити те ж саме в контексті дизайну та верстки сторінок. За допомогою прототипу замовник і виконавець приходять до спільного розуміння, як буде виглядати кінцевий результат.

Прототип сайту (умовно відноситься до етапу низької деталізації, Low-fidelity) — це зразок майбутнього ресурсу в спрощеному вигляді, який оформлюється, як правило, у вигляді чорно-білих схем, позначаються тільки елементи, на яких потрібно акцентувати увагу. Це швидке прототипування сайту, при якому ви опрацюєте тільки великі блоки і можете легко міняти структуру. Прототип сторінки сайту показує, де будуть розташовані основні елементи, як користувач буде взаємодіяти з ресурсом без візуалізації контенту. На цьому етапі найпростіше зображувати нові ідеї, видаляти елементи, міняти їх місцями. Це чернетка, в якій можна виправити недоліки, змінити структуру та набір функцій до того, як розробники почнуть писати сайт. З їх допомогою замовникам простіше скласти ТЗ і висловлювати свої думки, а веб-майстрам оцінювати вартість роботи.

Якщо обговорюєте прототип в офісі, зручно користуватися маркерною дошкою. Створити макет сайту з командою онлайн можна з допомогою одного із онлайн-інструментів. Онлайн прототип сайту зручно коментувати, правити та обговорювати з замовником.

Приклад прототипу сайту “від руки”:



Отже, прототип сайту допомагає:

- узгодити з замовником схематичний вигляд готового сайту до активної фази розробки;
- визначити обсяг робіт, терміни та бюджет;
- уникнути ситуації, коли замовник безконтрольно вносить правки в проект, тому що всі домовленості були усними;
- узгодити план роботи та улагодити розбіжності в команді;
- змінити архітектуру сайту на ранній стадії, запобігти можливим помилкам.

Статичний недеталізований макет (ескіз, sketch)

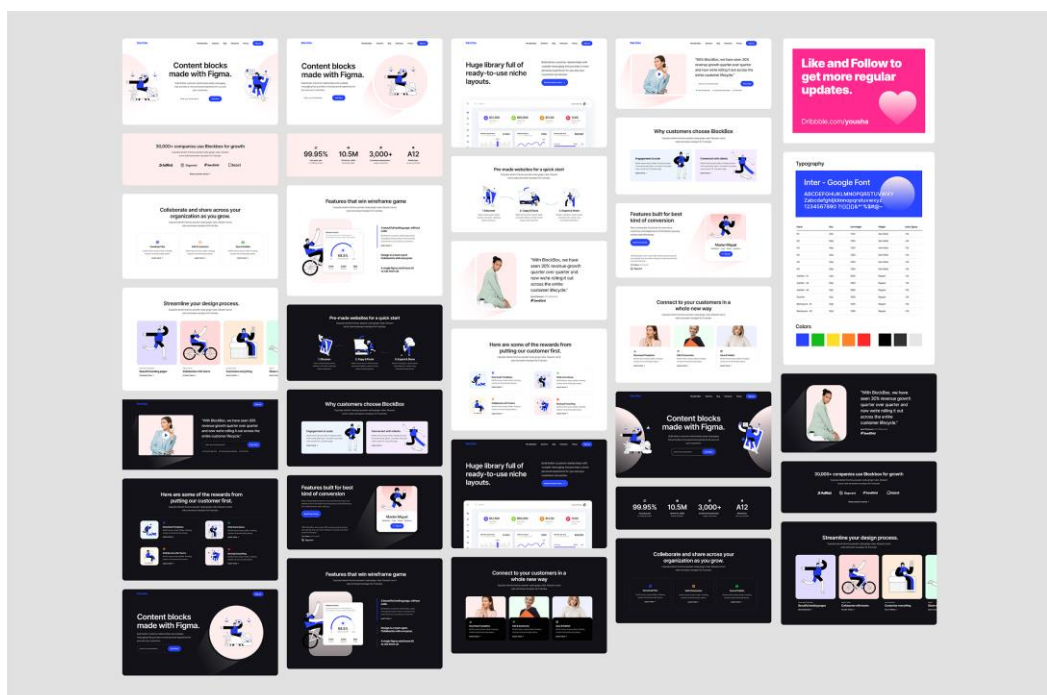
Коли прототип і структура сайту готові, приступайте до основних елементів. Визначте:

- де буде розташовуватися кожен об'єкт сторінки;
- якого розміру він буде;
- як елементи будуть поєднуватися між собою.

Цей етап є етапом створення ескізів, відноситься до етапу високої деталізації (High-fidelity). На цьому етапі недостатньо намалювати структуру сайту — її потрібно доповнити розміткою блоків, створити каркас сайту. Ескізи є більш функціональними моделями макетів сайтів. Дозволяють представити замовнику повну візуалізацію всіх сторінок. Схематичні квадрати замінюються на повноцінні блоки із зображеннями і орієнтовним контентом. Особливості:

- Для створення використовуються онлайн-сервіси або спеціалізоване ПЗ. Макет включає повну ієрархію сторінок, завдяки цьому можна реалізувати детальний макет сайту.
- Застосування таких макетів дозволяє оцінити обсяг роботи і визначитися зі структурою. Тому вони в першу чергу потрібні виконавцям.
- Розробка займає більше часу, в порівнянні з попереднім етапом.

Приклад ескізу сайту (High-fidelity):



Незважаючи на те, що створення ескізів сайтів займає час (в середньому від 1 до 10 днів), він прискорює процес розробки. Попередня візуалізація майбутнього проекту дозволяє уникнути

помилки і постійних правок з боку замовника. Це полегшує роботу дизайнерів, верстальників та back-end програмістів.

Статичний деталізований графічний макет (графічний шаблон)

Графічний шаблон сайту - це результат роботи дизайнера за завданням замовника. Тепер розробка макету сайту вимагає наповнити його змістом. На сторінках з'являється текст, заголовки, ілюстрації. Важливо, щоб контент та фірмовий стиль (брендбук) був готовий до моменту, коли ви почнете розставляти кольорові акценти та створювати настрій сайту. Графічний шаблон сайту виглядає як готовий сайт: з усіма блоками тексту і зображень, кнопками, фоновими малюнками і так далі. Єдина відмінність в тому, що в макеті немає функціонального наповнення: наприклад, сторінка з товарами заповнена шаблонами описів.

На сайті поки немає переходів між сторінками, але вже видно, яким контентом наповнена кожна з них. У створенні графічного шаблону необхідні навички копірайтера, дизайнера, ілюстратора і навіть маркетолога. Також не зайвим буде мати навички front-end та back-end-розробників, щоб оцінити можливості майбутньої веб-реалізації графічного шаблону.

Графічний шаблон допомагає:

- Заздалегідь побачити, як будуть виглядати побажання замовника на практиці, включаючи, наприклад, мобільну версію сайту і елементи анімації.
- Розробити дизайн в одному стилі для всіх сторінок сайту і всіх його типових елементів.
- Продумати наповнення елементів - меню, посилань, тексту і зображень.
- Скоординувати всю команду проекту, задати чіткі рамки для роботи.

Інтерактивна деталізована модель сайту (веб-макет)

Далі графічному шаблону потрібно додати динаміки, перетворити статичний графічний файл в динамічний з допомогою спеціалізованих програм чи онлайн-сервісів, або шляхом створення веб-шаблону чи справжнього сайту.

На цьому етапі прототип сайту перетворюється в діючу інтерактивну модель. Можна перевірити, як елементи сторінки взаємодіють між собою, як працює анімація та відео. Ви можете відкрити сторінку в браузері, щоб переглянути, як вона відображається на екрані ПК, планшеті та телефоні, як змінюються окремі блоки та їх взаємне розташування. (див. [Лекція 5.](#)) З таким прототипом простіше тестувати сайт і приймати його в роботу.

Функції створення інтерактивного макету сайту:

- Візуалізація ідей замовника дозволяє подивитися на конкретному прикладі, як поєднуються кольори, виглядають шрифти, вибрані зображення разом з логотипом і т.д.
- Оцінка того, як буде виглядати сайт для цільової аудиторії (ЦА). За наявності актуальної клієнтської бази, замовник може запитати думку ЦА.

Наявність інтерактивного макета дозволяє замовнику оперативно вносити правки по ходу розробки проекту. Клієнт може відкрити макет і внести коментарі. Така форма взаємодії спрощує співпрацю, тому що замовнику не доводиться «пояснювати на пальцях», що він хоче. Макет сайту – це свого роду візуалізація технічного завдання.

Макет сайту на конструкторі

Якщо ви хочете швидко запустити сайт і готові вносити правки в процесі роботи ресурсу, можна зробити сайт на конструкторі. Така розробка макетів сайту вимагає менше навичок і ресурсів, але водночас є обмеженою по можливостях розробки індивідуальних дизайнерських і структурних рішень. У конструкторі є приклад макету сайту, який можна

взяти за основу. В результаті модель вашого сайту природним чином перетворюється на працюючий ресурс. Одним із таких конструкторів є <https://sites.google.com>

2. Способи створення макетів сайту

Перед тим як почати створювати макет сайту, потрібно визначити для яких цілей він створюється:

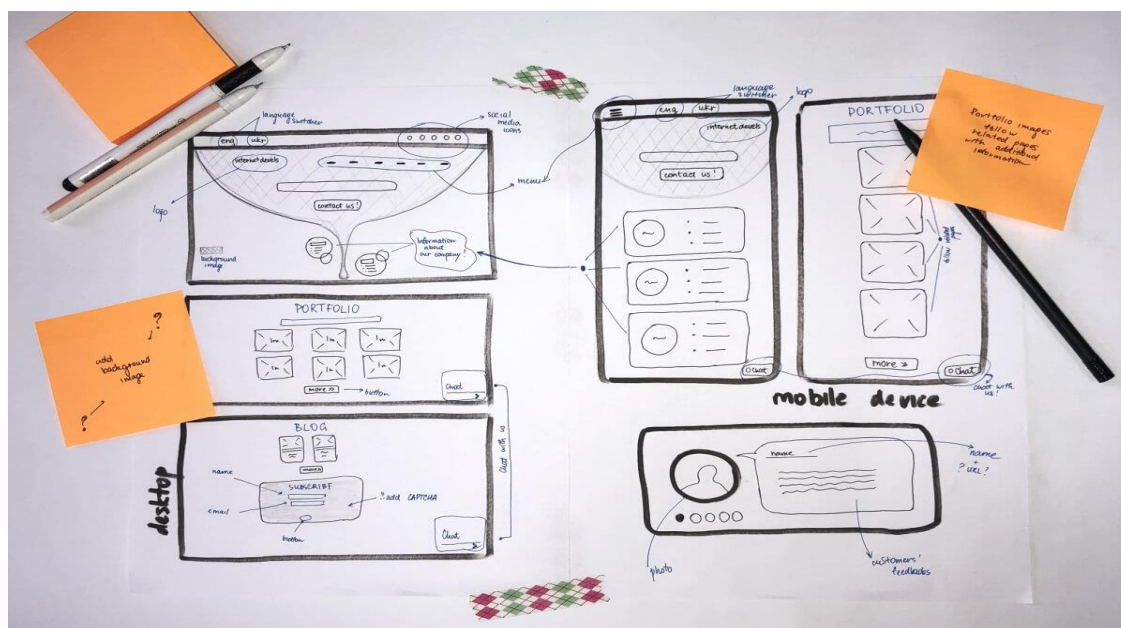
- Визначення розташування блоків на сторінці.
- Систематизація структури всього сайту – складання ієрархії сторінок.
- Візуалізація, тобто складання детального плану, в якому вказано розташування та вигляд основних елементів.

Якщо завдання полягає у визначенні структури сторінок, підійдуть варіанти категорії Low-fidelity. Якщо ж потрібна візуалізація, раціональніше створити детальніший макет в спеціалізованій програмі або через онлайн-сервіси. Існує 4 способи створення макета сайту: від руки, за допомогою спеціалізованих програм, онлайн-сервісів або використовуючи готові шаблони. Розглянемо всі варіанти.

Від руки

Перед тим, як створювати дизайн сайту можна скласти план структури на папері. Для цього не потрібні навички верстальника або знання геометрії. Досить озброїтися ручкою, лінійкою і аркушем паперу. Деякі замовники так і роблять – перед зустріччю з виконавцями малюють від руки приблизний план проекту. Для багатьох це простіше, ніж заповнювати десятки сторінок брїфів або складати детальніший ТЗ.

Як зробити прототип сайту на папері? Чітких вимог немає, головне щоб схема була зрозуміла всім сторонам (клієнту, менеджеру проекту і виконавцям). Схему можна зробити на аркуші формату А4 або більшого формату, якщо мова йде про багатосторінковий сайт. Приклади прототипу, створеного ручкою:



Недолік такого способу в примітивності візуалізації. На паперових макетах неможливо вносити серйозні правки, за необхідності серйозних коригувань доводиться переробляти креслення. Тому подібні схеми використовуються на етапах обговорення проекту між

замовником і менеджером. Для подальшої розробки проекту повинні застосовуватися інші інструменти.

Програмне забезпечення

На етапах первинної розробки використовуються програми для прототипування сайту. Вони володіють широким функціоналом, дозволяючи опрацьовувати структуру майбутнього проекту. Створювані макети легко верстати і переносити у фреймворки.

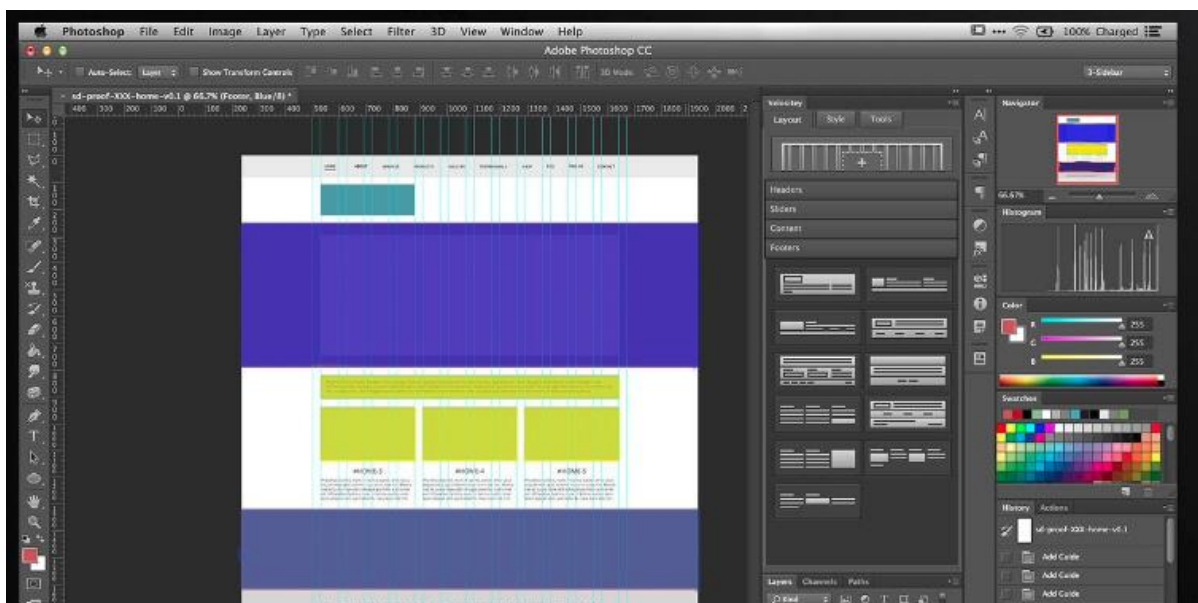
Переваги прототипування за допомогою професійних програм:

1. можна додавати інтерактивні елементи
2. це естетично і професійно
3. є можливість редагувати окремі розділи

Недоліки прототипування за допомогою професійних програм:

1. як правило, вимагає фінансових витрат
2. вимагає спеціальних знань
3. вимагає більше часу, щоб завершити проект

Однією із найпопулярніших програм, якими користуються розробники є графічний редактор Adobe Photoshop, безкоштовним аналогом якого є додаток Photopea. Незважаючи на появу інших програм, даний редактор залишається незамінним інструментом для дизайнерів.



Плюси:

- Для використання не потрібне з'єднання з мережею Інтернет.
- Багатофункціональність.
- Підтримка всіх графічних форматів.
- [Пакетна обробка і можливість створення скриптів](#)

Мінуси:

- Платний.
- Додаток споживає багато оперативної пам'яті (ОЗУ), на офісних ПК або ноутбуках працює з перебоями.

- Складність розробки – щоб розібратися в функціоналі потрібно витратити досить багато часу.

- Неможливість створення інтерактивного макету.

Порівняно із Photoshop, додаток Photopea є безкоштовним, має онлайн-версію і може напряму працювати із файлами, які розташовані на хмарних ресурсах, дозволяє переглянути стилі елементів і т. д., водночас у ньому реалізовано не всі функції доступні у Adobe Photoshop.

Axure RP – професійна програма для прототипів додатків і сайтів. Можливостей для візуалізації мінімум, тому її застосовують під час створення Low-fidelity макетів.

Плюси:

- Розташування елементів змінюється шляхом перетягування, що економить час.
- Можна зробити клікабельним макет (так замовнику буде простіше оцінити юзабіліті).
- Створення інтерактивних макетів.

Мінуси:

- Мінімальний тариф – 25 доларів на місяць. Trial-версія надається всього на 30 днів.
- Відсутній вбудований графічний редактор, а для створення візуалізації (банери, логотипи) доводиться використовувати інше програмне забезпечення.

Figma – професійна програма для створення макетів сайту. Векторний редактор дозволяє експортувати створені файли практично в будь-які формати.

Плюси:

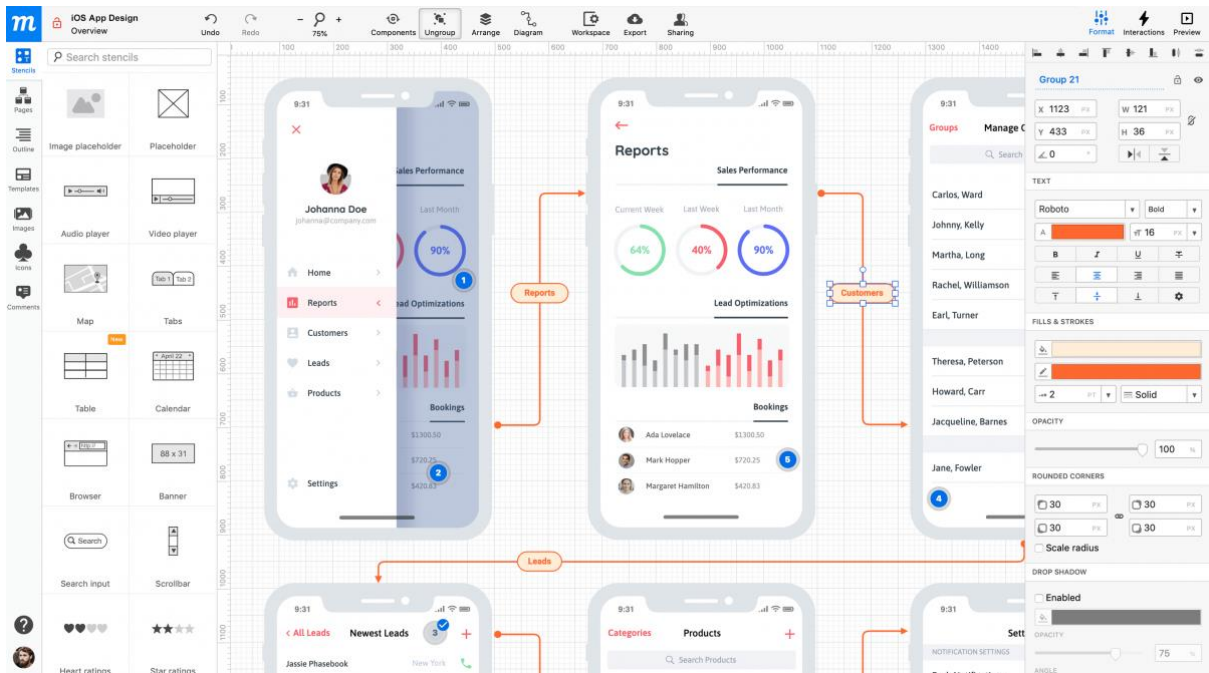
- Є безкоштовна версія для освітніх цілей. Платні тарифи дешевші, в порівнянні з багатьма аналогами.
- Доступно сотні плагінів та бібліотек.
- Дозволяє створити інтерактивний макет.
- Є онлайн-версія, що дозволяє організувати командну роботу.

Мінуси:

- відсутній інструментарій для опрацювання растрових зображень.
- менші можливості для створення дизайнерської графіки у порівнянні із тим же Adobe Photoshop.

Онлайн-сервіси

Прототипування (Prototyping) за допомогою онлайн-інструментів — один з найбільш зручних способів прототипування веб-сайту. Велика перевага в тому, що це онлайн-інструмент, і доступ до нього може отримати будь-хто. За функціоналом такі сервіси схожі до програм, часто є аналогами десктопних версій, наприклад додаток Photopea, figma.com.



Переваги прототипування за допомогою онлайн-інструментів:

1. вся команда має доступ до проекту;
2. не вимагає великих фінансових витрат;
3. простий і зручний у використанні.

Недоліки прототипування за допомогою онлайн-інструментів:

1. повністю використовувати всі можливості доступно тільки у платних тарифах;
2. на створення якісного продукту потрібен час.

Розглянемо три популярних сервіси та наведемо плюси і мінуси кожного з них.

Draftium – сервіс для створення прототипів сайтів онлайн. Користувачам доступна база із сотень готових шаблонів, які можна редагувати за допомогою вбудованого конструктора.

Плюси:

- Простий інтерфейс.
- Готові шаблони адаптовані під всі пристрої.
- Висока швидкість роботи.

Мінуси:

- У безкоштовній версії всього 50 шаблонів.

Wireframe – онлайн конструктор для створення Low-fidelity проектів. Інтерфейс сервісу гранично простий – користувач малює блоки за допомогою стандартних інструментів (як в Paint).

Плюси:

- Щоб розібратися з інтерфейсом вистачить 10-15 хвилин.
- Сервіс не вимогливий до «заліза», працює навіть із невеликою (2-3 Гб) кількістю оперативної пам'яті.

Мінуси:

- У базовій версії мінімальний функціонал, платні тарифи починаються з 16 доларів в місяць.
- Немає можливостей для створення складних графічних елементів.

Marvel – функціональний онлайн-сервіс з десятками інструментів візуалізації. З його допомогою можна створювати масштабні проекти, готові для перенесення у фреймворк.

Плюси:

- Користувачам платних тарифів доступні сотні готових шаблонів.
- Функціональний графічний редактор.
- Наявність безкоштовної версії з базовим функціоналом.

Мінуси:

- Відносно складний інтерфейс.

Інші програми для прототипування та онлайн-сервіси, які вам можуть знадобляться при створенні макета: [XMind](#), [Diagrams1](#).

Використання готових шаблонів

В умовах дефіциту часу чи ресурсів можна використовувати готові макети сайтів.

3. Розробка графічного макету сайту

Незалежно від того, які програми для макетування сайтів ви використовуєте, є загальний підхід до роботи над макетом. Він підкаже, як створити макет сайту, який відповідає потребам і запитам клієнтів.

КРОК 1 Визначте мету сайту та складіть ТЗ

Перш, ніж створити дизайн сайту, з'ясуйте та задокументуйте:

- хто входить в вашу цільову аудиторію: стать, вік, поведінка клієнтів, їх інтереси;
- яке завдання бізнесу та клієнтів вирішує сайт, яка його роль серед всіх бізнес-процесів;
- що ви як компанія хочете донести до клієнта, яке основне послання та заклик до дії;
- яка мета кожної сторінки, як відвідувач буде взаємодіяти з її елементами та яке призначення кожного з них;
- технічні рішення, які важливі для роботи сайту: чи досить сайту-візитки або це повинен бути інтернет-магазин з кошиком і онлайн-платежами
- техзавдання (детальніше цей та наступний етапи див. [Лекція 3.](#))
- наявність чи розробку брендбука (фірмового стилю), підбір кольорів і шрифтів.

Наведені вище пункти визначають мету сайту та вимоги до його структури і дотримання фірмового стилю. Коли ви знаєте, чого хочете, можна переходити до створення прототипу.

КРОК 2 Зробіть прототип сторінок

За допомогою прототипу зобразіть ідею, бачення, як ви плануєте реалізувати поставлені в попередньому пункті завдання. Визначте кількість сторінок, базові елементи, їх розташування. Зробіть виноски та запишіть, яка мета цієї сторінки, блоку, варіанти їх розміщення.

На даному етапі важливо визначитися, навіщо вам сайт і не захоплюватися деталізацією. ПЗ для створення прототипу сайту в цьому плані більш «підступне», тому що в пориві творчості

можете не помітити, як через кілька годин вже вибираєте колір для кнопки «Додати в корзину».

З рукописними інструментами простіше тримати себе в руках. Беріть олівець, ручку або маркер одного кольору — і чим більше розмір аркуша, тим товстіше повинен писати інструмент. Так вам буде не зручно малювати дрібні деталі та простіше думати про структуру сайту, а не його наповнення.

КРОК 3 Оцифруйте прототип

Коли затвердили прототип, пора перенести його з паперу в програму для прототипування. В процесі чорно-білий прототип доповнюють контентом: заголовками, текстом, картинками. Потім за допомогою програми для прототипування сайтів додають дизайн. Він повинен відповідати стилю бренду, створювати бажаний настрій, асоціюватися з компанією. В результаті прототип переросте у графічний шаблон, який повинен бути максимально схожим на готовий ресурс.

Розглянемо детальніше безпосередньо створення графічного макету сайту, в якому можна виділити наступні етапи:

1. Створіть сітку з напрямних, під яку будуть підлаштовуватися об'єкти на сторінці.
2. Розбийте сторінку згідно прототипу - визначте місце для хедера, футера, основних блоків контенту.
3. Створіть кілька базових елементів - приклади кнопки, опису, заголовка і інші потрібні вам елементи, їх винесіть за робочу область або в інший документ, збережіть як компоненти тощо. Потім зможете вставляти їх в потрібні місця.
4. Починайте детально проробляти кожну з областей.
5. Не забудьте зобразити інтерактивні елементи в різних станах. Наприклад, якщо кнопка змінює стан при натисканні або шрифт збільшується при наведенні, створіть окремі шари для різних станів (це зручно робити з допомогою смарт-об'єктів) чи створіть анімацію.

Важливо! Кожен елемент малюйте на окремому шарі і логічно називайте. Кнопка - *button*, назва товару - *item name* і так далі. Не забувайте логічно згрупувати шари і давати групам імена.

Хорошим тоном буде зробити опис графічного макету сайту для верстальника. Свого роду інструкцію до нього. В описі можна зафіксувати всі основні характеристики шаблону:

- вибрані кольори,
- розміри сітки,
- опис базових елементів і так далі.

Також додайте в опис елементи, які можуть викликати сумніви у верстальника, наприклад, скріншоти станів інтерактивних елементів з описом анімації. Опишіть типографіку, яку використовували: шрифти, їх кольори і розміри, міжрядковий інтервал. За наявності фірмового стилю, цей опис повинен ґрунтуватися на ньому.

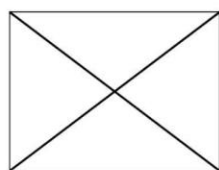
Сюди ж можна винести порівняння версій макетів сторінок для різних пристроїв: що змінюється, коли сайт переглядають на комп'ютері, планшеті або телефоні, тощо.

Розглянемо покроково, як створити макет сайту в додатку Photorea за цією схемою. Створимо графічний ескіз головної сторінки Інтернет-магазину з продажу чашок, на ній будуть хедер, блоки з категоріями товарів і футер, попередньо створивши її прототип.

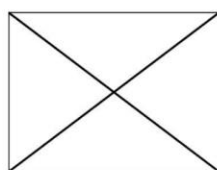
Logotype

Доставка Гарантії Про нас

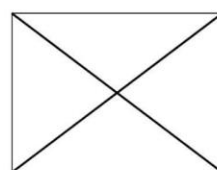
Слоган!!!!



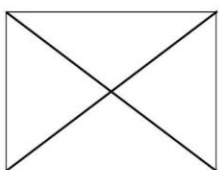
Класичні



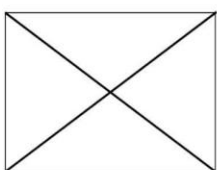
Тематичні



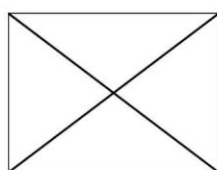
Туристичні



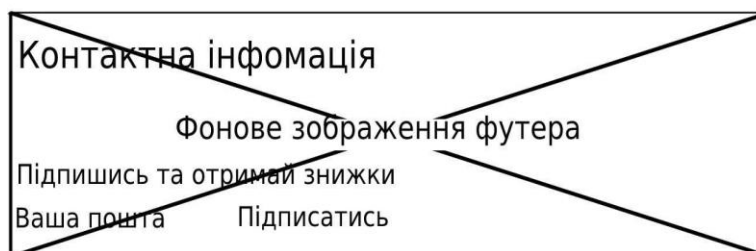
Ручна роботи



Подарункові



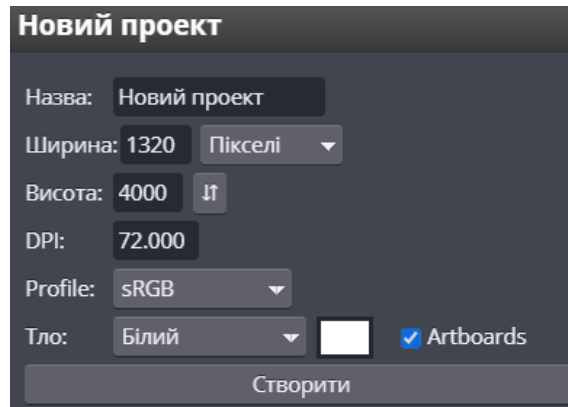
Під замовлення



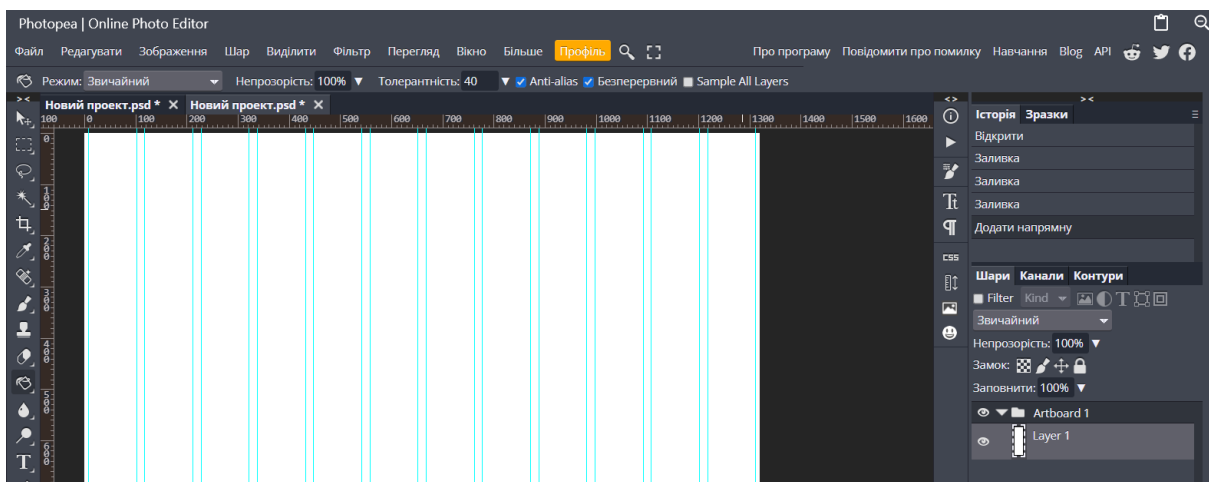
Прототип головної сторінки

Проаналізуйте, з яких пристроїв частіше заходять (заходитимуть) на ваш сайт - телефонів, планшетів або комп'ютерів. Насамперед зверстайте версію під найпопулярніший варіант, а від неї вже розробляйте інші.

Створюємо новий файл в Photopea. Введемо розміри 1320 на 4000 pixels. Це поширений розмір, який підходить під розмір екрану комп'ютера або ноутбука, висота дозволить створити кілька сторінок. Крім того, відповідає одному із розмірів, заданих у варіантах сітки [Bootstrap 5.0](#). Колірна модель - sRGB, роздільна здатність - 72 пікселя на дюйм, фон білий, поставте відмітку біля [Artboards](#) (монтажна область).



Тепер задаємо модульну сітку. Заходимо у вкладку "Вікно-Більше-Guide Guy". Задамо 12 колонок, так буде зручніше розділяти сторінку на 2, 3, 4 або 6 блоків, і дасть змогу при потребі верстати з допомогою CSS фреймворка [Bootstrap](#). Також поставимо значення відступів між колонками 16 пікселів, верхнє і нижнє поле поставимо по нулях, праве і ліве - по 8 пікселів. Ширину колонки задавати не потрібно, вона визначиться автоматично і буде рівною $(1320-12*16)/12= 94px$. Або навпаки, задати ширину колонки, тоді не задавати відступи між колонками. Горизонтальні лінії поки не будемо задавати. Ви можете вибрати свої значення, але пам'ятайте, що відступи і поля повинні бути кратні одній цифрі, наприклад, 8.

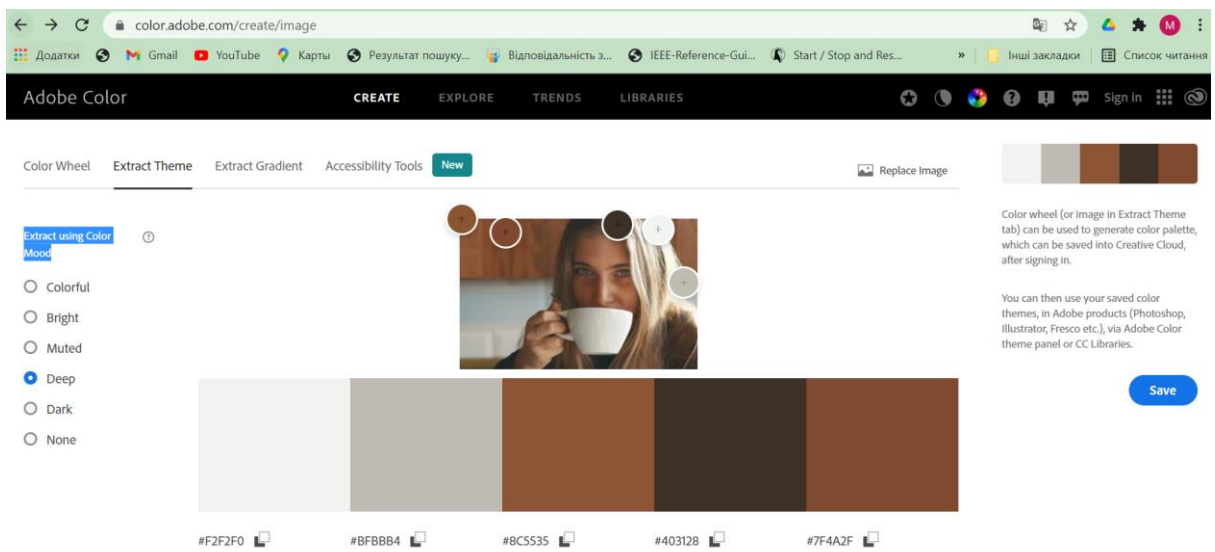
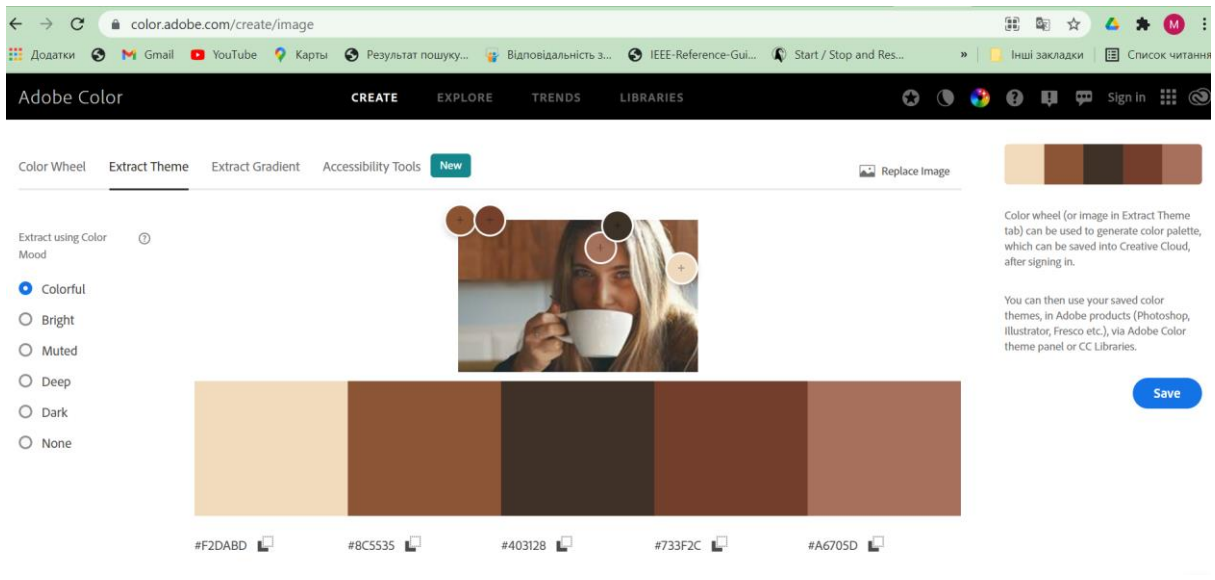


Створення сітки направляючих, переважно з 12 стовпців

Переіменуємо і заблокуємо фоновий шар.

Наступний крок - вибрати кольори. За наявності фірмового стилю, потрібно відштовхуватися від нього. За його відсутності або за потреби доповнення кольорами, можна, наприклад, підібрати фонову картинку головного екрану і від неї відштовхуватися. Зображення для сайту можна взяти в одному з [безкоштовних фотобанків](#), наприклад [Unsplash](#), <https://pixabay.com/>.

Виберемо кольорову гаму за допомогою сервісу [Adobe Color](#). Заходимо в сервіс, вибираємо в меню «Витяг теми/ Extract Theme», завантажуюємо картинку і отримуємо кілька готових поєднань. Можете змінити параметр Extract using Color Mood, а також місця звідки брати кольори.



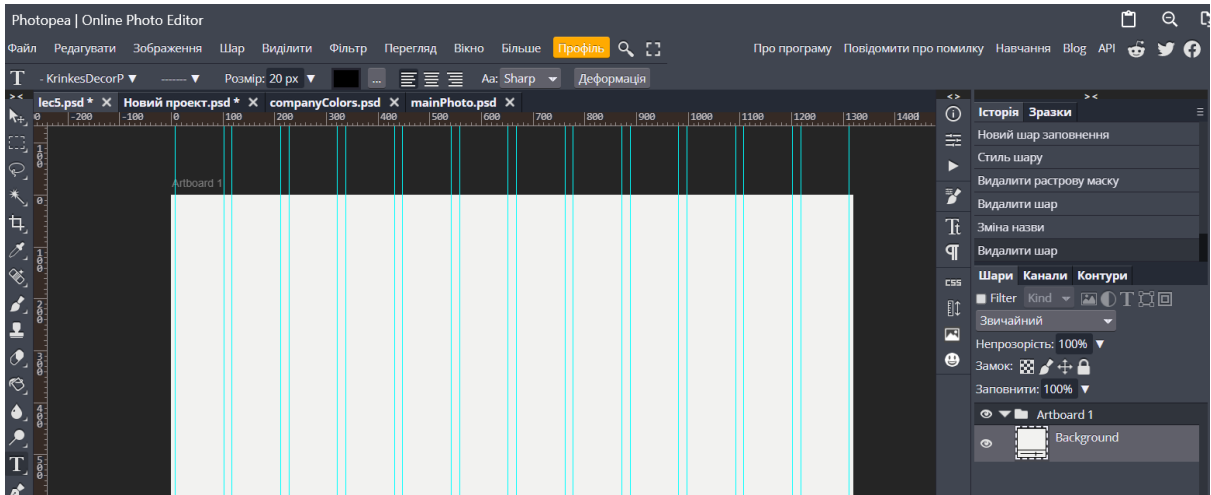
Сервіс Adobe Color допоможе підібрати кольори по фотографії

Коди отриманих кольорів можна скопіювати, також можна створити новий документ, в якому створити квадрат і залити першим кольором, зробити 4 копії квадрату і залити їх решту кольорами, а потім звідти брати піпеткою потрібний колір.



Готово, можна приступати безпосередньо до зображення ескізу.

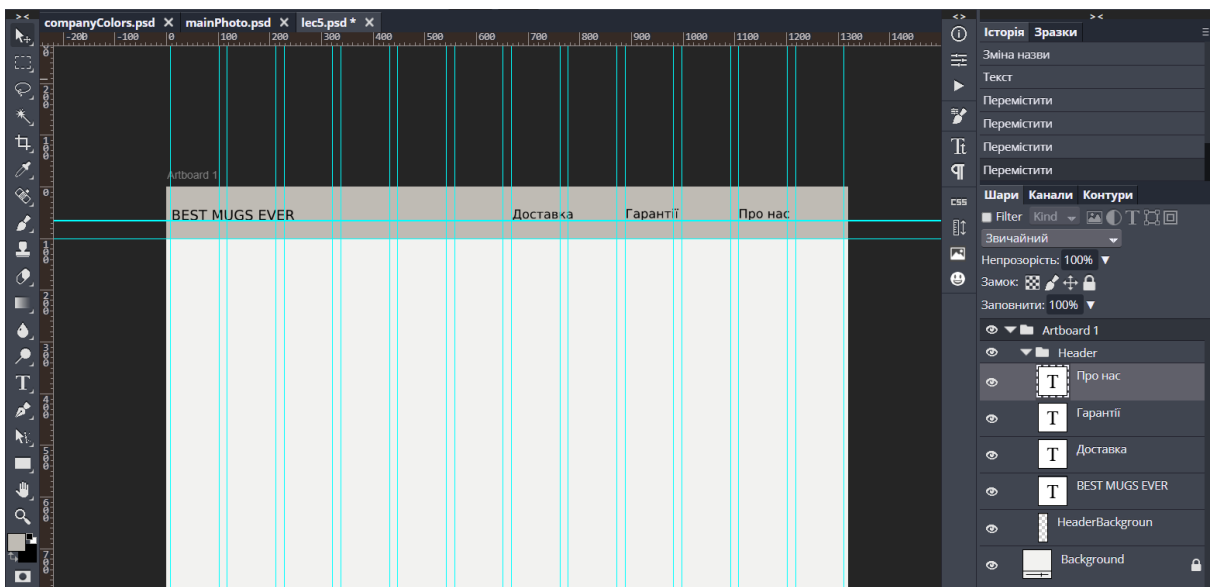
Тепер задаємо колір фону. Для цього використовуємо коригувальний шар - так буде простіше при необхідності змінити його колір, щоб не заливати заново. В меню «Шари» вибираємо пункт «Новий шар заповнення», натискаємо на нього і вибираємо опцію «Заповнення суцільним кольором», вказуємо потрібний відтінок (#f2f2f0). Тепер залишається лише видалити маску з цього шару (через контекстне меню «Видалити растрову маску»), щоб залишився тільки колір. Не забудьте перейменувати шар - краще на латиниці, можете відразу його заблокувати за допомогою кнопки з іконкою замку в панелі шарів.



Відповідно до розробленого попередньо прототипу, розробимо ескіз стартової сторінки Інтернет-магазину із продажу чашок.

Насамперед створимо хедер з логотипом компанії і посиланнями на інші розділи сайту. В основі буде прямокутник, залитий одним з обраних кольорів висотою 100 px. Замість логотипу просто напишемо придуману назву в шапці і створюємо заголовки для інших розділів сайту.

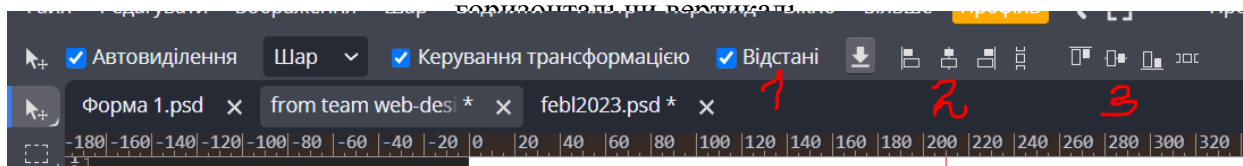
Для вирівнювання “витагнемо” з лінійки дві горизонтальні напрямні лінії. Також напрямними лініями відділимо висоту сторінки, яка буде відобразитися при завантаженні головної сторінки сайту і мала б поміститися на екрані. Цю висоту можна знайти із співвідношення ширини і висоти роздільної здатності екрану, яка, як правило, рівна 1,778 (для співвідношення 16:9). Звідси висота першого екрану головної сторінки дорівнює $1320/1,778 \sim 750$ px. Саме в межах цього прямокутника потрібно старатися розмістити по максимуму корисних та інформативних елементів. Згрупуємо всі ці шари в папку Header. Щоб згрупувати шари: виділяємо потрібні і Ctrl+G.



У хедер поміщаємо логотип і посилання на розділи

Зверніть увагу, що в опціях інструмента Переміщення (V) при встановленні прапорця біля Відстані (1) при виборі об'єкта чи групи об'єктів відобразатимуться їх розміри та відстані зверху, збоку тощо.

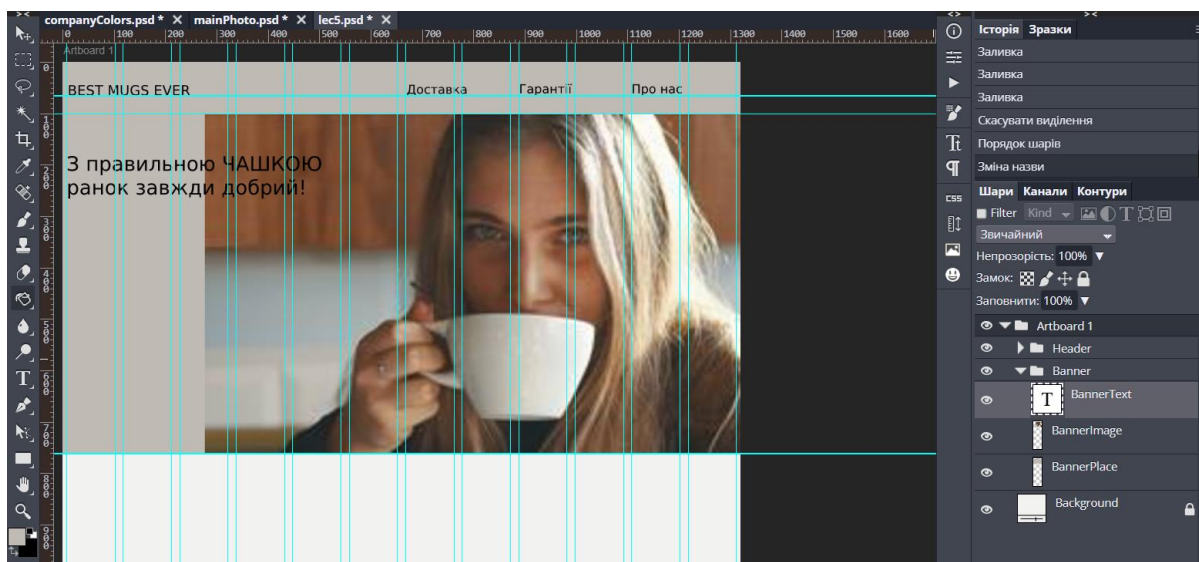
Група опцій під номером 2 та 3 дозволяє вирівняти виділені об'єкти до певної лінії по



Опції інструмента Переміщення (V)

Тепер додамо головне зображення сторінки і напишемо на ньому невеликий слоган. Контейнер для банера, сам банер і текст на ньому згрупуємо в папку Banner. Фото у блоки певної форми краще додавати через **маску відсікання (Clipping Mask)**. Для цього, знаходячись на шарі під зображенням з допомогою інструменту Прямокутник (U) створюємо прямокутник розміром боксу для зображення, далі при нажатій клавіші Alt клікаємо на саме зображення. В результаті наше зображення поміститься у прямокутник, тепер ми можемо його трансформувати, рухати, воно все одно не вийде за межі цього прямокутника.

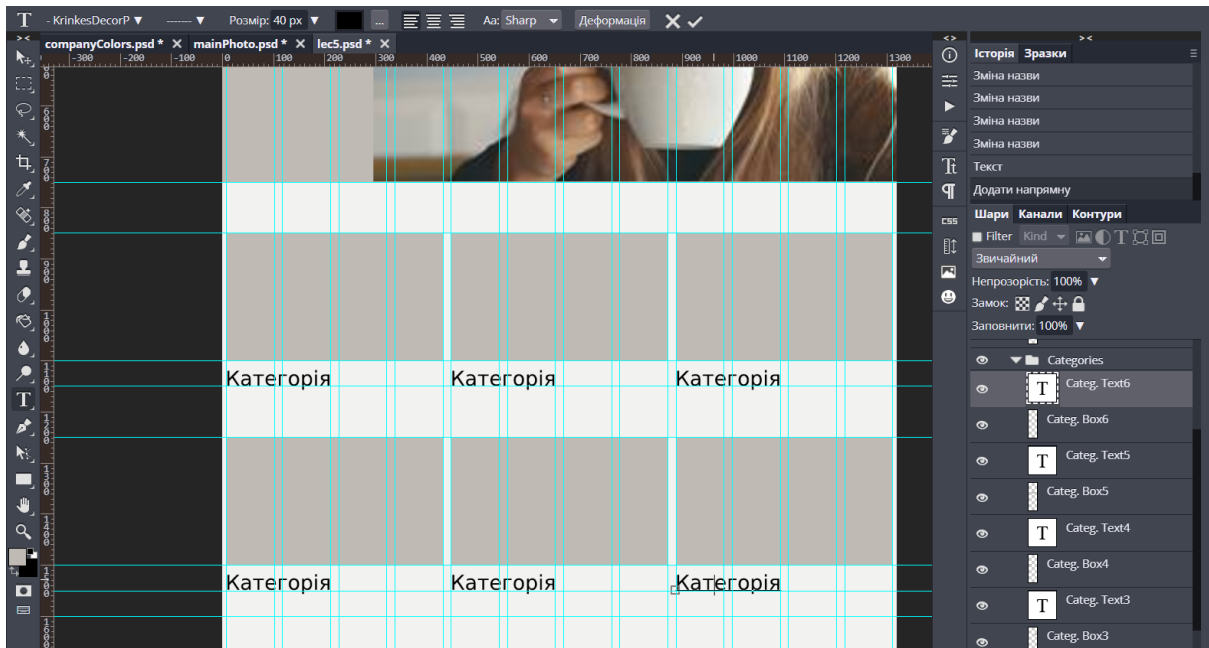
Якщо будете розміщувати картинку як у цьому прикладі, не залишаючи поле з одного боку - обов'язково зробіть позначку для верстальника, як повинна вести себе картинка при розтягуванні сторінки в ширину. Наприклад, що вона повинна залишатися «приклеєною», тобто праворуч не повинно виникати порожнє поле; або ж вона повинна збільшуватися разом зі сторінкою і т. д.



Під

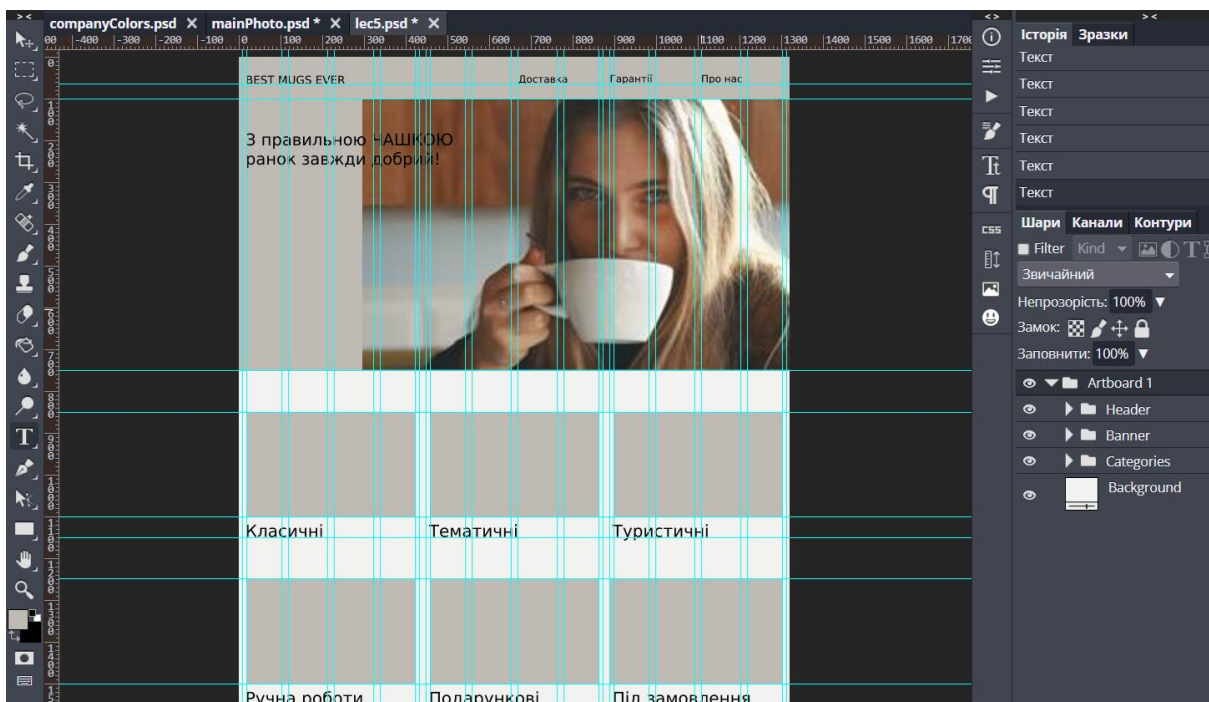
хедером розмістимо головне фото і слоган магазину

Прийшла черга зробити блоки категорій, їх буде шість, які теж створимо з допомогою інструменту Прямокутник (U). Щоб не робити їх занадто маленькими, розмістимо їх в два ряди по три блоки. З огляду на те, що в шаблоні 12 колонок, один блок займе рівно чотири колонки. Висоту блоку і відстані по вертикалі може визначити із висоти одного екрану, яка рівна 750 px: у нас буде два відступи і два рядки з блоками: якщо відступ зверху візьмемо рівним 100px, а вертикальний відступ між блоками 150px (100px відступ + 50px текст), а під другим рядком теж виділимо 50px на текст і 50px на відступ, то висота блоку рівна $(750-150-100)/2=250px$. Підберемо розмір шрифту в залежності від ширини сторінки: $1vw=1320*0.01=13,2px$. Виберемо розмір шрифту приблизно рівним $3vw$, це буде приблизно 40px. Не будемо відразу опрацьовувати кожен блок, а створюємо шаблон з сірим фоном і текстом Категорія, розмножимо його на потрібні позиції:



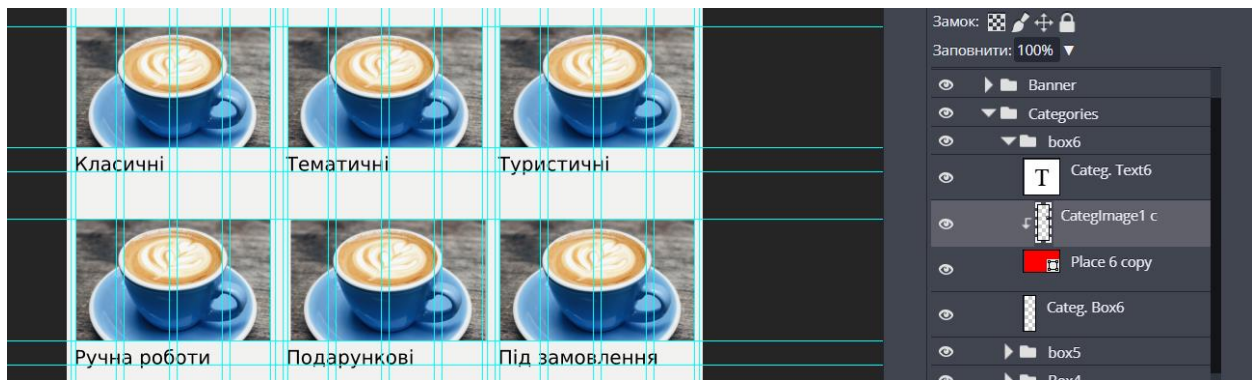
Категорії товарів представимо у вигляді шести симетричних блоків

Додамо назви категорій. На цьому етапі видно, що поля між блоками вийшли досить вузькими і фотографії будуть зливаються. Щоб це виправити, зменшимо кожен блок на 10 пікселів зліва і справа. Зверніть увагу, що логотип і слоган зсунулися на 10 пікселів вправо, щоб відступи відповідали відступу категоріям.

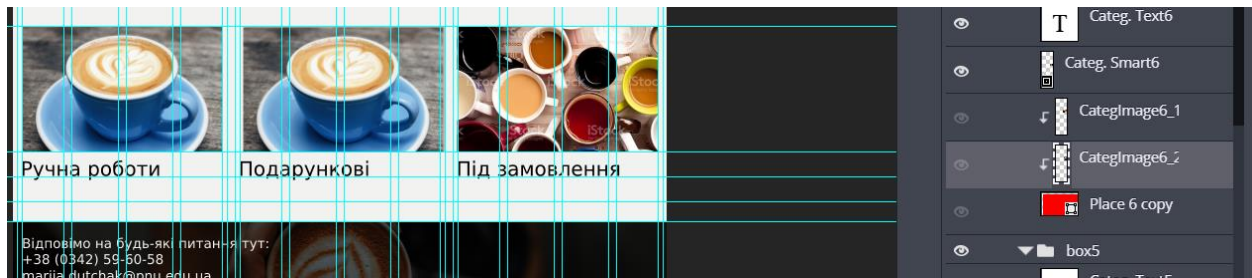


Збільшення полів між блоками

У ескіз для категорій додаємо ілюстративні зображення, не обов'язково остаточні, можуть бути додані якісь шаблонні. Паралельно групуємо елементи, так щоб всі шари, що стосуються однієї картинки були в одній групі. Для даних зображень теж створимо 6 окремих Масок відсікання (Clipping Mask).

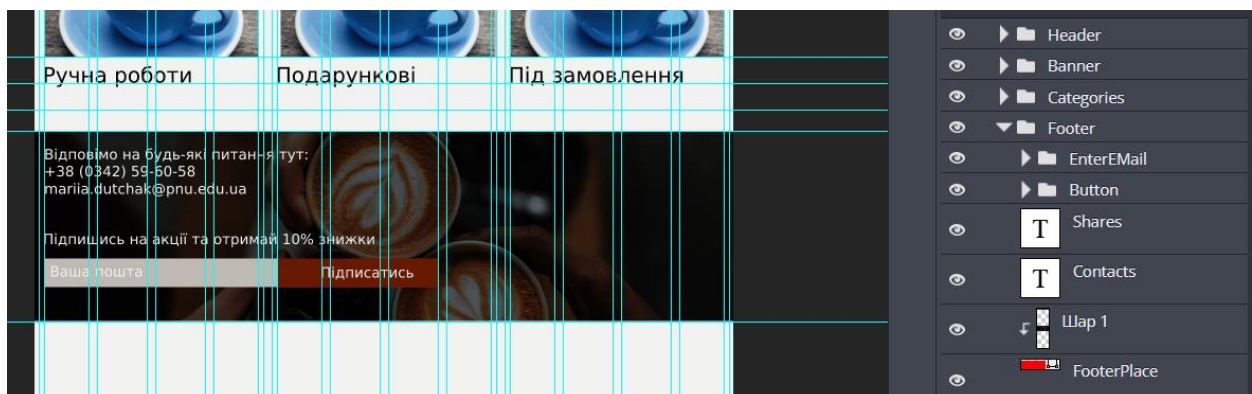


Також для показу альтернативних зображень можна місце під зображенням перетворити у Smart Object, а у зв'язаний файл додати альтернативні зображення.



У наш ескіз залишилось ще додати футер. Поставимо сюди форму для збору e-mail і вкажемо контакти для зв'язку. Щоб не ставити звичайний кольоровий фон, поставимо у футер зображення. Для цього вставляємо в ескіз потрібне зображення, за потреби попередньо можемо змінити його яскравість і контрастність, так щоб воно стало темнішим чи світлішим, щоб текст на ньому став більш читабельним. Для даного зображення теж створимо Маску відсікання (Clipping Mask).

Таким чином, в результаті використання маски відсікання один раз задавши потрібний розмір блоку, ми зможемо в нього підставляти зображення будь-якого розміру, зображення все рівно не вийде за його межі. Маску можна легко скасувати, натиснувши на шар з маскою правою кнопкою миші і знявши відмітку біля "Маска відсікання / Clipping Mask).



У футер поміщаємо фонове фото, на ньому розміщуємо форму підписки і контакти

Зверніть увагу, що елементи футера також відступають на 10 пікселів від направляючої, як і категорії над ними. Не забуваємо обов'язково групуємо шари і даємо їм зрозумілі назви, щоб верстальщику потім було простіше розібратися.

Нагадування: Ctrl+; - показати/відобразити направляючі.

З правильною ЧАШКОЮ
ранок завжди добрий!



Класичні



Тематичні



Туристичні



Ручна роботи



Подарункові



Під замовлення

Відповімо на будь-які питання тут:
+38 (0342) 59-60-58
mariia.dutchak@pnu.edu.ua

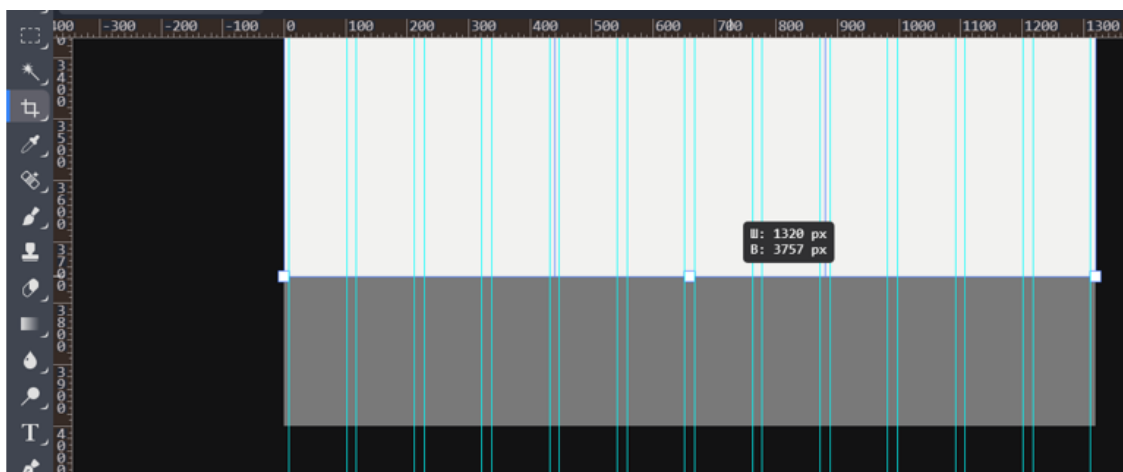
Підпишись на акції та отримай 10% знижки

Ваша пошта

Підписатись

Готовий ескіз сайту

Дальше під футером можна продовжити роботу із створення внутрішніх сторінок сайту. Зайву частину полотна, що залишилася після завершення роботи над ескізом можна обрізати, чи навпаки продовжити, якщо вона надто довга чи коротка. Це можна зробити з допомогою інструменту Кадр (C):



У результаті ми отримуємо ескіз нашого майбутнього сайту для перегляду на екрані ПК. Дальше потрібно буде розробити мобільну версію, у якій, наприклад, категорії можуть бути не у три колонки, а у дві, меню не в рядок, а через клікання на кнопку (т. з. “гамбургер”) і т.д.

Для того, щоб наш ескіз став графічним макетом (шаблоном), потрібно його доопрацювати, задавши потрібний вигляд шрифтам, посиланням, кнопкам і т.д., додавши ілюстративні фото до категорій, можливо додавши кілька різних виглядів елементам, які змінюють свій вигляд в залежності від дій користувача і т. д. При цьому деякі елементи шаблону Ви можете перетворити на смарт-об'єкти, з якими можна зв'язати кілька різних варіантів відображення і запропонувати замовнику вибрати один із них. Не забувайте при цьому групувати шари і надавати їм інтуїтивно зрозумілі назви. Після того, як ми зверстаємо наш графічний шаблон, він стане веб-шаблоном і буде готовим до заповнення реальними товарами та іншим необхідним контентом.

4. Помилки в створенні шаблону сайту

Найпоширеніші помилки при шаблонуванні можна розділити на дві групи - помилки, які впливають на сприйняття і загальну «красу» сайту, і помилки, через які верстальник неправильно створює сайт по готовому шаблону.

Помилки в дизайні

Коли на сайті економлять, то наймають дешевого дизайнера або зовсім доручають справу непрофесіоналу. Тоді з'являються типові помилки в дизайні, які псують вигляд шаблону сайту.

Уникайте цих помилок:

- Незбалансована колірна гамма або непоєднуваного шрифти
- Надлишок елементів - занадто багато блоків, кнопок, написів. Відвідувач загубиться на такій сторінці і не виконає цільову дію.
- Відсутність мобільної версії сайту.

Помилки, критичні для верстки

Іноді з шаблоном сайту все добре, а при верстці вилазять недоліки. Це відбувається, тому що залишилися непомітні помилки, які верстальник не розуміє і переносить «як є». Ось чого треба уникати, щоб шаблон був зверстаний ідеально:

- **Безлад в шарах.** Потрібно видалити непотрібні шари - приховані, порожні. Інтуїтивно зрозуміло назвати все, що залишилися і логічно згрупувати. В ідеалі потрібно робити це на етапі розробки шаблону сайту, а не при здачі, інакше самі заплутаєтеся.

- **Ефекти прозорості та накладання.** Щоб зробити колір світліше, просто використовуйте інший колір, але ні в якому разі не прозорість. Також не застосовуйте ефекти накладання - вони непередбачувано відображаються в різних браузерах.

- **Елементи «гуляють» туди-сюди на пару пікселів.** Вирівнюйте все строго по сітці, інакше верстальник може просто обрізати об'єкт, який виступає за направляючу.

- **Плутанина з відступами.** Перевірте відступи - вони повинні бути виражені цілим парним числом, щоб верстальщику було простіше їх перенести.

- **Недокомплект файлів.** Додайте шрифти і всі зображення до шаблону - окремий архів для шрифтів, окремий для зображень. Якщо шрифти є на Google Fonts, можете дати посилання на них.

- **Рандомні розміри відступів, шрифтів, інших елементів, особливо дробові.** Старайтесь, щоб контекстно зв'язана інформація помістилася в один екран пристрою перегляду. Старайтесь розміри вказувати кратні 1% ширини полотна. Тоді це дозволить при верстці сторінки точніше задавати їх розмір у відносних одиницях ширини [vw](#) (viewport), де 1vw дорівнює 1% ширини пристрою виводу і створювати резинову верстку сайту.

- **Зображення і шрифти не додано до шаблону окремими файлами.** Їх витягування додає верстальщику додаткової роботи, результат якої не завжди відповідатиме задумці веб-дизайнера

Поради:

Поєднуйте всі шари одного зображення. З кожного видимого шару у ГР у створюваній HTML верстці сайту виходить окремий HTML/CSS елемент. Якщо в шаблоні для створення одного зображення (такого як логотип/бекграунд/т.п.) використовується кілька шарів, об'єднайте шари в групу і позначте групу тегом #merge. Або можна конвертувати шари в Smart Object.

Не запобігайте генерації CSS3 стилів для шарів. При створенні шаблону використовуйте більше 'Shape', 'Layer mask' та 'Clipping Mask'. При цьому використовуйте для шарів тільки ті ефекти, які можна перетворити на CSS3 (усі, крім: 'Satin', 'Bevel & Emboss', 'Pattern Overlay'). Чи не растеризуйте шари!

Позначте шари тегами відповідно до документації. Якщо з шару або групи шарів у ГР Ви хочете отримати в HTML верстці сайту посилання, форму введення, таблицю, список тощо, позначте шар або групу відповідним тегом.

Лекція 5. Створення веб-макету. Grid модуль в CSS

Мета: здійснити ознайомлення студентів з Grid модулем в CSS та адаптивною версткою вебсторінок.

Grid модуль в CSS

Grid модуль в CSS був розроблений CSS Working Group для того, щоб надати найкращий спосіб створення шаблонів в CSS. Він потрапив в Candidate Recommendation в лютому 2017 року, а основні браузерери почали його підтримку в березні 2017 року.

Вступ

CSS Grid це нова модель шаблону, оптимізована для двовимірних шаблонів. Це ідеальна модель для шаблонів сайтів, форм, галерей і всього, що вимагає точного і чутливого позиціонування.

У ранні роки Інтернету, HTML таблиці часто використовувалися для багатоколонних шаблонів, форм і т.д. Але у цього методу є свої недоліки. Це означає те, що представлення мало бути зроблено на рівні розмітки, таким чином не було відмінностей між представленням і контентом. Крім того, таблиці були створені для того, щоб утримувати табличні дані, а не для створення шаблонів. І не торкаючись семантичних проблем, табличні шаблони не створені для адаптивного дизайну.

[Флоати](#) (плаваючі блоки) в кінцевому підсумку замінили табличні шаблони як загально прийнятний і рекомендований метод створення шаблонів, так як він дозволяє нам позиціонувати елементи незалежно від розмітки. Проте, поки цей метод вважався значним поліпшенням на тлі табличної верстки, у нього теж були свої обмеження. Флоати в основному були розроблені для шаблонів документів і вони не були придатними для складних шаблонів додатків, які зараз поширені в мережі. Флоат-елементи складно контролювати, особливо на девайсах різних розмірів. Це призвело до розвитку різних фреймворків, зокрема одного з найпопулярніших - **Bootstrap**. Більшості з них потрібна додаткова розмітка, яка відволікала від всієї концепції поділу контенту. Таким чином CSS Working Group була в пошуку кращого рішення.

Потім з'явився [flexbox](#) — режим верстки, спеціально розроблений для створення надійних адаптивних сторінок. Flexbox спростив правильне вирівнювання елементів та їх вмісту і став найкращою системою CSS для більшості веб-розробників.

Але тепер ми маємо нового претендента за звання «найкращої системи для верстки макетів HTML». Це CSS Grid. CSS Grid модель дозволяє створювати складні шаблони за відносно короткий час і меншого розміру коду.

У той час як [flexbox](#) вже дозволив розробникам почати рухатися від флоат-елементів, flexbox тільки працює в одному вимірі. Grid CSS же це робить в двох, таким чином краще підходячи для створення складних шаблонів.

[Порівняння CSS grid та Flexbox](#)

Як це працює?

Grid-шаблон працює за системою сіток. Grid це набір пересічених горизонтальних і вертикальних ліній, які створюють розміри і позиціонують систему координат для контенту в grid-контейнері.

Щоб створити Grid, вам просто потрібно виставити елементу властивість **display: grid**. Це автоматично зробить всіх прямих нащадків цього елемента grid-елементами. Ви можете використовувати різноманітні grid властивості для вирівнювання, задання розміру,

позиціонування і т.д. Зазвичай першим кроком є визначення того, скільки колонок і рядків є в grid.

1	2	3
4	5	6
7	8	9
10	11	12

Це приклад ґріда (сітки) з чотирма рядками і трьома колонками. Він складається з 12 grid елементів. Кожен з цих елементів зелений і між ними є невелика відстань.

Всі ці grid елементи одного розміру, але вони можуть бути будь-якого розміру, якого ми захочемо. Деякі можуть охоплювати кілька стовпців і рядків, інші можуть залишатися розміром з одне відділення.

Створюємо Grid

Ось приклад простого 3 × 3 ґріда з невеликими проміжками між сітками.

1	2	3
4	5	6
7	8	9

А ось код:

```
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
  <title> Example </title>
</style>
#grid {
  display: grid;
  grid-template-rows: 1fr 1fr 1fr;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 2vw;
}
#grid > div {
  font-size: 5vw;
  padding: 0.5em;
  background: gold;
  text-align: center;
}
</style>
</head>
<body>
<div id = "grid">
  <div> 1 </div>
  <div> 2 </div>
  <div> 3 </div>
```



```
<div> 4 </div>
<div> 5 </div>
<div> 6 </div>
<div> 7 </div>
<div> 8 </div>
<div> 9 </div>
</div>
</body>
</html>
```

Як бачимо, HTML розмітка для CSS Grid виглядає так як звичайний HTML, який складається з елементів вкладених у зовнішній елемент. Але саме для наших цілей, зовнішній `<div>` це контейнер грід. Відповідно, всі елементи вкладені в нього будуть грід елементами (елементами сітки).

Але по-факту, це не буде повноцінними грідом, поки ми не застосуємо певний CSS для нього. Ось код, який створює його:

```
#grid {
display: grid;
grid-template-rows: 1fr 1fr 1fr;
grid-template-columns: 1fr 1fr 1fr;
grid-gap: 2vw;
}
```

Це правило застосовується до зовнішнього `<div>`, так як йому було призначено ID `#grid`.

Ось пояснення того, що написано в цьому CSS:

```
display: grid
```

Перетворює елемент в grid контейнер. Це все, що потрібно для того, щоб створити грід. Тепер у нас є грід-контейнер і грід-елементи. Значення грід створює блоковий контейнер. Ви також можете використовувати `display: inline-grid`, щоб створити рядковий грід-контейнер. Або ж ви можете використовувати `display: subgrid`, щоб створити підсітку, це значення використовується на самих grid елементах.

`fr` (fractional unit) – це нова форма одиниці вимірювання, в основному асоціюється з CSS Grid. Найближчий переклад fractional unit в контексті Grid розробки і застосування `fr` це одиниця гнучкості (дослівний: дробова одиниця). Якщо ви вказуєте ширину в `1fr`, то не залежно від кількості елементів, ширина і висота кожного елемента буде рівномірно розділена серед дочірніх елементів. Кількість може бути і дробовою.

Нехай загальна сума часток дорівнює $1.5fr + 3fr + 4.5fr = 9fr$. Якщо батьківський контейнер має ширину `900px`, перше значення `1.5fr` матиме ширину з розрахунку $(900px/9fr)*1.5fr=150px$, друге – `300px` і т. д. Використання одиниць гнучкості надають великі можливості в розробки гнучких макетів.

```
grid-template-rows: 1fr 1fr 1fr
```

Вибудовує рядки в гріді. Кожне значення представляє розмір ряду. В цьому випадку всі значення рівні `1fr`. Замість `fr`, звичайно, можна було б використовувати різні значення, такі як `100px`, `7em`, `30%` і так далі, також можна комбінувати різні одиниці розміру. Ви також можете призначати імена рядкам разом з їх розмірами.

```
grid-template-columns: 1fr 1fr 1fr
```

Теж саме, що і вище, тільки визначає колонки в грідах.

```
grid-gap: 2vw
```

Виставляє розрив, тобто відстань між грід елементами. Тут використано відносну одиницю ширини [vw](#) (Viewport width, viewport - це видима користувачеві область веб-сторінки, це те, що може бачити користувач, не вдаючись до прокручування), але також можна використовувати `10px`, `1em` і т. д. Один `vw` дорівнює 1% ширини пристрою виводу, `1em` рівний розміру шрифту. Властивість `grid-gap` заміняє використання одночасно двох властивостей: `grid-row-gap` (відстань між рядками) і `grid-column-gap` (відстань між колонками), і встановлює ці відстані однаковими.

Ну, а інша частина коду просто призначає різні стилі ґрид елементам.

```
#grid > div {  
font-size: 5vw;  
padding: 0.5em;  
background: gold;  
text-align: center;  
}
```

CSS-селектор `#grid > div` буде вибирати тільки безпосередні дочірні `div`-елементи контейнера із значенням `id=grid` і застосовувати до них властивості описані у фігурних дужках.

Якщо потрібно, щоб сітка була розтягнута на всю сторінку, то для селектора `#grid` можна додати наступне правило: `“height: 100vh;”`, яке визначає висоту сітки у 100% сторінки (viewport), тобто `1vh` дорівнює 1% висоти сторінки (viewport).

Функція `repeat()`

Ви можете використовувати функцію `repeat()` для повторюваних оголошень значення розміру елемента. Для прикладу, замість того, щоб задати це:

```
grid-template-rows: 1fr 1fr 1fr 1fr 1fr;
```

Ми можемо задати так:

```
grid-template-rows: repeat(5, 1fr);
```

Що значно скоротить кількість коду, особливо, якщо ви працюєте з великою кількістю повторюваних ґрид.

Створюємо шаблон сайту з CSS Grid

Ґриди включають в себе інтуїтивний синтаксис, в якому ви можете віртуально «бачити» шаблон в коді, по ньому стає легко створювати і змінювати ваш шаблон. Навіть значні зміни можуть бути зроблені за кілька секунд. Цей інтуїтивний синтаксис також допомагає з адаптивним веб-дизайном. Створення різних шаблонів для різних пристроїв стає досить дріб'язковою справою при використанні CSS Grid.

Давайте тепер створимо шаблон сайту, який виглядає таким чином:



А ось код цього шаблону:

```
<html>  
<head>  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<meta charset="utf-8">  
<title> Example </title>
```

```

<style>
body {
display: grid;
grid-template-areas:
"header header header"
"nav article ads"
"footer footer footer";
grid-template-rows: 60px 1fr 60px;
grid-template-columns: 20% 1fr 15%;
grid-gap: 10px;
height: 90vh;
margin: 0;
}
header, footer, article, nav, div {
padding: 20px;
background: gold;
}
#pageheader {
grid-area: header;
}
#pagefooter {
grid-area: footer;
}
#mainarticle {
grid-area: article;
}
#mainnav {
grid-area: nav;
}
#siteAds {
grid-area: ads;
}
</style>
</head>
<body>
<header id = "pageheader"> header </header>
<article id = "mainarticle"> article </article>
<nav id = "mainnav"> nav </nav>
<div id = "siteAds"> Ads </div>
<footer id = "pagefooter"> footer </footer>
</body>
</html>

```

Давайте ближче подивимося на наш код. Як бачимо, `<body>` стане грід-контейнером, всі інші елементи стануть грід-елементами.

Ця властивість визначає наш шаблон:

```

grid-template-areas:
"header header header"
"nav article ads"
"footer footer footer";

```

Просто дивлячись на код, ми можемо бачити, що це 3x3 грід (три ряди і три колонки). Таким чином у нас виходить п'ять грід областей на дев'яти грід осередках, так як деякі грід-області займають кілька осередків.

Ми так само бачимо, що шапка займає весь перший ряд в три осередки, а підвал займає весь нижній ряд, також займаючи три осередки. Навігаційна, тематична і рекламна секції ділять місце в другому ряду, де кожному з цих елементів дістається по одній клітинці.

Тепер ми можемо призначити кожному з цих грід-областей кожному елементу:

```
#pageheader {
  grid-area: header;
}
#pagefooter {
  grid-area: footer;
}
#mainarticle {
  grid-area: article;
}
#mainnav {
  grid-area: nav;
}
#siteAds {
  grid-area: ads;
}
```

Властивість `grid-area` це скорочення властивості, яка дозволяє вам розміщувати грід-елементи в гріді. У нашому випадку, ми просто відсилаємось до назв, які ми попередньо вказали в `grid-template-areas`.

Велика частина шаблону вже завершена. Решта коду відноситься до встановлення розмірів, відстаней, висот та декоративних властивостей.

Наступний код видає розміри по рядках і колонках:

```
grid-template-rows: 60px 1fr 60px;
grid-template-columns: 20% 1fr 15%;
```

Перший і третій рядки - обидва в `60px` заввишки, а другий рядок забирає все, що залишилося.

Перший стовпець дорівнює `20%`, а третій `15%`. Другий же забирає все, що залишилося.

Змінимо шаблон

Ви можете змінити шаблон просто перерозподіливши грід-області в `grid-template-areas`.

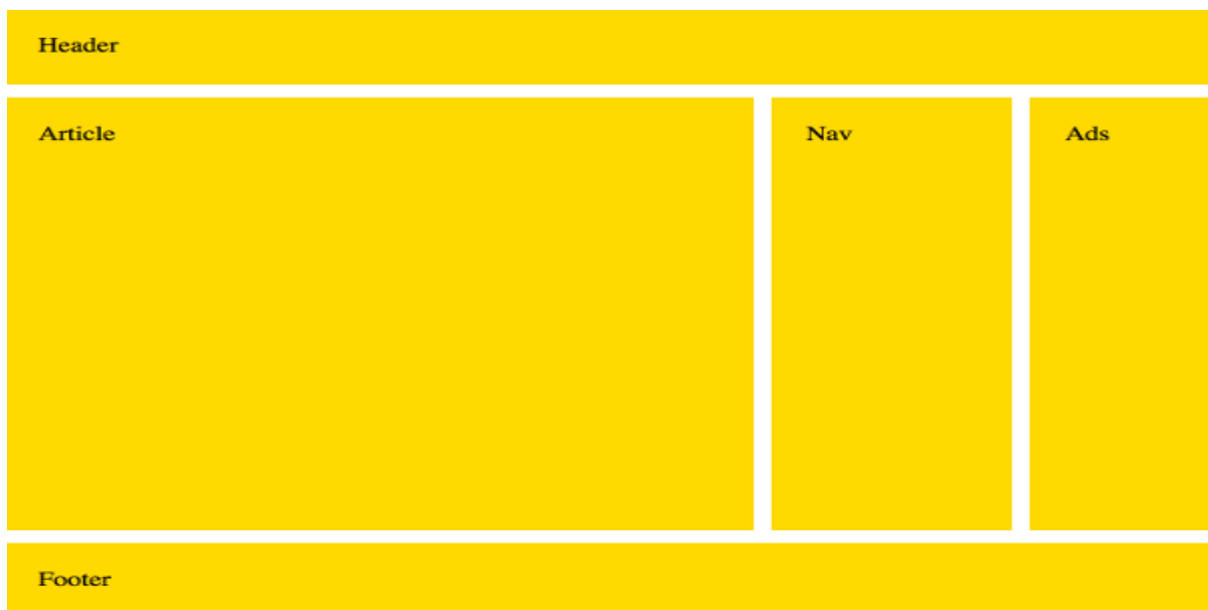
Таким чином, якщо ми змінимо на це:

```
grid-template-areas:
"nav header header"
"nav article ads"
"nav footer ads";
```

То в результаті отримаємо такий шаблон:



Проте, вам може знадобитися регулювання розмірів, якщо ви рухаєте ґрид поменше в місце побільше. Для прикладу, щоб зробити так:



Навігація тепер займає місце, де розташовувався контент, отже, розміри були відрегульовані. В іншому випадку, у нас був би вузький контент і широка навігація. Таким чином, код зараз виглядає таким чином.

```
grid-template-areas:  
"header header header"  
"article nav ads" /* До цього було "nav article ads" */  
"footer footer footer";  
grid-template-rows: 60px 1fr 60px;  
grid-template-columns: 1fr 20% 15%; /* До цього було '20% 1fr 15%' */
```

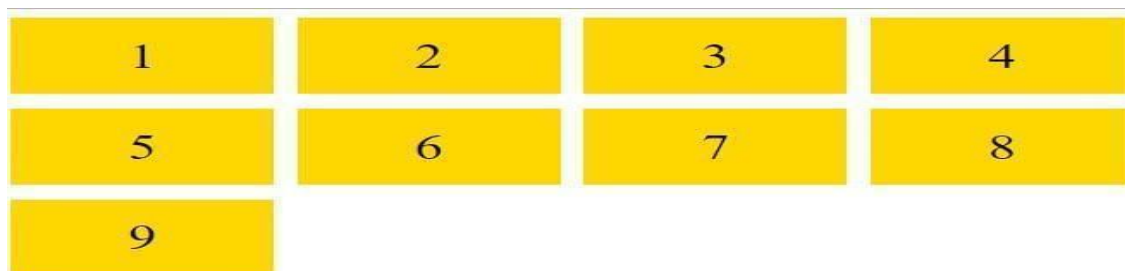
Створюємо адаптивний Grid

У шаблоні на ґрідах є такі значення як `auto-fill` і `auto-fit`, які дозволяють вам створювати ґрид з множиною треків певного розміру, які будуть міститися в контейнері. Вони дозволяють зробити ґрид адаптивним, тобто в ньому елементи можуть змінювати свої позиції в залежності від зміни розмірів вікна браузера.

Приклад використання `auto-fill`:

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta charset="utf-8">
<title> Example </title>
<style>
#grid {
display: grid;
grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
grid-gap: 2vw;
}
#grid> div {
font-size: 5vw;
padding: .5em;
background: gold;
text-align: center;
}
</style>
</head>
<body>
<div id = "grid">
<div> 1 </div>
<div> 2 </div>
<div> 3 </div>
<div> 4 </div>
<div> 5 </div>
<div> 6 </div>
<div> 7 </div>
<div> 8 </div>
<div> 9 </div>
</div>
</body>
</html>
```

Ось результат:



Змініть розмір браузера, щоб побачити, що відбувається. Також дослідіть поведінку елементів при зміні розміру вікна перегляду, клікнувши на `Toggle device toolbar` та `Style`

панелі DevTools браузера Google Chrome, яку можна викликати клікнувши на пункт “Перевірити” контекстного меню вікна браузера або F12.

Код, який відповідає за це:

```
grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
```

Функція `repeat()` повторює треки таку кількість разів, яка задана першим параметром. Використання `auto-fill` змусить треки повторюватися стільки разів, поки вони не заповнять контейнер, при цьому `auto-fill` намагається вмістити максимальну кількість треків мінімального розміру. Таким чином визначається їх максимальна кількість, а потім їх ширина встановлюється рівною ширині батьківського блоку поділеною на цю кількість треків.

Розмір цих треків вказано в другому параметрі. У нашому випадку, ми використовуємо `minmax(150px, 1fr)`, щоб вказати на те, що мінімальний розмір колонки дорівнює 150px, а максимальний 1fr. Таким чином, колонкам видається мінімальний розмір в 150px і максимум скільки залишилось місця. Такі треки будуть повторюватися стільки разів, скільки потрібно для того, щоб вкластися в контейнер.

Auto-fit

`auto-fit` працює практично так само, як і `auto-fill`. Різниця тут лише в тому, що `auto-fit` приховує всі порожні треки в кінці розміщення, а наявні розтягує на всю ширину батьківського блоку, в той час як `auto-fill` ні. Для кращого розуміння розглянемо приклад.

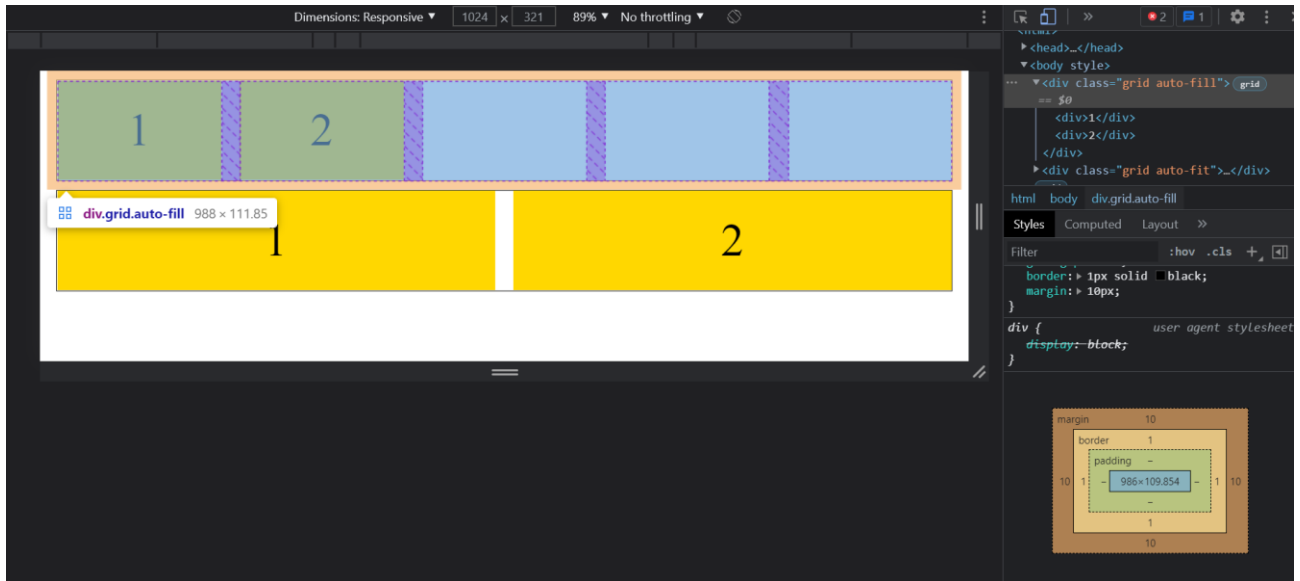
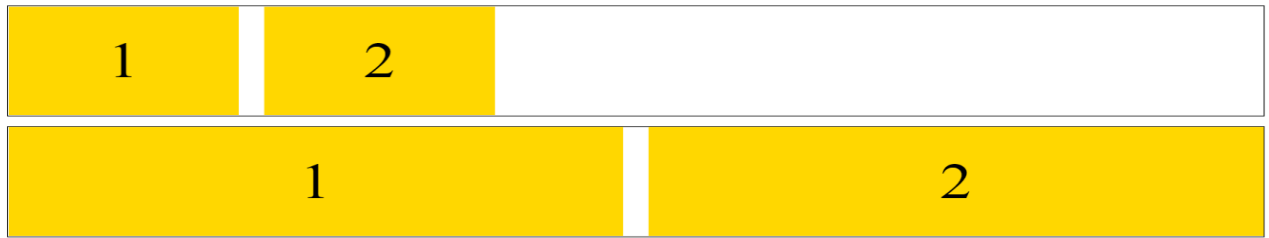
```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta charset="utf-8">
<title> Example </title>
<style>
.grid {
  display: grid;
  grid-gap: 2vw;
  border: 1px solid black;
  margin: 10px;
}
.grid > div {
  font-size: 5vw;
  padding: .5em;
  background: gold;
  text-align: center;
}
.auto-fill {
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
}
.auto-fit {
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
}
</style>
</head>
<body>
<div class="grid auto-fill">
  <div>1</div>
  <div>2</div>
</div>
<div class="grid auto-fit">
  <div>1</div>
  <div>2</div>
```



```

</div>
</body>
</html>

```



Як бачимо, auto-fill залишає порожні треки визначених розмірів в кінці, а auto-fit приховує порожні треки, що веде до заповнення треків розтягнутими елементами для заповнення простору.

Грід з медіа запитами

Однією з сильних сторін грид є те, що ви можете створити абсолютно інший шаблон за секунди. Це робить грід ідеальними для медіа запитів. Ми можемо просто змінити значення властивості grid-template-areas і вказати його в медіа запиті.

```

<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
  <title> Example </title>
  <style>
    header, footer, article, nav, div {
      padding: 1.2em;
      background: gold;
    }
    #pageheader {
      grid-area: header;
    }
    #pagefooter {

```

```

grid-area: footer;
}
#mainarticle {
grid-area: article;
}
#mainnav {
grid-area: nav;
}
#siteads {
grid-area: ads;
}

body {
display: grid;
grid-template-areas:
"header header header"
"nav article ads"
"footer footer footer";
grid-template-rows: 80px 1fr 70px;
grid-template-columns: 20% 1fr 15%;
grid-row-gap: 10px;
grid-column-gap: 10px;
height: 100vh;
margin: 0;
}

/* Stack the layout on middle devices /viewports. */
@media (max-width:800px) {
body {
grid-template-areas:
"header header"
"nav nav"
"article ads"
"footer footer";
grid-template-rows: 80px 70px 1fr 70px;
grid-template-columns: 1fr 15%;
}
}
/* Stack the layout on small devices /viewports. */
@media (max-width:575px) {
body {
grid-template-areas:
"header"
"article"
"ads"
"nav"
"footer";
grid-template-rows: 80px 1fr 70px 1fr 70px;
grid-template-columns: 1fr;
}
}
</style>
</head>

<body>

```

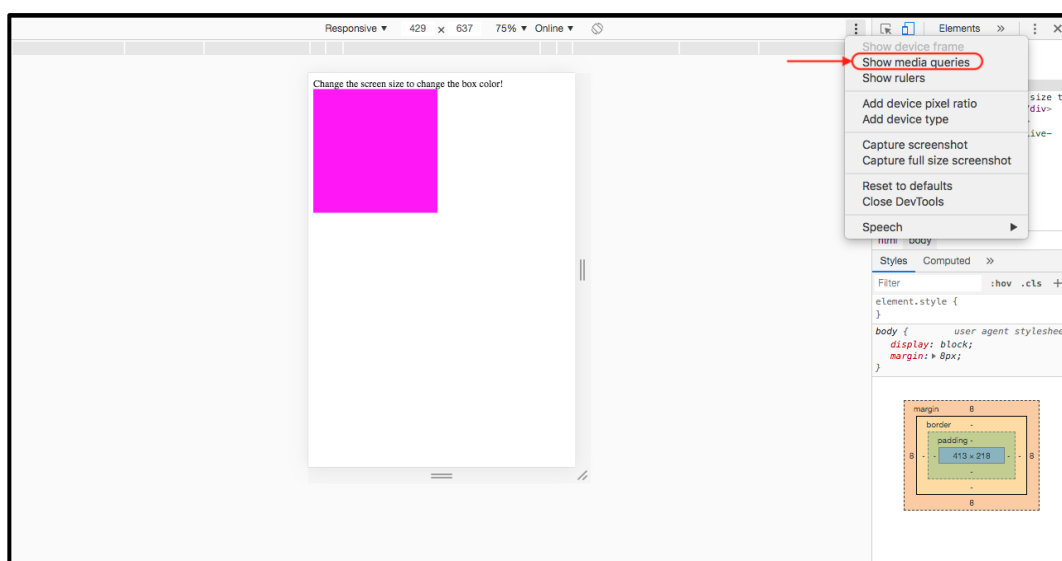
```
<header id = "pageheader"> header </header>
<article id = "mainarticle"> article </article>
<nav id = "mainnav"> nav </nav>
<div id = "siteads"> Ads </div>
<footer id = "pagefooter"> footer </footer>
</body>
</html>
```

Ось результат

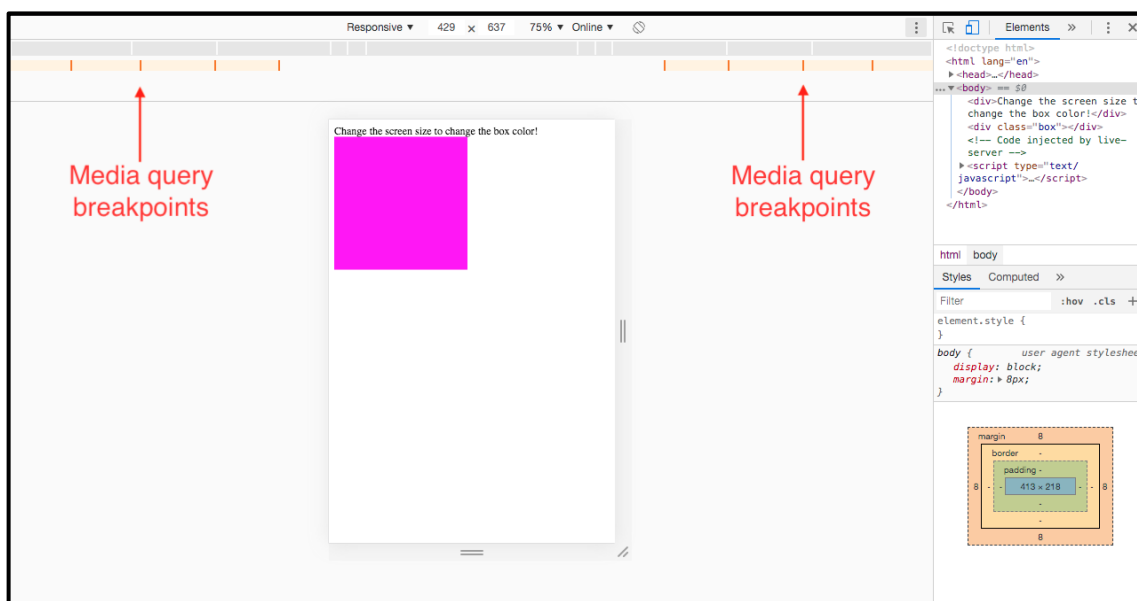


Це трьохколонний шаблон для пристрою виводу з високою роздільною здатністю, який перетворюється в одну колонку на пристроях з низькою роздільною здатністю. Зменшіть розміри вікна і подивіться як змінюється відображення.

Щоб дослідити з допомогою інсектора коду і побачити межі медіа запитів, натисніть на три вертикальні точки в крайньому правому куті панелі інструментів пристрою. Ви побачите багато доступних опцій. Нас цікавить *Показати медіа-запити (toggle device toolbar)*.



Натисніть цей параметр, і на панелі інструментів з'являться смуги з позначками меж медіа запитів. Якщо ви їх не бачите, спробуйте зменшити масштаб до 75% або й менше. Ви повинні побачити щось подібне:



Подивіться як змінюється відображення сторінки при зміні її ширини в залежності від потрапляння в певний діапазон. Можете також по чергово клікати на відповідну частину слуг.

Таким чином, цей приклад буде виглядати по-іншому в залежності від розміру екрана.

Вся справа полягає в перепризначенні значень у властивості `grid-template-areas`. Зверніть увагу, що ми також підкоригували значення в `grid-template-rows` і в `grid-template-columns`, щоб вони підходили під новий шаблон. Зокрема, тут повинна бути тільки одна колонка і вона повинна займати весь вільний простір. А так як всі грід елементи будуть йти один під одним, ми призначили 5 рядків і визначили їх висоти.

Поєднуємо `grid` с `block`

Залежно від вимог до вашого шаблону, можна змінити мобільну версію на `display: block`. Як тут:

```
@media (max-width: 575px) {  
  body {  
    display: block;  
  }  
}
```

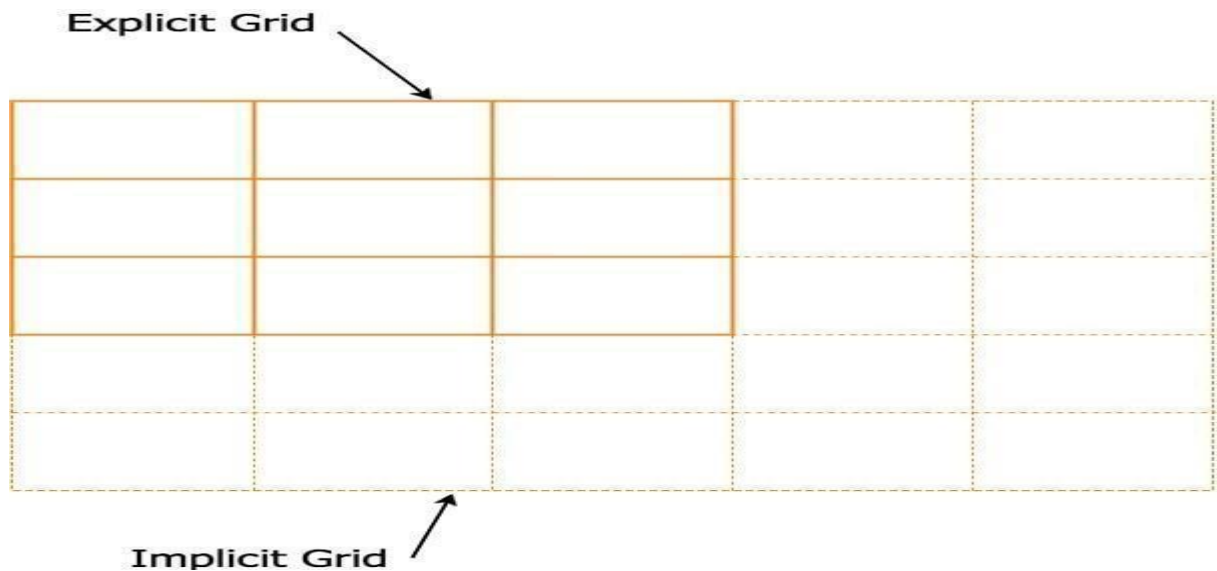
Це буде працювати також як і в прикладі вище, але по-дефолту, елементи встануть за своїм вихідним порядком. В наведеному вище прикладі, мобільна версія має `nav` під `ads`, але якби ми використовували `display: block`, то `nav` був би вищим `ads`.

Використовуючи цей метод, вам також може знадобитись додавання декількох зовнішніх відступів для компенсації відсутності розривів, які були включені у версію з грід.

Явні і неявні ґріди

CSS Grid використовує концепцію явного ґріда і неявного. Це ключова концепція, якої вам потрібно дотримуватися при створенні Ґрід, в іншому випадку під кінець ви побачите скупчення рядків і колонок, відображення яких ви не очікували.

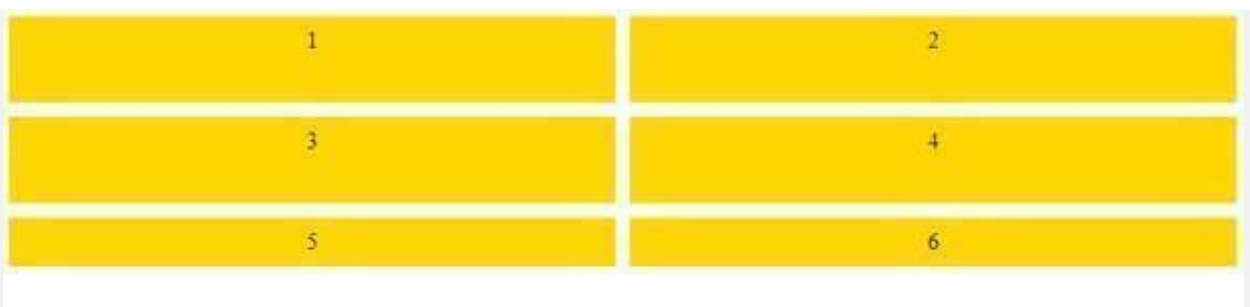
Явний ґрід, це ґрід, який ви визначаєте в `grid-template-rows`, `grid-template-columns` і в `grid-template-areas`.



Явні і неявні ґріди

Ви можете мати елементи, які не поміщаються в ваш «явно» визначений ґрід. Для прикладу, ви визначили ґрід, який може вмістити тільки шість елементів, але сам контейнер насправді складається з дев'яти елементів. Тільки шість елементів будуть розміщені в явний ґрід і три залишаться. І ось де починаються неявні ґріди.

Неявні ґріди автоматично генеруються ґрід-контейнером, кожен раз коли ґрід-елементи розташовуються за межами явного ґріда. Контейнер генерує неявні ґрід треки, додаючи неявні рядки в ґрід. Ці рядки разом з явними ґрід і формують неявні. Ось приклад:



```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta charset="utf-8">
<title> Example </title>
<style>
#grid {
display: grid;
grid-template-rows: 60px 60px;
grid-template-columns: 1fr 1fr;
```

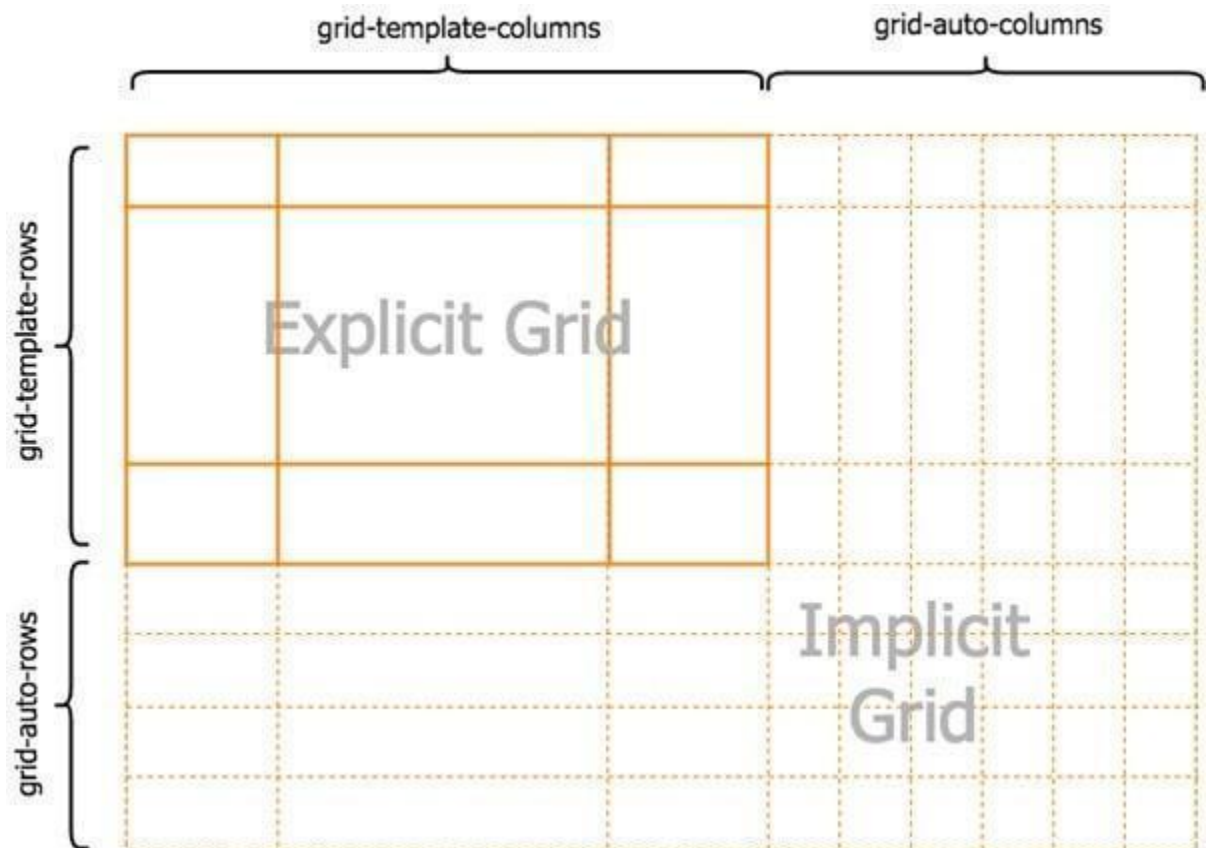
```
grid-gap: 10px;
}
#grid> div {
padding: .5em;
background: gold;
text-align: center;
}
</style>
</head>
<body>
<div id = "grid">
<div> 1 </div>
<div> 2 </div>
<div> 3 </div>
<div> 4 </div>
<div> 5 </div>
<div> 6 </div>
</div>
</body>
</html>
```

У цьому прикладі, ми явно визначаємо два рядки і дві колонки, що вмістять чотири гріди елемента. Проте, тут присутні шість грід елементів, таким чином був створений неявний грід, для того, щоб вмістити два додатковий елемент.

І це дуже добре. Якщо б не був створений неявний грід, то два додаткові елементи створили б повний безлад в сітці.

Виставляємо розмір треку для неявних грід

Можливо ви помітили, що додатковий ряд не такий заввишки, як попередні два. Це тому що ми виставили висоту рядка у властивості `grid-template-rows`, але застосовується вона тільки для явних грід. Висота рядка на неявних грідах повинна виставлятися за допомогою властивості `grid-auto-rows`. Але так як ми цього не зробили, виходить, що неявний ряд використовує розмір треку `auto`, який ґрунтується на контенті. Ось так розташовуються властивості виставлення розміру треку:



Загалом виходить так:

Явний грід використовує `grid-template-rows` і `grid-template-columns`

Неявний грід використовує `grid-auto-rows` і `grid-auto-columns`

У наступному прикладі ми зробимо явні і неявні рядки однієї висоти (`60px`).



```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#grid {
display: grid;
grid-template-rows: 60px 60px;
grid-template-columns: 1fr 1fr;
grid-gap: 10px;
grid-auto-rows: 60px;
}
#grid> div {
padding: .5em;
background: gold;
text-align: center;
}
```

```

</style>
</head>
<body>
<div id = "grid">
<div> 1 </div>
<div> 2 </div>
<div> 3 </div>
<div> 4 </div>
<div> 5 </div>
<div> 6 </div>
</div>
</body>
</html>

```

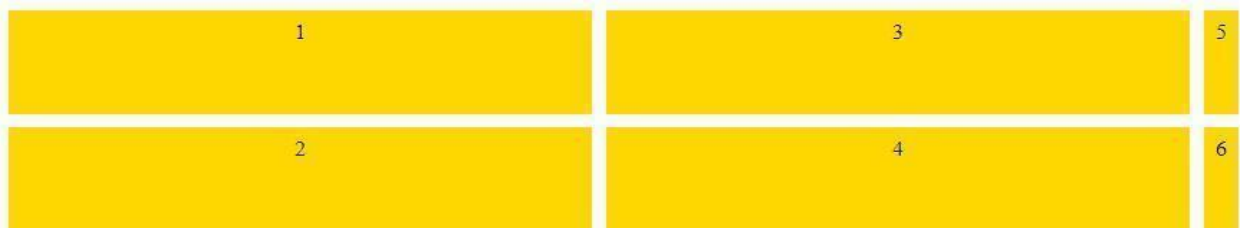
grid-auto-flow - рядки або колонки

Поки що ми створювали лише додаткові рядки, щоб укласти додаткові грід-елементи. Але що, якщо ми хочемо використовуватися додаткові колонки?

Для цього у нас є властивість **grid-auto-flow**. Це властивість дозволяє вказувати, що потрібно використовувати для авто-розміщення елементів: рядки або колонки. Іншими словами, ви можете вказати як буде рости неявний грід - у вигляді рядків або колонок. Початкове значення тут - row, що пояснює те, чому на прикладі вище були додані додаткові рядки, замість колонок. Але якщо ви віддасте перевагу використанню колонки, то ви можете використовувати це:

```
grid-auto-flow: column;
```

Ось що відбудеться, якщо ми застосуємо це до першого прикладу.



```

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#grid {
display: grid;
grid-template-rows: 80px 80px;
grid-template-columns: 1fr 1fr;
grid-gap: 10px;
grid-auto-flow: column;
}
#grid> div {
padding: .5em;
background: gold;
text-align: center;
}
</style>
</head>

```

```

<body>
<div id = "grid">
<div> 1 </div>
<div> 2 </div>
<div> 3 </div>
<div> 4 </div>
<div> 5 </div>
<div> 6 </div>
</div>
</body>
</html>

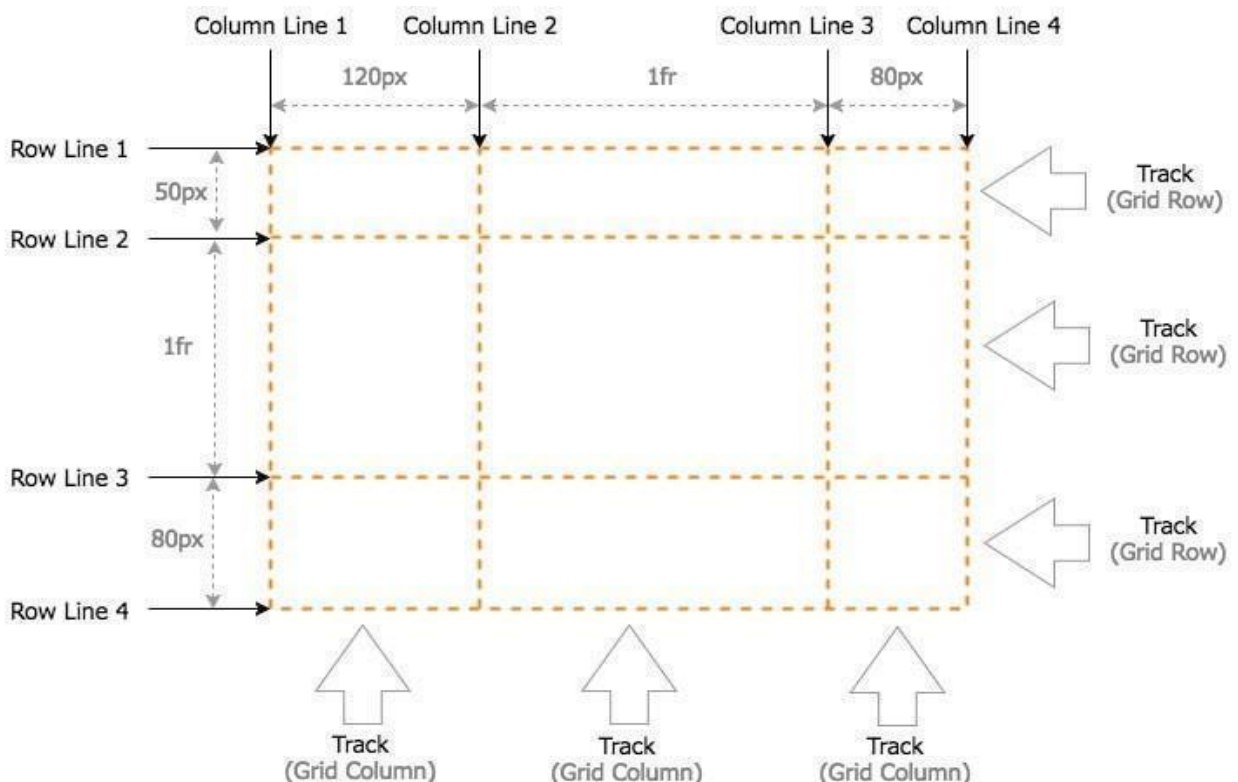
```

Само собою, ви можете спробувати піти далі в цьому прикладі і використовувати `grid-auto-columns` властивість, щоб змінити ширину автоматично згенерованої колонки. Так що якщо ви хочете, щоб всі колонки вище, були однакової ширини, вам треба використовувати `grid-auto-columns: 1fr`.

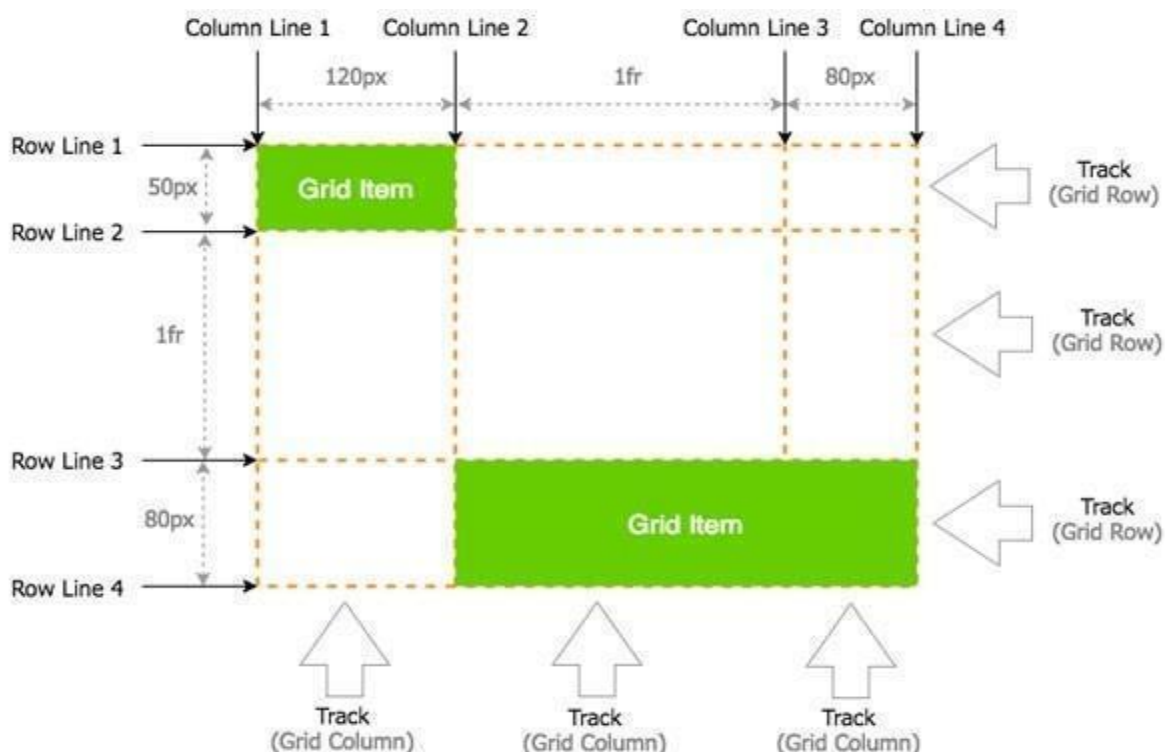
Майте на увазі, що виставляючи `column` для `grid-auto-flow` ви змінюєте потік грід елементів. Ви вже могли помітити, що грід елементи в прикладі вище розміщуються уздовж кожної колонки, замість кожного рядка.

Розміщення Grid-елементів

Розміщення грід-елементів в гріді зводиться до їх вирівнювання щодо грід-ліній.



Грід лінії це горизонтальні і вертикальні лінії в ґріді. Кожен рядок і колонка має ґрід лінію з кожної своєї сторони. У кожній ґрід лінії є числовий індекс, до якого ви можете звертатися під час розміщення ґрід елементів.



На прикладі на рисунку вище, перший ґрід елемент починається з рядкової лінії 1 і колонкової лінії 1, а закінчується на рядковій лінії 2 і колонкової лінії 2. Другий ґрід елемент починається на рядковій лінії 3 і колонкової лінії 2, а закінчується на рядковій лінії 4 і колонкової лінії 4.

Це можна зробити за допомогою наступного коду.

```
#item1 {
  grid-row-start: 1;
  grid-column-start: 1;
  grid-row-end: 2;
  grid-column-end: 2;
}
#item2 {
  grid-row-start: 3;
  grid-column-start: 2;
  grid-row-end: 4;
  grid-column-end: 4;
}
```



```

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#grid {
display: grid;
grid-template-rows: 50px 1fr 80px;
grid-template-columns: 120px 1fr 80px;
grid-gap: 5px;
height: 100vh;
}
#grid>div {
background-color: #66CC00;
}
#item1 {
grid-row-start: 1;
grid-column-start: 1;
grid-row-end: 2;
grid-column-end: 2;
}
#item2 {
grid-row-start: 3;
grid-column-start: 2;
grid-row-end: 4;
grid-column-end: 4;
}
</style>
</head>
<body>
<div id = "grid">
<div id = "item1"> </div>
<div id = "item2"> </div>
</div>
</body>
</html>

```

У цьому випадку нам не обов'язково вказувати кінцеві значення для першого елемента, тому що він охоплює тільки один трек. Теж саме можна сказати і про кінцеву рядкову лінію другого грід елемента, оскільки по замовчуванню, якщо ви не вказуєте кінцеву лінію, то елемент охопить тільки один трек.

Насправді, нам не потрібно було вказувати будь-яке позиціонування для першого грід елемента, так як він знаходиться в позиції по замовчуванню, то він був би там в будь-якому випадку. Але якби ми не вказали позиціонування для другого елемента, то він би перебував відразу за першим елементом займаючи лише 1 трек.

Використовуючи ключове слово `span` та ціле число можна задати скільки треків у рядку чи стовпці сітки буде охоплено елементом:

```
grid-column-end: span 2 ;
```

Якщо опущено ціле число, то за замовчуванням використовується значення 1. Негативні значення або значення 0 неприпустимі.

Іменування грід ліній

Ви також можете присвоїти імена грід лініям для полегшення звернення до них. Це можна робити виставляючи властивості `grid-template-rows` і `grid-template-columns`, наприклад як тут:

```
#grid {
display: grid;
/* Set the tracks and name the lines */
grid-template-rows: [row1-start] 50px [row2-start] 1fr [row3-start] 80px [row3-end];
grid-template-columns: [col1-start] 120px [col2-start] 1fr [col3-start] 80px [col3-end];
grid-gap: 5px;
height: 90vh;
}
...
/* Now refer to those named lines */
#item2 {
grid-row-start: row3-start;
grid-column-start: col2-start;
grid-row-end: row3-end;
grid-column-end: col3-end;
}
```

Іменні грід області

Грід області можна присвоїти ім'я у властивості `grid-template-areas` самого грід контейнера. Це те, що ми робили до цього, коли створювали шаблон сайту. Щоб відновити це в пам'яті, то це виглядає так:

```
#grid {
display: grid;
/* Іменні грід області */
grid-template-areas:
"a a"
"b c";
...
}
...
/* Тепер застосовуємо кожен грід елемент до іменованої грід області */
#a {
grid-area: a;
}
#b {
grid-area: b;
}
#c {
grid-area: c;
}
```

Ви можете вказати вільну позицію, використовуючи крапку (.) Або серію крапок без пробілів. Для прикладу:

```
grid-template-areas:
"a a"
". b";
або
```

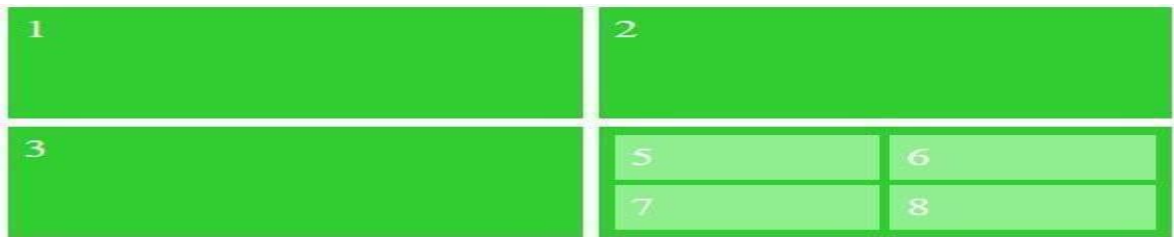


```
grid-template-areas:  
"header header"  
"... content";
```

Створюємо вкладений грід

Грід елементи можуть самі ставати грідом (сіткою) в CSS Grid. Тобто ви зможете вкладати грід-елемент в інший грід-елемент, тим самим створюючи вкладений грід.

Щоб створити такий вкладений грід, все що вам потрібно зробити - це застосувати `display: grid` (або `display: inline-grid`) до грід елементу і він сам стане грідом. Ви також можете використовувати `display: subgrid` для створення підгріда. Подивіться на прикладі, як це відбувається.



```
<html>  
<head>  
<meta charset="utf-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title> Example </title>  
<style>  
#outer-grid {  
display: grid;  
grid-template-rows: 1fr 1fr;  
grid-template-columns: 1fr 1fr;  
grid-gap: 8px;  
}  
#outer-grid > div {  
background-color: limegreen;  
color: white;  
font-size: 4vw;  
padding: 8px;  
}  
#inner-grid {  
display: grid;  
grid-template-columns: 1fr 1fr;  
grid-gap: 5px;  
}  
#inner-grid > div {  
background: lightgreen;  
padding: 8px;  
}  
</style>  
</head>  
<body>  
<div id = "outer-grid">  
<div> 1 </div>  
<div> 2 </div>  
<div> 3 </div>
```

```

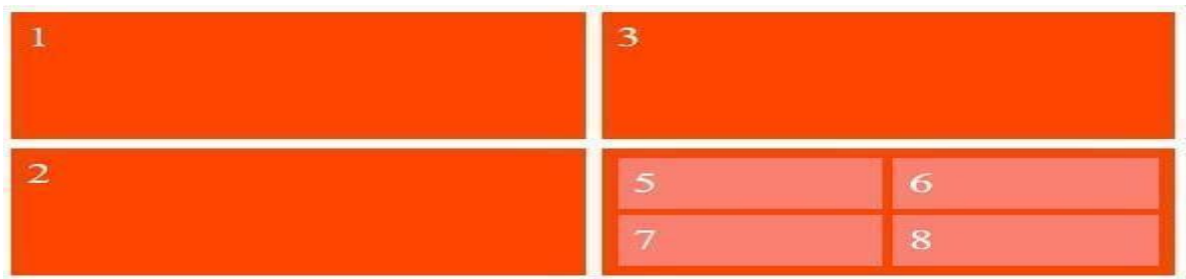
<div id = "inner-grid">
<div> 5 </div>
<div> 6 </div>
<div> 7 </div>
<div> 8 </div>
</div>
</div>
</body>
</html>

```

Успадкування

Більшість грид властивостей не успадковуються, це означає те, що ваш вкладений грид НЕ буде успадковувати значення свого батьківського гріда. Це дозволяє вносити зміни в батьківський грид без не навмисного впливу на вкладений грид.

Для прикладу, ви виставили `grid-auto-flow: column` на батьківському гріді, але ви не виставили властивість для вкладеного гріда. В цьому випадку, вкладеному гріду буде виставлено значення `row`, тому що це значення по замовчуванню для цієї властивості.



Як тут:

```

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#outer-grid {
display: grid;
grid-template-rows: 1fr 1fr;
grid-template-columns: 1fr 1fr;
grid-gap: 8px;
grid-auto-flow: column;
}
#outer-grid > div {
background-color: orangered;
color: white;
font-size: 4vw;
padding: 8px;
}
#inner-grid {
display: grid;
grid-template-columns: 1fr 1fr;
grid-gap: 5px;
}
#inner-grid > div {

```

```
background: salmon;
padding: 8px;
}
</style>
</head>
<body>
<div id = "outer-grid">
<div> 1 </div>
<div> 2 </div>
<div> 3 </div>
<div id = "inner-grid">
<div> 5 </div>
<div> 6 </div>
<div> 7 </div>
<div> 8 </div>
</div>
</div>
</body>
</html>
```

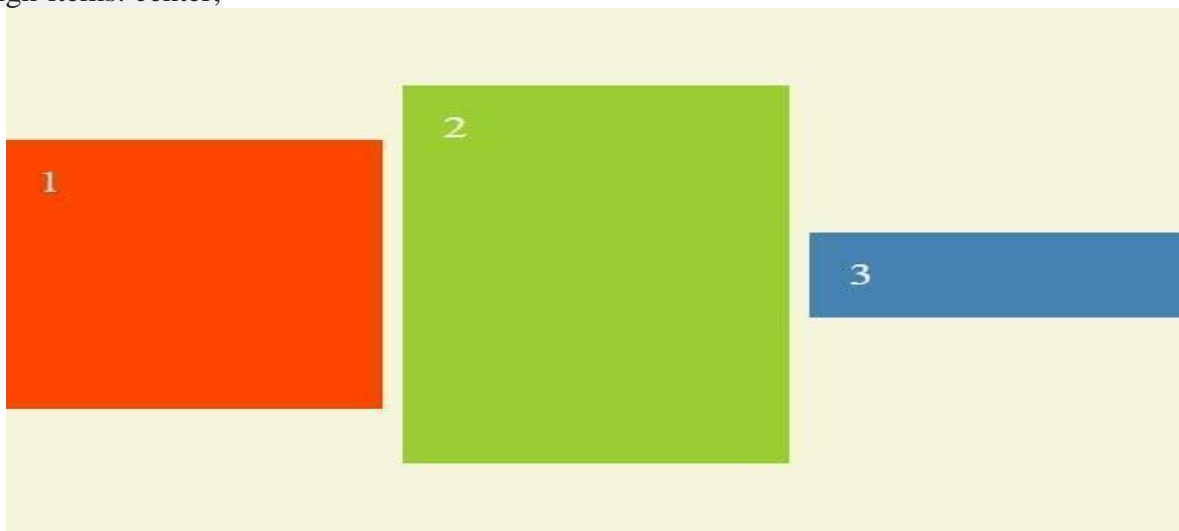
Зверніть увагу, що на батьківському гріді, числа йдуть вертикально вниз по колонках, замість того, щоб розташовуватися горизонтально уздовж рядка, але вкладений грід все таки йде в горизонтальному напрямку вздовж рядка.

CSS Grid вирівнювання

В основному, більшість властивостей вирівнювання працює також на грід елементах, як і на інших елементах. Але також є деякі властивості вирівнювання, які застосовуються тільки для грід і флексів.

Властивість `align-items` вказує стандартне значення `align-self` для всіх грід елементів, які беруть участь в грід контейнері в контексті його форматування.

```
align-items: center;
```



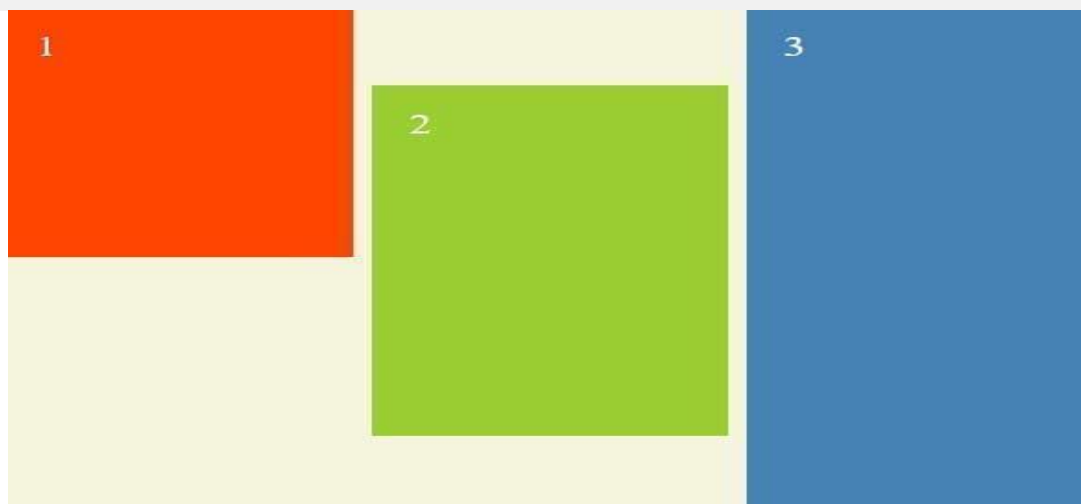
На прикладі вище, ми застосовуємо `align-items: center` для грід контейнера, отже все грід елементи будуть вирівняні по центру блокової осі.

Але так як це значення по замовчанню, будь-який з грід елементів може переписати його за допомогою `align-self` властивості.

Align-self властивість

Це властивість вирівнює елемент всередині блоку контейнера уздовж блокової /колоночної /поперечної осей.

```
.red {  
background: orangered;  
height: 40%;  
align-self: baseline;  
}  
.green {  
background: yellowgreen;  
height: 60%;  
}  
.blue {  
background: steelblue;  
height: auto;  
align-self: stretch;  
}
```

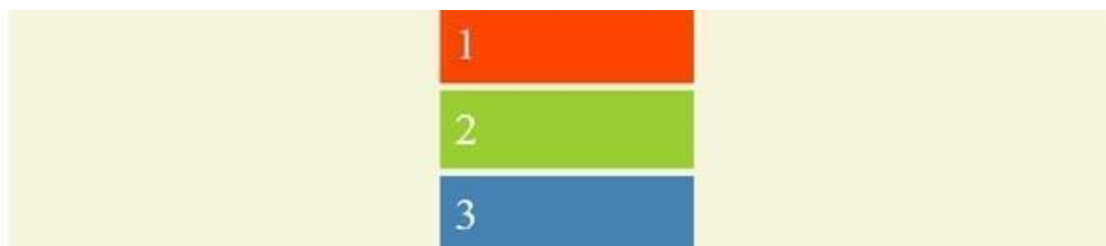


[Тут червоному елементу](#) виставлено значення `baseline`, а синій елемент має значення `stretch` (елемент розтягується та займає увесь доступний простір контейнера). Висота синього елемента дорівнює `auto`, таким чином він розтягується на всю висоту області ґрида.

Проте, ми не виставляли властивостей для зеленого елемента. Він використовує дефолтне значення, яке в нашому випадку дорівнює `center` в `align-items` ґрид контейнера.

Властивість justify-items

Це властивість вказує стандартне значення для `justify-self`-значення всіх ґрид елементів в ґрид контейнері.



```
justify-items: center;
```

Властивість justify-self

Це властивість може використовуватися для вирівнювання індивідуальних ґрид елементів вздовж малої /лінійної /головною осей.

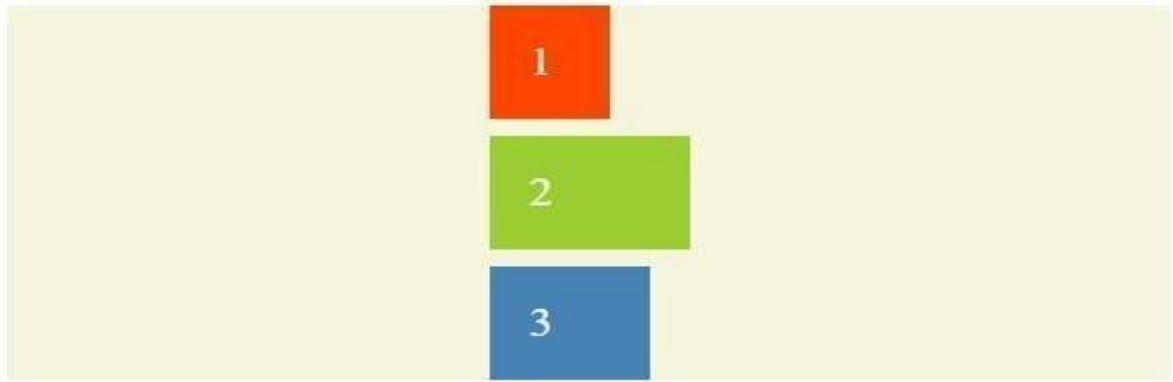


```
.red {  
background: orangered;  
justify-self: end;  
}  
.green {  
background: yellowgreen;  
}  
.blue {  
background: steelblue;  
justify-self: start;  
}
```

justify-content властивість

Властивість `justify-content` вирівнює контент гріда контейнера уздовж головної осі.

Його можна застосовувати для вирівнювання всього гріда в грід контейнері, в тому випадку, якщо грід трек забирає менше місця, ніж його грід контейнер. Це може статися, якщо ви виставили розмір треку в абсолютній величині, наприклад в пікселях, в той час як грід контейнер бере більше місця, ніж всі треки разом.



```
justify-content: center;
```

Властивість align-content

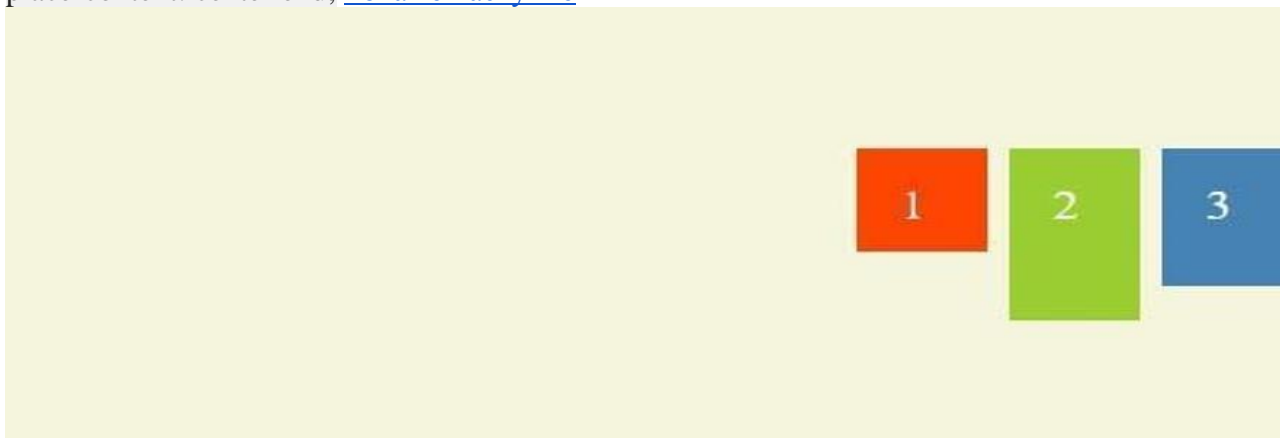
Це властивість схоже з `justify-content`, крім того, що це властивість вирівнює уздовж поперечної /блокової осей.



```
align-content: center;
```

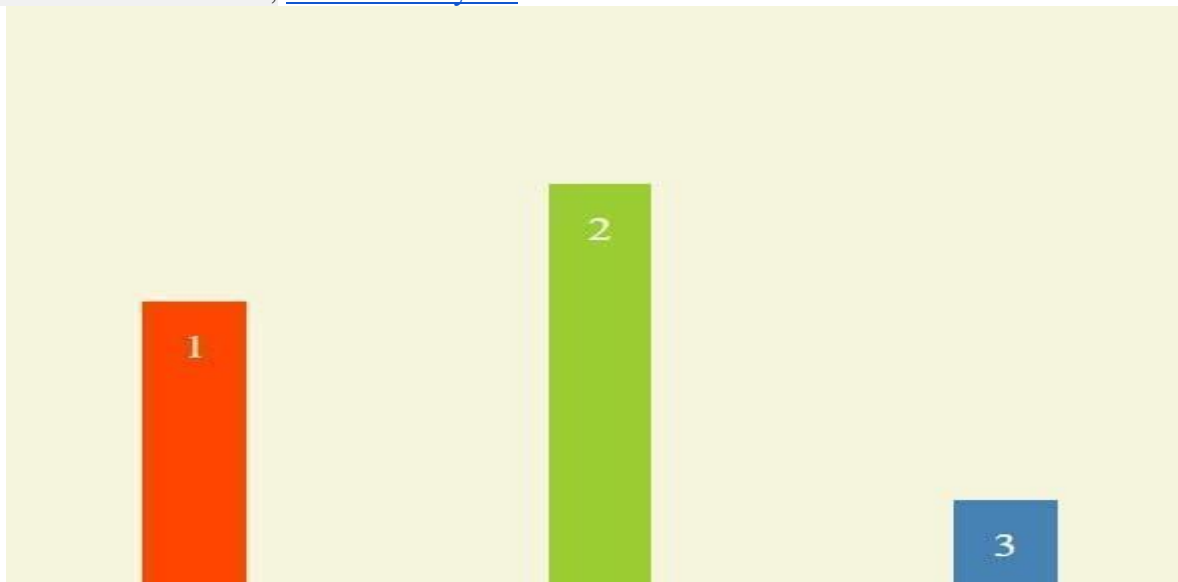
Властивість place-content

Ця властивість є скороченням для justify-content і align-content.
place-content: center end; [покаже наступне](#)



Властивість place-items

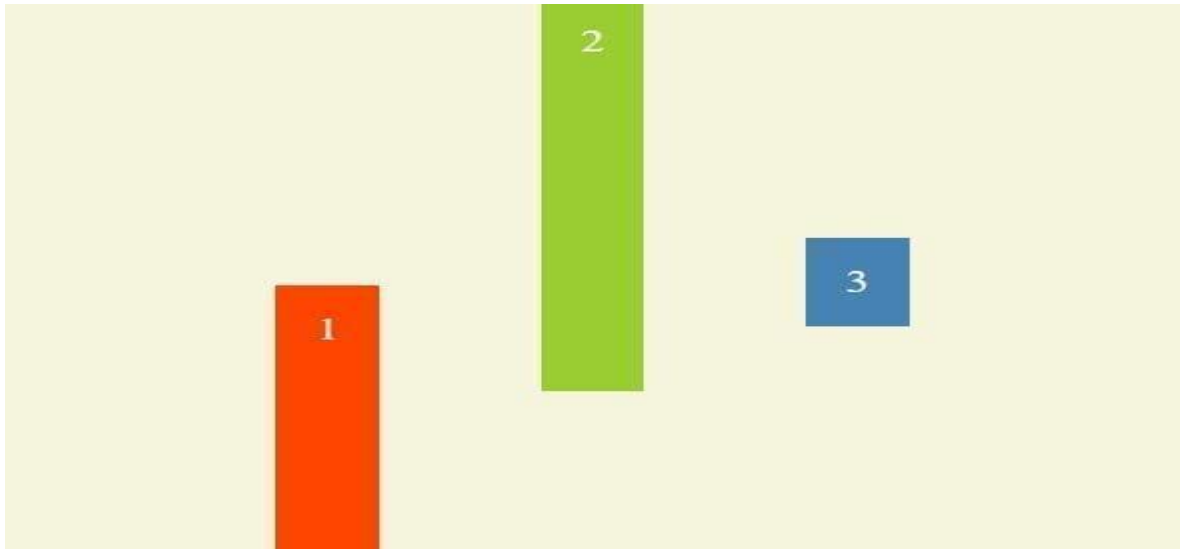
Ця властивість є скороченням для justify-items і align-items.
place-items: end center; [покаже наступне](#)



Властивість place-self

Ця властивість є скороченням для justify-self і align-self

```
.red {  
background: orangered;  
height: 40%;  
place-self: end;  
}  
.green {  
background: yellowgreen;  
height: 60%;  
place-self: start center;  
}  
.blue {  
background: steelblue;  
height: auto;  
place-self: center start;  
}
```

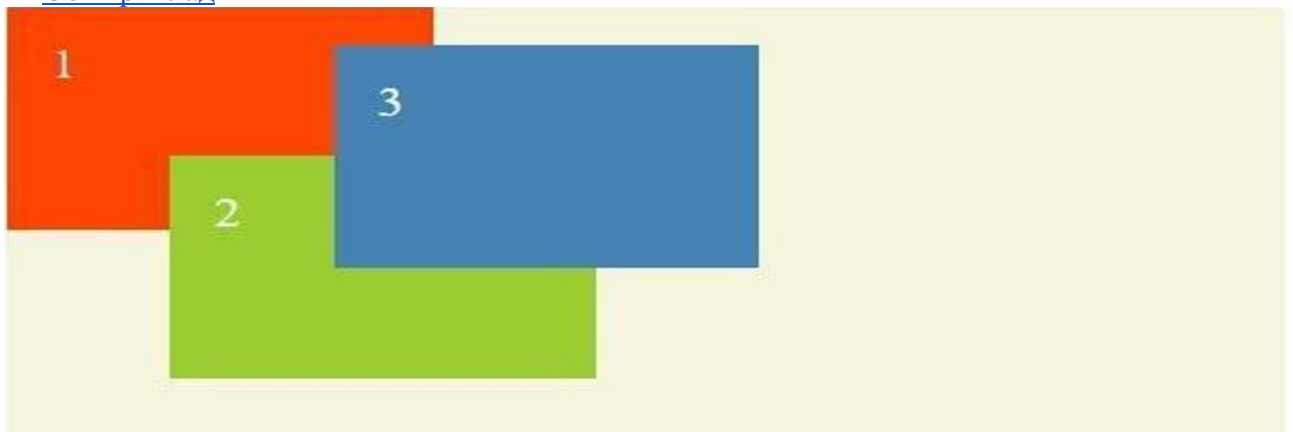
Накладення Grid елементів

Використання шаблонів CSS на ґрідах дозволяє накладати ґрид елементи один на одного за допомогою перетину ґрид областей, негативних зовнішніх відступів і інших технік позиціонування. Коли відбувається накладення, елементи накладаються один на одного в заданому порядку або відповідно до z-index.

Простий порядок

Якщо ви явно не використовували z-index для будь-якого ґрид елемента, то будь-які накладені один на одного елементи будуть розташовані ґрунтуючись на простому розташуванні елементів в документі. Так, саме так. Вони будуть розташовуватися в тому порядку, в якому вони розташовуються в розмітці.

[Ось приклад](#)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>z-index</title>
</head>
<style>
#grid {
```

```

display: grid;
grid-template-columns: 1fr 1fr 1fr;
grid-template-rows: 1fr 1fr 1fr;
height: 90vh;
background-color: beige;
}
.red {
background: orangered;
}
.green {
background: yellowgreen;
position: relative;
left: -20vw;
top: 20vh;
}
.blue {
background: steelblue;
position: relative;
left: -40vw;
top: 5vh;
}
#grid > div {
padding: 20px;
font-size: 4vw;
color: white;
}
</style>
<body>
<div id="grid">
<div class="red">1</div>
<div class="green">2</div>
<div class="blue">3</div>
</div>
</body>
</html>

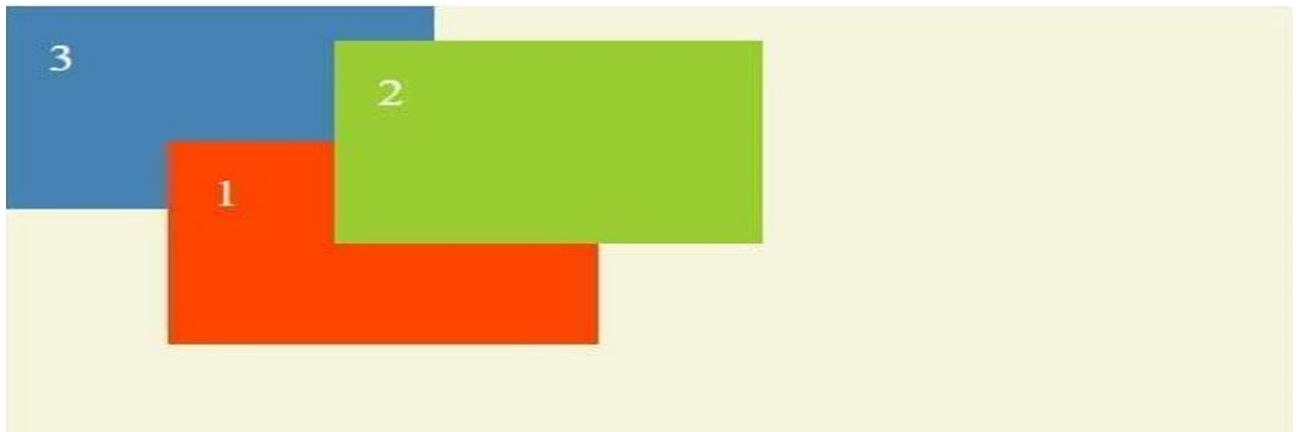
```

Змінений порядок в документі

Цей термін застосовний тоді, коли порядок кожного грід елемента змінюється після призначення своїх значень елементам. Як і фактом, це буде звичайний вихідний код, якщо ви спеціально нічого не міняли в порядку.

Тим не менш, ви можете використовувати властивість `order`, для того, щоб розставити елементи так як вам потрібно.

Використовуючи цю техніку, грід елементи накладаються в порядку зростання від нижчого. Ось приклад

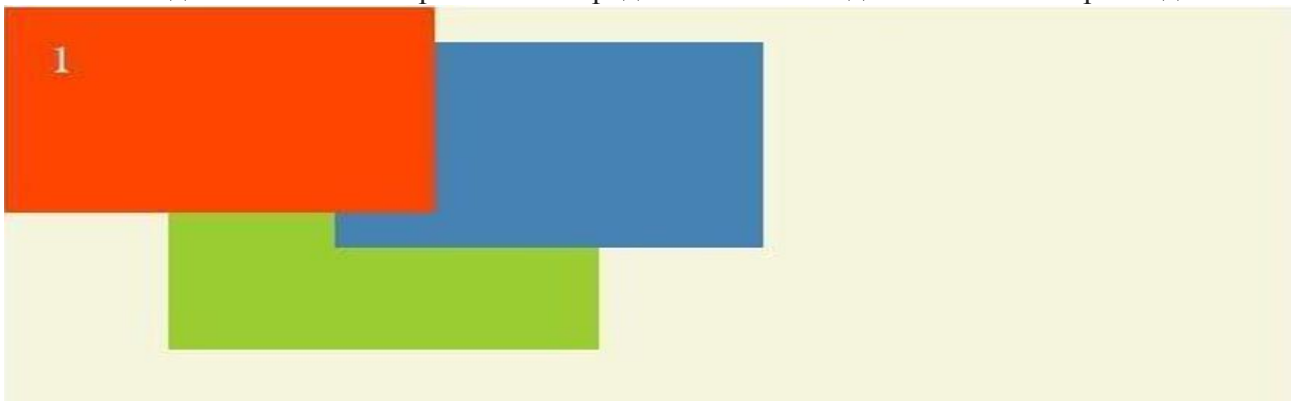


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<style>
#grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
  height: 90vh;
  background-color: beige;
}
.red {
  background: orangered;
  position: relative;
  left: -20vw;
  top: 20vh;
  order: 2;
}
.green {
  background: yellowgreen;
  position: relative;
  left: -40vw;
  top: 5vh;
  order: 3;
}
.blue {
  background: steelblue;
  top: 5vh;
  order: 1;
}
#grid > div {
  padding: 20px;
  font-size: 4vw;
  color: white;
}
</style>
```

```
<body>
<div id="grid">
<div class="red">1</div>
<div class="green">2</div>
<div class="blue">3</div>
</div>
```

Використовуємо z-index властивість

Так само можна використовувати властивість z-index щоб розкласти елементи. Ця властивість дозволяє вам контролювати порядок елементів вздовж z-осі. Ось приклад.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>z-index</title>
</head>
<style>
#grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
  height: 90vh;
  background-color: beige;
}
.red {
  background: orangered;
  z-index: 3;
}
.green {
  background: yellowgreen;
  position: relative;
  left: -20vw;
  top: 20vh;
  z-index: 1;
}
.blue {
  background: steelblue;
  position: relative;
  left: -40vw;
  top: 5vh;
  z-index: 2;
}
```

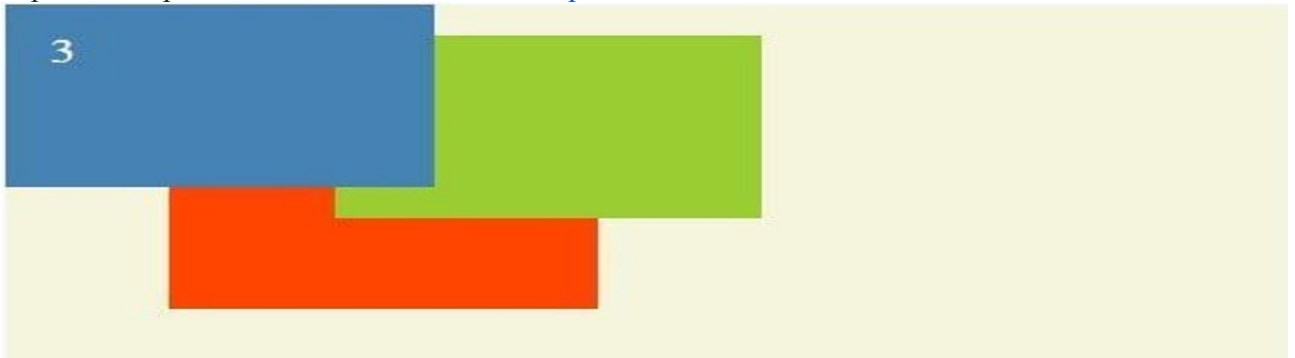
```

}
#grid > div {
padding: 20px;
font-size: 4vw;
color: white;
}
</style>
<body>
<div id="grid">
<div class="red">1</div>
<div class="green">2</div>
<div class="blue">3</div>
</div>
</body>
</html>

```

Комбінуємо z-index і order

Якщо у грід елементів був змінений порядок як на `z-index`, так і на `order`, то в порядку накладення `z-index` грає вирішальну роль. Це дозволяє вам змінювати порядок елементів без втрати контролю над накладенням. [Ось приклад.](#)



```

.red {
background: orangered;
position: relative;
left: -20vw;
top: 20vh;
order: 2;
z-index: 1;
}
.green {
background: yellowgreen;
position: relative;
left: -40vw;
top: 5vh;
order: 3;
z-index: 2;
}
.blue {
background: steelblue;
top: 5vh;
order: 1;
z-index: 3;
}
<div id = "grid">
<div class = "red"> 1 </div>

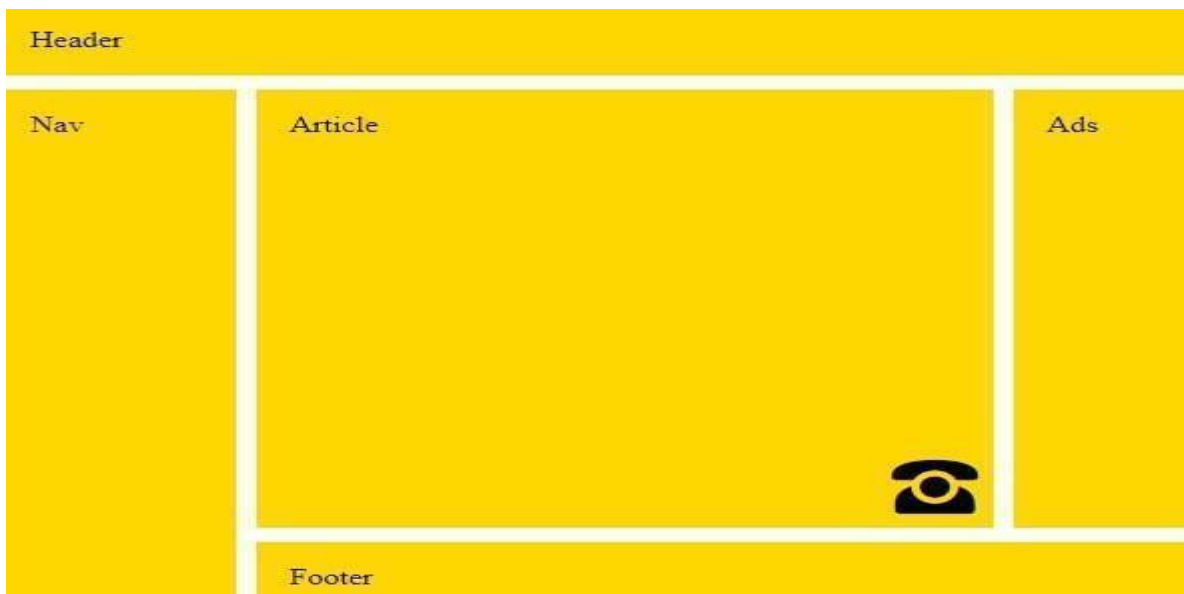
```

```
<div class = "green"> 2 </div>
<div class = "blue"> 3 </div>
</div>
```

Абсолютне позиціонування і ґриди

Прямо в ґридах ви можете використовувати абсолютне позиціонування, буквально так як і в блокової моделі. Якщо ви виставите `position: relative` для ґрид елемента і далі будете використовувати `position: absolute` для одного з його нащадків, то будь-яке позиціонування вказане вами для нього буде працювати в межах цього ґрид елемента.

[Ось наглядний приклад](#) ґрида з абсолютним позиціонуванням, де воно використовується для того, щоб розмістити іконку в лівому нижньому кутку ґрид елемента.



```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#grid {
display: grid;
position: relative;
grid-template-columns: 1fr 1fr;
grid-template-rows: 1fr 1fr;
height: 90vh;
background-color: beige;
}
#grid> div {
padding: 20px;
font-size: 4vw;
color: white;
text-align: center;
}
.red {
background: orangered;
grid-column: auto;
```



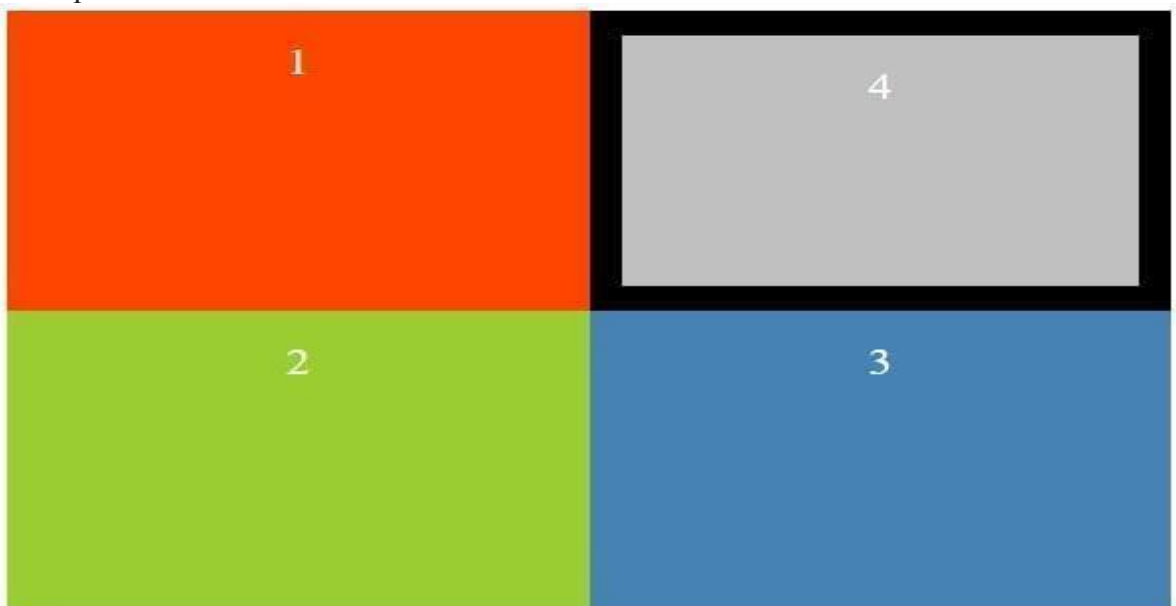
```

grid-row: auto;
}
.green {
background: yellowgreen;
grid-column: auto;
grid-row: auto;
}
.blue {
background: steelblue;
grid-column: auto;
grid-row: auto;
}
.gray {
border: 16px solid black;
background: silver;
grid-column: 2/3;
grid-row: 1/2;
}
</style>
</head>
<body>
<div id = "grid">
<div class = "red"> 1 </div>
<div class = "green"> 2 </div>
<div class = "blue"> 3 </div>
<div class = "gray"> 4 </div>
</div>
</body>
</html>

```

Але тим не менш, є кілька речей, які потрібно пам'ятати при позиціонуванні елементів в ґріді.

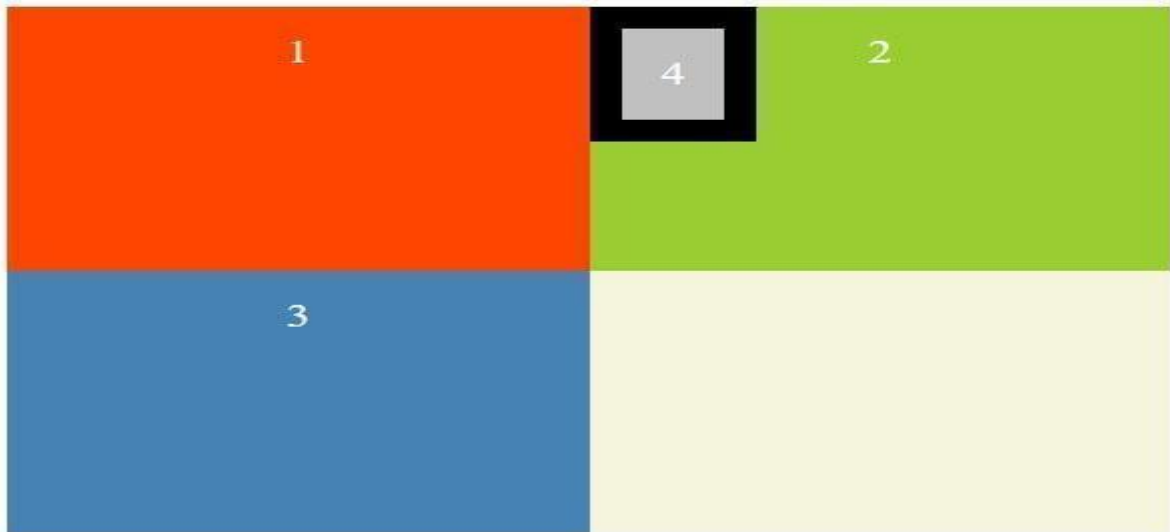
Абсолютно позиційований елементи не є частиною ґрід шаблону і вони забирають місце. Отже, вони не впливають на розташування інших ґрід елементів. Ось кілька прикладів для демонстрації.



[У цьому прикладі](#), явно вказано тільки бокс під номером чотири, у другій колонці першого рядка, а всі інші бокси мають параметри auto. Четвертий бокс фактично виштовхує

другий бокс зі свого слота - бокси 2 і 3 мають авторозміщення в наступних доступних грід областях. Отже, четвертий бокс вплинув на розстановку грід елементів таким чином.

Виставляючи четвертому боксу `position: absolute`, він [все одно буде знаходитися](#) на позиції другого боксу, але другий бокс також буде знаходитися на своєму місці недоторканим, також як і третій бокс.



```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Example </title>
<style>
#grid {
display: grid;
position: relative;
grid-template-columns: 1fr 1fr;
grid-template-rows: 1fr 1fr;
height: 90vh;
background-color: beige;
}
#grid> div {
padding: 20px;
font-size: 4vw;
color: white;
text-align: center;
}
.red {
background: orangered;
grid-column: auto;
grid-row: auto;
}
.green {
background: yellowgreen;
grid-column: auto;
grid-row: auto;
}
.blue {
background: steelblue;
grid-column: auto;
grid-row: auto;
}
```

```

}
.gray {
border: 16px solid black;
background: silver;
grid-column: 2/3;
grid-row: 1/2;
position: absolute;
}
</style>
</head>
<body>
<div id = "grid">
<div class = "red"> 1 </div>
<div class = "green"> 2 </div>
<div class = "blue"> 3 </div>
<div class = "gray"> 4 </div>
</div>
</body>
</html>

```

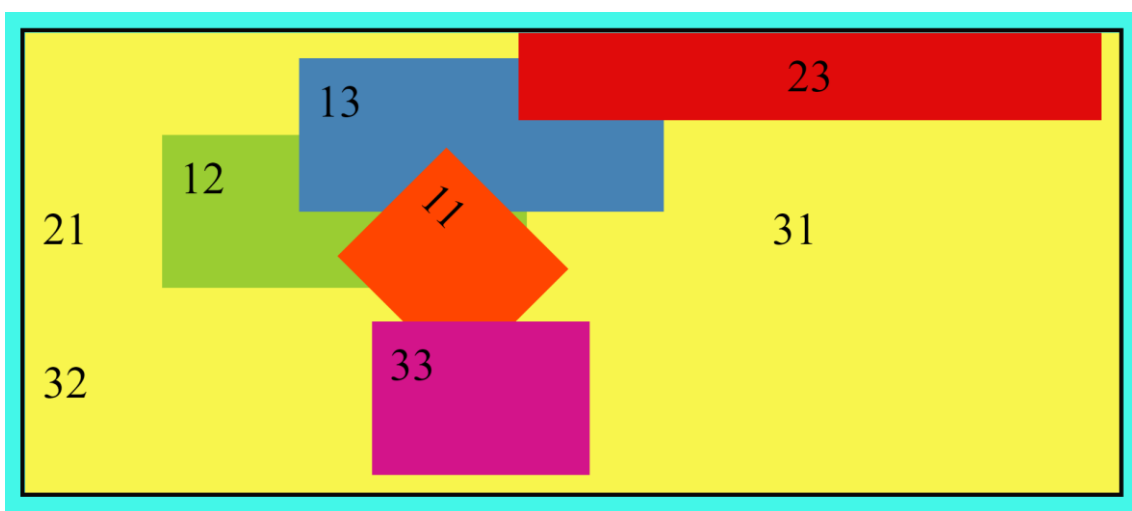
Спозиціоновані елементи відсікаються, щоб поміститися під свій контент.

Також варто пам'ятати, що позиційований елементи скорочуються для того, щоб вміщати своє утримання. Це ви можете чудово спостерігати на прикладі вище. Це відрізняється то того, що роблять звичайні гріди елементи. Останні в свою чергу розтягуються, щоб поміститися в свою гріди область, так що якщо ви вкажете, щоб гріди елемент займав дві колонки і два рядки, він так і буде робити.

На нормальних гріди елементах, значення `auto` для властивості розміщення гріди виставлено `span 1`, але на абсолютно позиційованих елементах воно накладається на внутрішній відступ гріди-контейнера, що може привести до дивацтв у верстці, якщо ви це не врахували.

Пам'ятайте, що значення `auto` початкове значення для властивостей розміщення гріди, тобто воно встановлюється по замовчуванню, якщо ви не вказали інше значення.

Ось приклад застосування гріди, позиціонування, зміни порядку і трансформації:



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>z-index</title>
</head>
<style>
body{
  margin: 20px;
  border: 5px solid #080808;
  background-color: rgb(67, 247, 232);
}
#grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
  height: 90vh;
  background-color: rgb(248, 245, 77);
  position: relative;

}
.red {
  background: orangered;
  z-index: 3;
  position: relative;
  left: 30vw;
  top: 30vh;
  width: 37.5%;
  transform: rotate(45deg)

}
.green {
  background: yellowgreen;
  position: relative;
  left: -20vw;
  top: 20vh;
  z-index: 1;
}
.blue {
  background: steelblue;
  position: relative;
  left: -40vw;
  top: 5vh;
  z-index: 2;
}
#grid > div {
  padding: 20px;
  font-size: 4vw;
  /* color: white; */
}

```

```

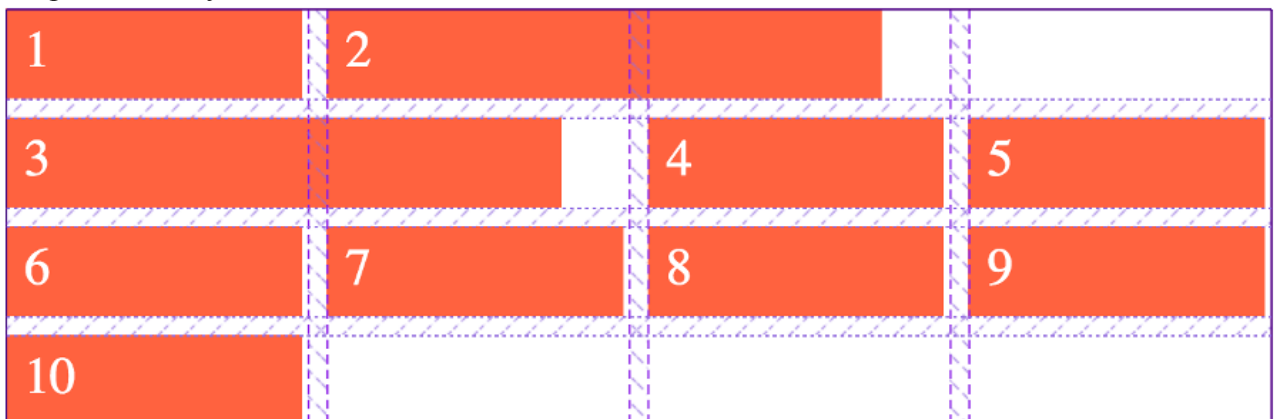
#abs {
  background: rgb(224, 11, 11);
  width: 50%;
  position: absolute;
  right: 20px;
  text-align: center;
  z-index: 5;
}
#rel {
  background: rgb(211, 20, 138);
  width: 50%;
  position: relative;
  right: 20px;
  bottom: 20px;
  z-index: 4;
}
</style>
<body>
<div id="grid">
<div class="red">11</div>
<div class="green">12</div>
<div class="blue">13</div>
<div>21</div>
<div>22</div>
<div id="abs">23</div>
<div>31</div>
<div>32</div>
<div id="rel">33</div>
</div>
</body>
</html>

```

Інспектор Грід

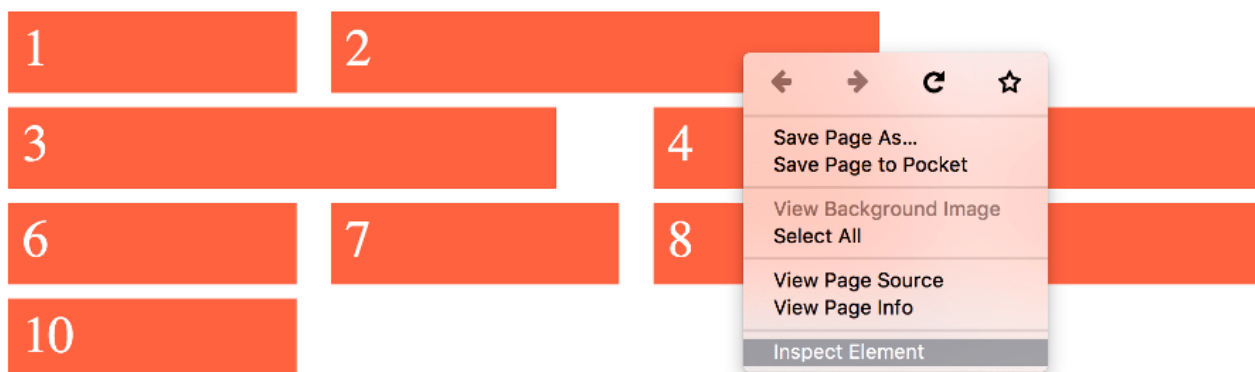
DevTools Google Chrome, Firefox та інших браузерів дає можливість перевірити грід шаблон будь-якого грїда, який ви бачите в браузері. Це може бути дуже корисно, якщо ви маєте справу зі складним грїдом, в якому грїд елементи розтягуються на безліч треків або з грїдом, який несильно заповнений. У таких випадках, може бути досить складним візуалізувати сам грїд, який містить в собі елементи.

Інспектор грїд в *DevTools* дозволить вам побачити лінії грїд і будь-які відступи між ними. Наприклад як тут:

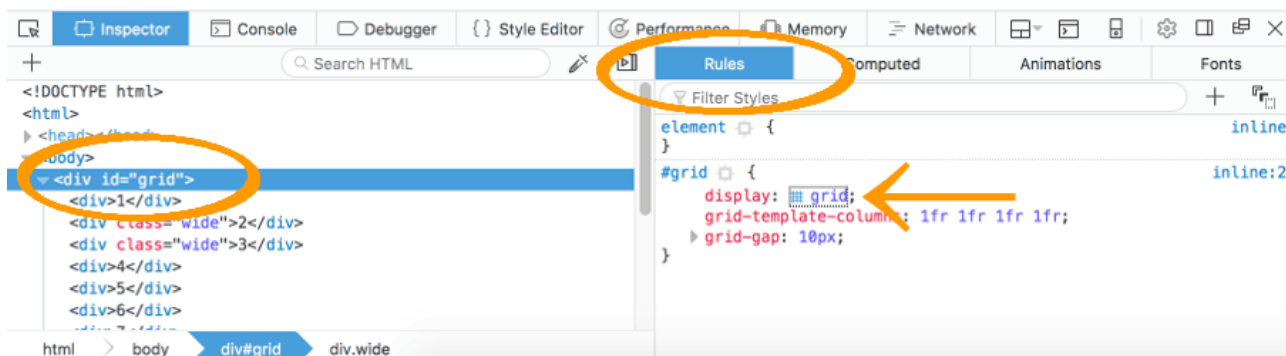
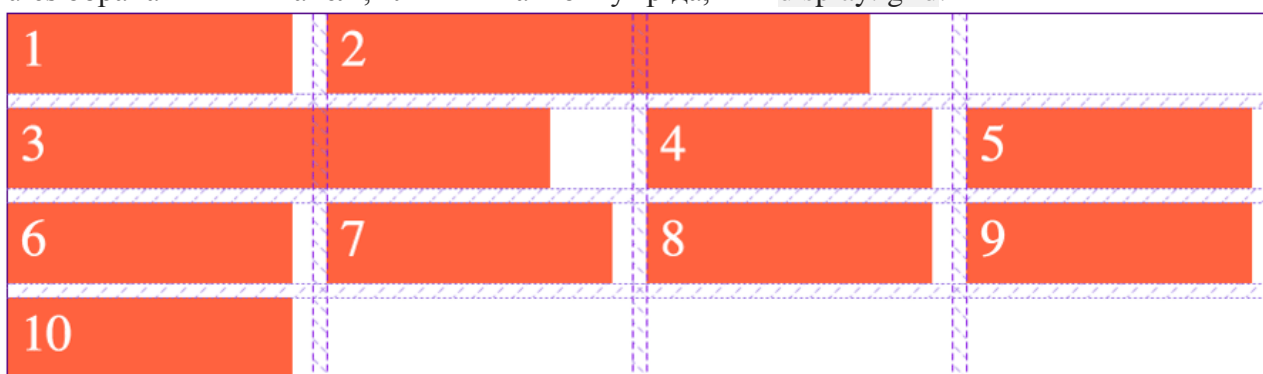


Як використовувати інспектор ґрид в браузері.

Відкрийте інспектор. Переглядаючи веб сторінку в браузері, клікніть правою кнопкою миші на ґріді, який хочете проінспектувати і виберіть Перевірити (Inspect Element) з контекстного меню або F12.



Переконавшись, що елемент з `display: grid` обраний в панелі з вихідним кодом і вкладка `rules` обрана в іншій панелі, клікніть на іконку ґрида, між `display: grid`.



ЛІТЕРАТУРА

1. Learn Photopea, URL: <https://www.photopea.com/learn/>
2. Figma for Beginners tutorial, URL: <https://help.figma.com/hc/en-us/sections/4405269443991-Figma-for-Beginners-4-parts->
3. Український веб-довідник, URL: <https://css.in.ua/>
4. Пікуляк М.В., Дутчак М.С. Курс лекцій з дисципліни «Web-дизайн» для студентів спеціальності 121 «Інженерія програмного забезпечення» ПНУ, 2024.
5. Дутчак М.С. Лабораторний практикум з дисципліни «Web-дизайн» для студентів спеціальності 121 «Інженерія програмного забезпечення» ПНУ, 2019.
6. Куленко М. Основи графічного дизайну: підручник.-3-тє вид., перероб. та доп., Рек. МОН К.: Кондор, 2015.
7. Брюханова Г. В. Комп'ютерні дизайн-технології: навч.посібник .-К.:ЦУЛ,2018 .- 180 с.
8. Тмєнова Н.П. Комп'ютерна графіка: навч.-метод.посібник. К.: ВПЦ "Київський ун-т", 2017.
9. Пасічник В. В., Пасічник О. В., Угрин Д. І. Веб-технології та веб-дизайн: підручник, Кн. 1: Веб-технології .-Рек. МОН .-Львів:Магнолія 2006,2013 .- 336 с.
10. Власій О. О., Дудка О. М. Комп'ютерна графіка. Обробка растрових зображень.
11. Івано-Франківськ: ПНУ, 2015
12. Пічугін М. Ф., Канкін І. О., Воротніков В. В. Комп'ютерна графіка: навч. посібник.- Рек. МОН, К.: ЦУЛ, 2013
13. Пасічник Володимир Володимирович, Пасічник О.В.
14. Веб-дизайн: підручник .- Рек. МОН .- Львів: "Магнолія 2006", 2012 .- 520 с.

Навчальний посібник

Пікуляк Микола Васильович
Дутчак Марія Степанівна

Навчальний посібник
з дисципліни «Web-дизайн»

для студентів спеціальності
121 «Інженерія програмного забезпечення

В авторській редакції

Підп. до друку 01.02.2024. Формат 60x84/16. Папір офсет.
Гарнітура «Times New Roman». Ум. друк. арк. 7,2.
Зам. № 1.8

Видавець і виготовлювач
Видавництво Прикарпатського національного університету
імені Василя Стефаника
76000, м. Івано-Франківськ, вул. С. Бандери, 1
Тел. 75-13-08. E-mail: vdvcit@pu.if.ua
Свідоцтво суб'єкта видавничої справи ДК № 2718 від 12.12.2006