

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА

М. С. ДУТЧАК

Методичні вказівки до виконання лабораторних робіт із
дисципліни “Front-end-розробка”

Івано-Франківськ
2024 р.

Д 84

Рекомендовано до друку Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника (протокол № 2 від 21 лютого 2024 р.)

Рецензенти:

Пікуляк М.В. – кандидат технічних наук, доцент кафедри інформаційних технологій факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника;

Савка І.Я. – кандидат фізико-математичних наук, ст. викладач кафедри інформаційних технологій факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника.

Дутчак М.С.

Д 84 Дутчак М.С. Методичні вказівки до виконання лабораторних робіт із дисципліни “Front-end-розробка”, Івано-Франківськ: ПНУ ім. В. Стефаника, 2024, 88 с.

У методичних вказівках наведено рекомендації щодо виконання лабораторних робіт з дисципліни «Front-end-розробка», що дозволяє здобувачам вищої освіти освоїти технології Front-end-розробки засобами мови розмітки гіпертексту HTML та каскадних таблиць стилів CSS.

Методичні вказівки рекомендовано для здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення» та інших спеціальностей ІТ-галузі.

УДК 004. 51

Д 84

© ПНУ ім. В.Стефаника, 2024

© Дутчак М.С., 2024

Передмова

Web-сервіси дедалі ширше знаходять впровадження практично в усіх сферах людської діяльності, тому освоєння інструментальних засобів та методології Web-розробки є актуальним етапом підготовки сучасного фахівця у сфері Web-технологій. Основою розробки Web-сторінок є мова розмітки гіпертексту HTML та каскадні таблиці стилів CSS, а також бібліотека Bootstrap. Саме цим інструментам Web-розробки і присвячені дані методичні вказівки до виконання лабораторних робіт із дисципліни “Front-end-розробка”, яка є важливою складовою частиною навчального процесу здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення», а також здобувачів вищої освіти інших спеціальностей IT-галузі.

Методичні вказівки містить теоретичний матеріал і завдання до виконання лабораторних робіт, які включають завдання для індивідуального виконання та спрямовані на закріплення теоретичного матеріалу. У методичних вказівках розглянуто основні можливості мова розмітки гіпертексту HTML та каскадні таблиці стилів CSS, а також бібліотеки Bootstrap, зокрема описано загальний синтаксис, основні конструкції, семантичну стилізацію елементів та різні способи версти вебсторінок, а також можливості CSS для роботу надання динаміки елементам та створення анімацій.

Наведений матеріал призначений для студентів, які вивчають Web-розробку на стороні клієнта, може бути використаний ними для самостійної роботи, а також для підготовки до виконання лабораторних і практичних робіт.

ЗМІСТ

Лабораторна робота №1. Створення елементарних Web-сторінок. Розміщення сайту на вебхостингу. Налаштування FTP-з'єднання.....	5
Лабораторна робота №2. Каскадні таблиці стилів CSS.....	13
Лабораторна робота №3. Карта зображення. Робота з таблицями. Динамічні ефекти.....	42
Лабораторна робота №4. Аркуші стилів. Тег div та span. Блочні та рядкові елементи. CSS-позиціонування.....	45
Лабораторна робота №5. HTML. Семантичні елементи.....	53
Лабораторна робота №6. CSS3. Стилзація елементів. Переходи, трансформації.....	55
Лабораторна робота №7. CSS. Блочна верстка веб-сторінок. Властивість display. CSS Grid.....	58
Лабораторна робота №8. CSS. Метод верстки Flexbox. Правила. Функції.....	61
Лабораторна робота №9. Верстка вебсторінок за допомогою фреймворка Bootstrap.....	65
Лабораторна робота №10. Вебформи. Псевдоелементи. Псевдокласи.....	68
Лабораторна робота №11. Верстка сайтів на основі PSD-макетів.....	71
Список літератури.....	73
Додаток А. Етапи реєстрації облікового на вебхостингу https://infinityfree.net/	74
Додаток Б. Етапи реєстрації на вебхостингу https://www.000webhost.com/	78
Додаток В. Налаштування FTP-з'єднання у програмах Total Commander, Файловий провідник та Far Manager.....	82
Додаток Г. Налаштувати VS Code для роботи із FTP-сервером хостингу..	84
Додаток Д. Налаштування ftp-з'єднання за допомогою програми Notepad++.....	86

Лабораторна робота №1. Створення елементарних Web-сторінок. Розміщення сайту на вебхостингу. Налаштування FTP-з'єднання.

Мета: навчитися створювати елементарні Web-сторінки засобами мови розмітки гіпертексту HTML і розміщувати їх на вебхостингу.

Тривалість 4 год.

1.1 Теоретичні відомості

1. Довідник по HTML тегам

1.2 Хід роботи

1.2.1 Створення елементарних Web-сторінок

На диску D локального ПК створіть папку дисципліни *front-end_lastname_group*, де *lastname* - *Ваше прізвище*, *group* - *назва групи*. Запустіть редактор вихідного коду VS Code, і в ньому відкрийте папку *front-end_lastname_group*: File -> Open folder - *front-end_lastname_group*. Дана папка повинна з'явитися у робочій області. У папці *front-end_lastname_group* створіть папку *lab1_lastname*.

1. Використовуючи редактор вихідного коду VS Code створіть HTML-файл під іменем *student.html* в папці *lab1_lastname* такого змісту:

```
<!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>Моя перша web-сторінка </title>
</head>
<body>
  ...
```

```
</body>
</html>
```

Дану структуру можна розгорнути ввівши знак оклику !, вибрати його із випадаючого списку, далі Enter або набравши із клавіатури.

Примітка. VS Code дозволяє пришвидшити роботу шляхом автоматичного написання структури контейнеру, наприклад ввівши літеру p і із списку вибрати p, далі Enter, далі VS Code розгорне `<p></p>`. А ввівши p.c11 структуру: `<p class="c11"></p>` і т.д.

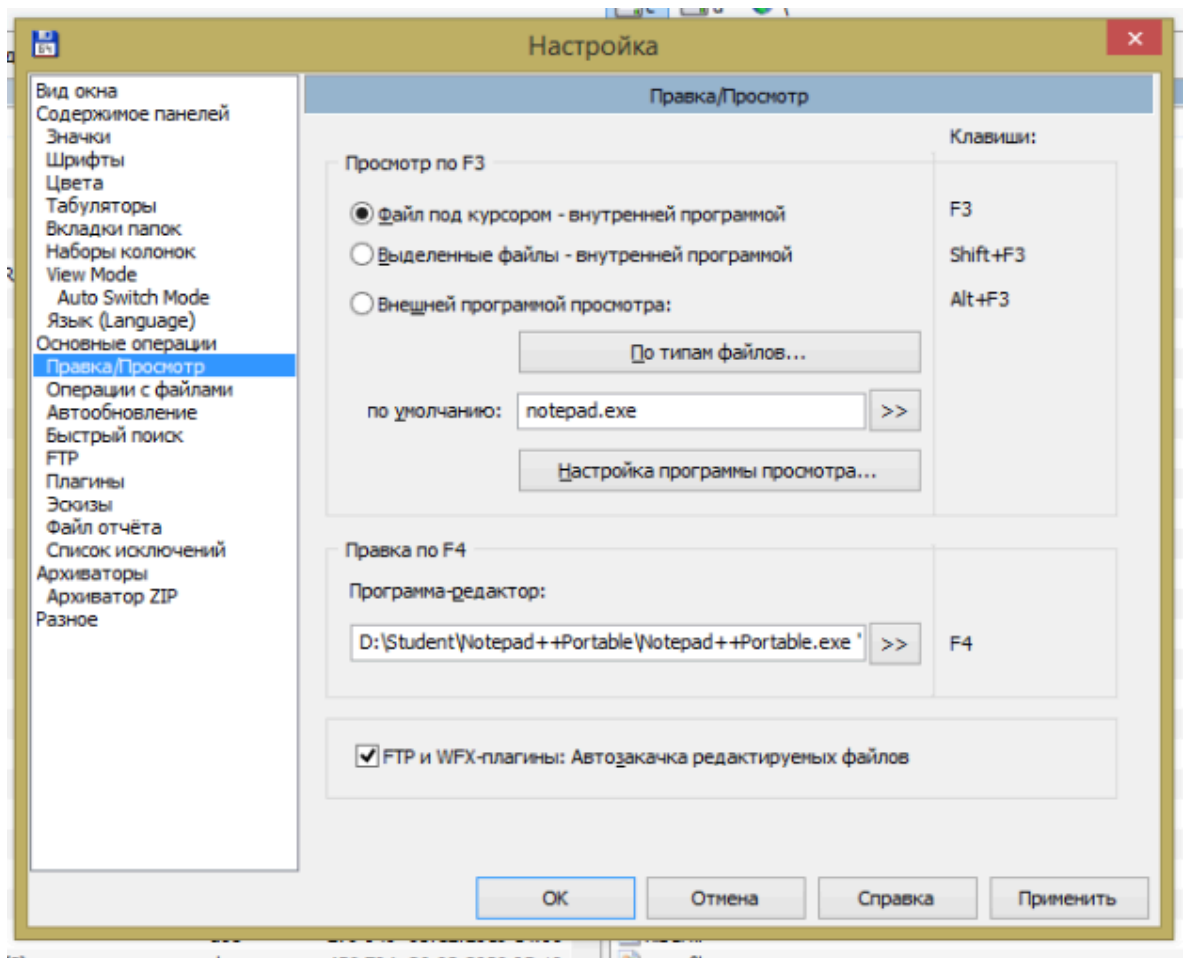
2. У файлі *student.html* вставте між тегами `<body>` і `</body>` наступний текст:

```
<h1> Сторінка користувача [Ваше прізвище] </h1>
<p>Я радий з вами познайомитись.</p>
<p> Це мої перші спроби створення власної сторінки. </p>
<p> Як студент я багато працюю з інформацією. </p>
<p>Мої захоплення: література, музика, комп'ютери, мандрівки рідним краєм. </p>
```

Збережіть ці зміни.

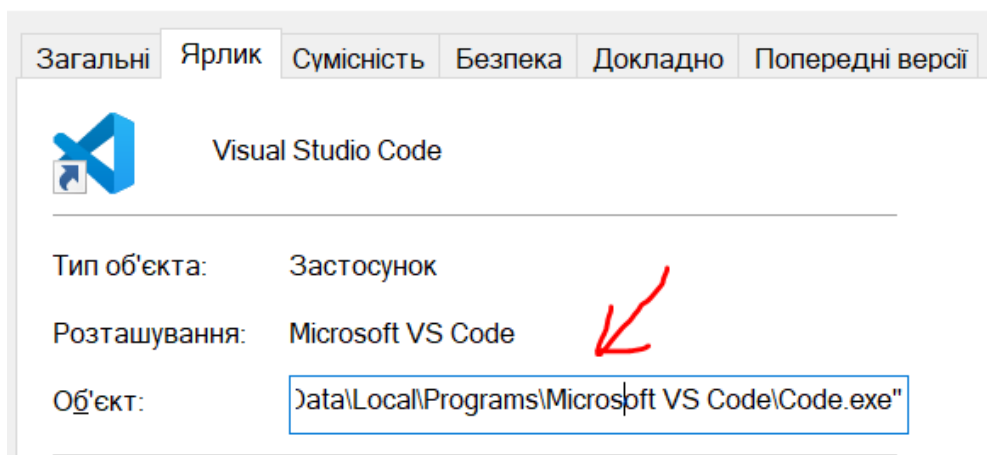
3. Не закриваючи вікна текстового редактора, перегляньте створений файл одним із доступних браузерів. Для цього у файловому менеджері Total Commander перейдіть у папку *lab1_lastname*, далі відкрийте файл *student.html* клікнувши Enter.

Зверніть увагу на розміри символів заголовку, створення абзаців. Поверніться до вікна текстового редактора VS Code, де відкритий код файлу *student.html*, закрийте його. Потім відкрийте використовуючи Total Commander в режимі редагування (F4). Для того, щоб в Total Commander змінити задану по замовчуванню програму редактора коду, заходите в Конфігурація - Налаштування - Правка/Перегляд - Програма-редактор, далі клікаєте на ">>" і шукаєте виконуючий файл необхідної програми.



Шлях до виконуючого файлу можна скопіювати із поля Об'єкт закладки Ярлик вікна властивостей ярлика програми, яке можна побачити, викликавши контекстне меню ярлика програми VS Code (правою кнопкою миші на ярлику програми)-> Властивості.

Visual Studio Code – властивості



У програмі Total Commander знаходячись курсором на файлі *student.html*. клікніть F4. Якщо він не відкрився у програмі VS Code, повторіть ще раз дане завдання.

У редакторі вихідного коду VS Code закрийте файл *student.html* і відкрийте знову папку дисципліни: File -> Open folder - *front-end_lastname_group*. Дана папка повинна з'явитися у робочій області.

4. Відкрийте файл *student.html* із папки *lab1_lastname* і у ньому поміняйте парні теги `<h1>`, `</h1>` на `<h3>`, `</h3>`. Перейдіть у вікно перегляду сторінки і оновіть її (Ctrl+R або F5). **Що змінилася?**

5. У папці *lab1_lastname* створіть папку *hyperlinks*.

6. У папці *hyperlinks* створіть HTML-документ *history.html*, додайте до нього зображення *.jpg, де * - назва jpg-файлу із папки відповідно до вашого варіанту: ``

Увага! Ваші файли у папці з номером, який відповідає Вашому номеру у списку групи. У цій і наступних роботах, якщо варіантів 10, а ваш номер у списку більше 10, то виберіть варіант, номер якого є остачею від ділення вашого номеру у списку групи на 10. Наприклад, якщо ви 18-ті у списку, то ваш варіант 8, якщо 27, то 7 і т. д.

7. Для зображення в HTML-документі задайте альтернативний текст – зображення недоступне.

8. Додайте у документ текст і одне зображення із файлу *1.doc* із папки вашого варіанту. Якщо зображення немає, то додайте будь-яке зображення. Тексту копіюйте стільки, щоб він займав мінімум півтори Web-сторінки, тобто щоб у браузері з'явилася вертикальна смуга прокрутки. Збережіть, перегляньте через браузер. Відформатуйте на свій вибір.

9. У папці *lab1_lastname* створіть файл *lab1.html*, в ньому додайте своє ПІБ, номер варіанту і посилання на файли *student.html* і *history.html*.

```
<a href="student.html">Файл student.html </a> <br>
```

```
<a href="hyperlinks/history.html">Файл history.html</a> <br>
```

10. На початку кожної із цих сторінок передбачте повернення на сторінку *lab1.html*.

Із файлу *student.html*:

`Стартова сторінка лабораторної роботи 1`

Із файлу *history.html*:

`Стартова сторінка лабораторної роботи 1`

`../` - перехід на рівень вище, `../..` - перехід на два рівні вище, і т.д.

11. На початку файлу *history.html* додайте заголовок "Закінчення".
Зробіть так, щоб при кліку:

- на зображення *.jpg відкривалась сторінка *student.html* із папки *lab1_lastname*.

``

- на заголовок "Закінчення" відкривався останній абзац HTML-документа *history.html*

Щоб створити швидку навігацію по Вашій Web-сторінці за допомогою гіперпосилань, потрібно у кінці документу для відповідного абзацу додати ідентифікатор:

`<p id="end">Це текст останнього абзацу web-сторінки </p>`

А на початку сторінки до посилання Закінчення додати гіперпосилання:

`<h3>Закінчення</h3>`

Самостійно додайте швидке повернення до початку сторінки із її закінчення з допомогою гіперпосилання і ідентифікатора.

1.2.2 Розміщення сайту на вебхостингу. Налаштування FTP-з'єднання.

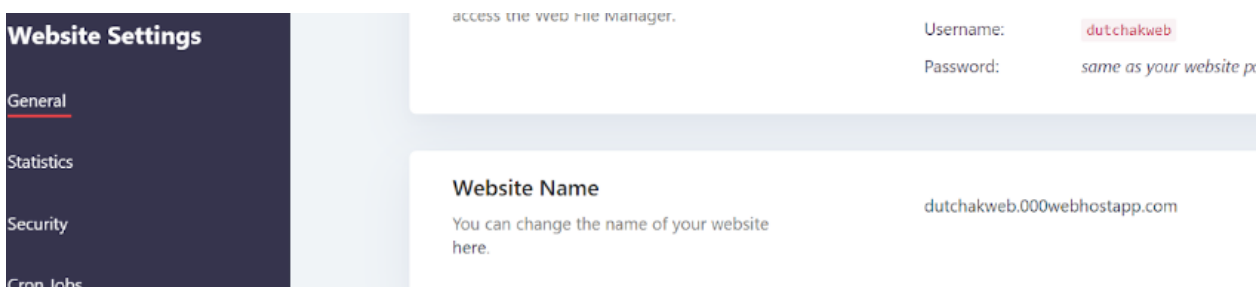
Інструкція для виконання перших трьох завдань знаходиться у додатках А-Д.

1. Зареєструйтеся на вебхостинг <https://infinityfree.net/>, <https://www.000webhost.com/>, <https://www.zzz.com.ua/> або на будь-якому іншому на свій вибір. Можливий випадок, що деякі вебхостинги не дозволяють реєструватися з університетської мережі, тоді можете спробувати це зробити з дому чи через мобільний Інтернет.

2. Використовуючи FTP-з'єднання будь-якого із файлових менеджерів (наприклад Total Commander, Far Manager) у робочій папці вебхостингу (`public_html`, `htdocs` тощо) створіть папку `front-end_lastname_group` і скопіюйте в неї папку `lab1_lastname` із виконаною першою лабораторною роботою.

3. Налаштуйте програму редактора вихідного коду VS Code для роботи з FTP-сервером вебхостингу.

4. Відкрийте у браузері файл `lab1.html` із папки `lab1_lastname`. Шлях до файлу повинен мати наступний вигляд: `WebsiteName/lab1_lastname/lab1.html`, де `WebsiteName` - адреса Вашої кореневої папки на вебхостингу:



5. Перевірте чи працюють всі посилання між файлами папки `lab1_lastname` згідно завдань даної лабораторної роботи. Якщо ні, то поправте чи, за потреби, додайте їх. Для написання шляхів використовуйте відносну адресацію, зокрема для переходу на рівень вище використовуйте: `../`

6. На вебхостингу у файл `lab1.html` папки `lab1_lastname` розмістіть скріншоти (фото) екрану із фінальними результатами виконання другої частини цієї лабораторної роботи, які б відображали встановлення ftp-з'єднання із вебхостингом засобами файлового менеджера (Total Commander, Far, Midnight commander) та редактора коду (VS Code, Sublime, Notepad++ (чи будь-яких інших аналогів цих програм, які Ви використовуєте), і вміст розміщених на вебхостингу папок `lab1_lastname` та *hyperlinks*.

7. У папці дисципліни на вебхостингу *front-end_lastname_group* поряд із папкою *lab1_lastname* створіть стартовий файл *index.html*, у якому вкажіть ваше ПІБ, номер варіанту (порядковий номер у списку групи), а також створіть посилання на стартовий файл даної лабораторної роботи орієнтовно із таким кодом:

```
<html>
<head>
<meta charset="UTF-8">
<title> Перелік лабораторних робіт студента <Ваше ПІБ> </title>
</head>
<body>
<h3> ПІБ <br> Варіант № ...</h3>
<h4>Дисципліна Front-end-розробка</h4>
<a href="lab1_lastname/lab1.html">Лабораторна робота №1 </a> <br>
</body>
</html>
```

8. У кожен html-сторінку із папки *lab1_lastname* у правому верхньому куті додайте посилання на стартовий файл *index.html* з текстом “На головну сторінку дисципліни”:

```
<a style="display: block; text-align: right" href="<hostname>">На головну
сторінку дисципліни</a>
```

Перевірте чи коректно працюють усі посилання. Всі наступні роботи теж розміщуйте на вебхостингу, папки із окремими лабораторними роботами називайте *labN_lastname*, де N- номер роботи, *lastname* - Ваше прізвище. На стартовій сторінці дисципліни *index.html* робіть посилання на стартові сторінки відповідних робіт під назвою *labN.html*, в них робіть посилання на виконані завдання відповідної роботи, і зворотні посилання на відповідну сторінку *labN.html*. Файл *labN.html* має знаходитися у відповідній папці *labN_lastname*.

9. Увага! Для здачі робіт, копії папок цієї та всіх наступних лабораторних робіт окрім розміщення на вебхостингу, завантажуйте також у папку дисципліни на Google Drive. При здачі лабораторних робіт у Classroom вказуйте посилання на відповідні папки із виконаними лабораторними роботами, розміщеними на Google Drive і знімки екрану, які демонструють розміщення файлів із виконаною роботою на вебхостингу. На знімках має бути видно назви файлів, їх розмір і дати.

10. Посилання на стартову сторінку вебхостингу і посилання на папку предмету на Google Drive вкажіть, заповнивши наступну форму:

<https://forms.gle/weYHxGrQFNcRLNff8>

Форму заповнюєте тільки один раз, за винятком зміни адреси вебхостингу чи посилання на папку предмету.

1.3 Контрольні запитання

1. Назвіть приклади парних та непарних тегів.
2. Для чого призначені атрибути?
3. Для чого і де використовують конструкцію “../”?
4. Яка клавіша та/або комбінація клавіш дозволяє оновити вебсторінку?

Лабораторна робота №2. Каскадні таблиці стилів CSS.

Мета: ознайомитися з поняттям «каскадні таблиці стилів» та навчитися працювати з стилями відображення гіпертексту CSS.

Тривалість: 2 год.

2.1. Теоретичні відомості

Мова HTML є мовою розмітки гіпертексту з дуже обмеженими можливостями. Вона була створена для того, щоб визначати структуру документа для розміщення його в Веб-середовищі, а не для роботи зі стилями.

Поступово в HTML були додані теги та технології, які дозволяли створювати та контролювати структуру та вигляд документа, з'явилися таблиці, фрейми, інструменти що вирівнюють текст, з'являється мова JavaScript.

У зв'язку з тим, що HTML не мав можливості управління зовнішнім виглядом Веб-сторінок, щоб вирішити ці проблеми, консорціумом W3C була розроблена технологія Cascading Style Sheets або CSS (Каскадні таблиці стилів) - технологія опису зовнішнього вигляду документа, написаного мовою розмітки. В CSS надається можливість призначити усім об'єктам стиль, опис якого може зберігатися в самому HTML-файлі, або в окремому файлі.

Один і той самий HTML або XML документ може бути відображений по-різному в залежності від використаного CSS. Стилі для відображення сторінки можуть бути:

- стилі автора (інформація надана автором сторінки):
 - зовнішні таблиці стилів (англ. *stylesheet*), частіше за все окремий файл або файли css;
 - внутрішні таблиці стилів, включені як частина документу або блоку
 - стилі для окремого елемента;

- стилі користувача:
 - локальний css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера;
- стилі браузера:
 - стандартний стиль браузера, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином використовується принцип *каскадності*, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена більш загальними стилями.

Переваги:

- інформація про стиль для цілого сайту або його частин може міститися в одному css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
- різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
- сторінки зменшуються в об'ємі та стають більш структурованими, за рахунок того що інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з їх урахуванням;
- прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

Правила в CSS

CSS надає можливість створювати правила, які легко змінювати, редагувати і застосовувати до усіх визначених нами елементів. Кожне правило, складається з двох частин. У лівій частині міститься *селектор*

(*selector*), а у правій - *блок визначення (declaration block)*. Блок визначення складається з набору *властивостей (property)* та їх *значень (value)*.

Селектор – елемент, який визначає правило.

Властивість описує елемент, що вводиться.

Значення визначають природу *властивостей*.

<i>Селектор</i>	<i>блок визначення</i>
селектор	{властивість: значення;}

Додавання таблиць стилів в html-документ

Існують три способи додавання правил CSS в HTML-документи.

Спосіб перший: додавання CSS в html-тег.

У цьому способі CSS додається в HTML-документ за допомогою HTML-атрибуту `style` у середині будь-якого HTML-тегу, що знаходиться у контейнері `<body>`.

```
<html>
<head>
<title>Приклад №1</title>
</head>
<body style="background-color: #FF0000;">
<p>Колір фону цієї сторінки - червоний</p>
</body>
</html>
```

Цей спосіб використовується у тому разі коли окремому елементу потрібно надати декілька стилів не використовуючи вбудовані або зовнішні стилі.

Застосування цього способу несе за собою певні недоліки:

- збільшується об'єм файлу, що приводить до збільшення часу завантаження веб-сторінки;
- ускладнює редагування документів.

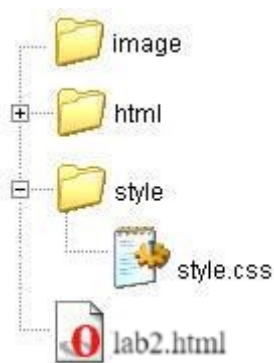
Спосіб другий: встановлення стилю для тегів в html-документі

CSS додається в HTML-документ за допомогою HTML-тегу `<style>` в середині контейнеру `<head>`. В ньому описуються всі стилі, що будуть використані.

```
<html>
  <head>
    <title>Приклад №2</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>Колір фону цієї сторінки - червоний</p>
  </body>
</html>
```

Спосіб третій: посилання на таблицю стилів

Зовнішня таблиця стилів являє собою звичайний текстовий файл з розширенням `css`.



Для того, щоб зробити посилання на зовнішній файл із html-документа, наприклад із файлу `example3.html` папки `html` на файл таблиці стилів (`style.css`) потрібно у контейнері `<head>` вставити наступну стрічку:

```
<link rel="stylesheet" type="text/css" href="../style/style.css">
```

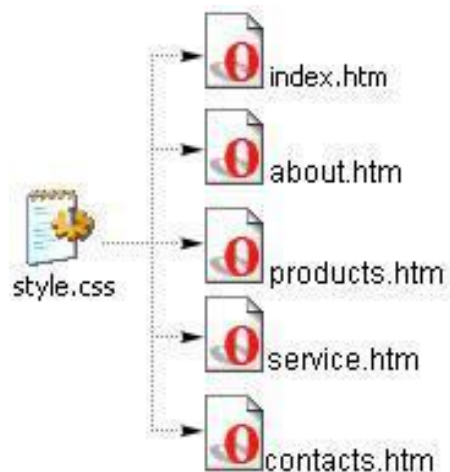
Це посилання вказує браузеру, що він повинен використовувати правила відображення HTML-файлу з CSS-файлу.


```
<html>
<head>
<title>Приклад №3</title>
<link rel="stylesheet" type="text/css" href="../style/style.css" />
</head>
<body>
<p>Колір фону цієї сторінки - голубий</p>
</body>
</html>
```

У файл `style.css` записуємо наступне:

```
body{
background-color: # 79e2f5;
}
```

Найважливішим тут є те, що один CSS-файл можна використовувати для управління відображення багатьох HTML-документів.



Ідентифікація і групування елементів

Атрибут `class` вказує, що елемент є членом певного класу. Модифікуємо список у файлі `lab2.html`, таким чином щоб одержати три списки посилань. Потрібно, щоб кольори посилань цих списків відрізнялися.

У файлі `lab2.html`:

```

<p>Список №1: </p>
<ul>
<li><a href="html/example1.html">Приклад 1</a></li>
<li><a href="html/example2.html">Приклад 2</a></li>
<li ><a href="html/example3.html">Приклад 3</a></li>
</ul>
<p>Список №2: </p>
<ul>
<li><a href="html/example4.html">Приклад 4</a></li>
<li><a href="html/example5.html">Приклад 5</a></li>
<li><a href="html/example6.html">Приклад 6</a></li>
</ul>
<p>Список №3: </p>
<ul>
<li><a href="html/example7.html">Приклад 7</a></li>
<li><a href="html/example8.html">Приклад 8</a></li>
<li><a href="html/example9.html">Приклад 9</a></li>
</ul>
<a href="html/example10.html">Приклад 10</a>

```

Щоб досягти нашої мети розділимо посилання на три категорії за допомогою присвоєння кожному посиланню перших двох списків класу атрибутом class, а третій список залишаємо без класу.

У файлі lab2.html додамо class="classname" в тег до якого ми будемо застосовувати клас:

```

<p>Список №1: </p>
<ul>
<li><a href="html/example1.html" class="list1">Приклад 1</a></li>
<li><a href="html/example2.html" class="list1">Приклад 2</a></li>
<li ><a href="html/example3.html" class="list1">Приклад 3</a></li>
</ul>
<p>Список №2: </p>

```

```

<ul>
<li><a href="html/example4.html" class="list2">Приклад 4</a></li>
<li><a href="html/example5.html" class="list2">Приклад 5</a></li>
<li><a href="html/example6.html" class="list2">Приклад 6</a></li>
</ul>
<p>Список №3: </p>
<ul>
<li><a href="html/example7.html">Приклад 7</a></li>
<li><a href="html/example8.html">Приклад 8</a></li>
<li><a href="html/example9.html">Приклад 9</a></li>
</ul>
<a href="html/example3.html" id="exam10">Приклад 10</a>

```

У файлі style.css опишемо властивості класів для посилань. Синтаксис опису класу має наступний вигляд:

селектор.	ім'я класу	{властивість: значення;}
-----------	------------	--------------------------

```

a {
color: blue;
}
a.list1 {
color: #FFBB00;
}
a.list2 {
color: #800000;
}

```

Окрім групування елементів вам може знадобитися ідентифікувати один унікальний елемент. Це можна реалізувати за допомогою атрибуту id.

Особливість id в тому, що в документі не може бути більш за один елемент з даним конкретним id. Кожен id має бути унікальним. У інших випадках потрібно використовувати атрибут class. У файлі lab2.html додамо id="id_name" для посилання на Приклад 10:

```
<a href="html/example3.html" id="exam10">Приклад 10</a>
```

Оформлення ідентифікатора у css-файлі буде мати наступний вигляд:

```
#exam10
{
color: #56c500;
}
```

КОЛІР ТА ФОН. Властивість `color` описує колір елемента. Наприклад, ми хочемо встановити зелений колір тексту абзаца, для цього у файлі `style.css` створимо таке правило:

```
p.ex4{
color: #00ff00;
}
```

А у `html`-файл:

```
<html>
<head>
<title>Приклад №4</title>
<link rel="stylesheet" type="text/css" href="../style/style.css"/>
</head>
<body>
<p class="ex4"> Колір тексту у цьому абзаці зелений</p>
</body>
</html>
```

Властивість `background-color` описує колір фону елемента. Для того, щоб змінити фон всієї веб-сторінки властивість `background-color` слід застосувати до тегу `<body>`. Також властивість `background-color` можна застосовувати до інших елементів веб-сторінки.

У файл style.css додаємо правило для Прикладу 5, що описують фон абзацу:

```
p.ex5 {  
  color: #00ff00;  
  background-color: #ff0000;  
}
```

```
<html>  
<head>  
<title>Приклад №5</title>  
<link rel="stylesheet" type="text/css" href="../style/style.css" />  
</head>  
<body class="body5">  
  <p class="ex5"> Колір тексту у цьому абзаці зелений на червоному  
фоні.</p>  
</body>  
</html>
```

За допомогою властивості background-image, зображення можна помістити на задній план елемента або зробити фоном веб-сторінки.

У цьому прикладі ми в якості зображення будемо використовувати



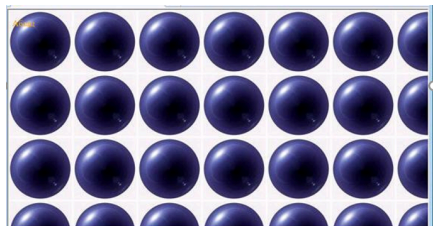
зображення шару:

Щоб зробити дане зображення фоном веб-сторінки достатньо в файлі style.css для Прикладу 5 застосувати параметри тега <body>:

```
body.body5 {  
  background-image: url("../images/ball.jpg");  
}
```

Властивість BACKGROUND-REPEAT.

Результатом виконання попереднього прикладу було повторення зображення шару, як по вертикалі так і по горизонталі:



Щоб керувати цим повторенням застосовується властивість background-repeat.

Таблиця значень background-repeat

Значення	Властивість
background-repeat: repeat-x	Зображення повторюється тільки по горизонталі.
background-repeat: repeat-y	Зображення повторюється тільки по вертикалі
background-repeat: repeat	Зображення повторюється по горизонталі і вертикалі.
background-repeat: no-repeat	Зображення не повторюється.

Модифікуємо Приклад 5:

```
body. body5 {  
  background-image: url("../images/ball.jpg");  
  background-repeat: repeat-x;  
}
```

Властивість BACKGROUND-ATTACHMENT надає можливість фіксування фонового зображення або прокручувати його разом з текстом.

Таблиця значень background-attachment

Значення	Властивість
background-attachment: scroll	Зображення не фіксується

background-attachment: fixed	Зображення зафіксовано.
------------------------------	-------------------------

Модифікуємо Приклад 5:

```
body. body5 {  
  background-image: url("../images/ball.jpg");  
  background-repeat: repeat-x;  
  background-attachment: fixed;  
}
```

За замовчуванням фонове зображення розміщується у верхньому лівому куті екрану. За допомогою властивості `background-position` зображення можна розміщувати у будь-якому місці екрану.

Значення властивості `background-position` являють собою вісь координат.

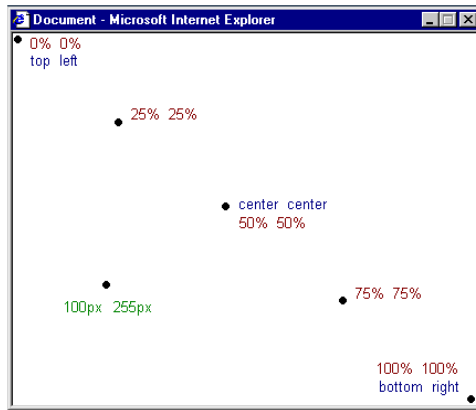
Модифікуємо Приклад 5:

```
body. body5 {  
  background-image: url("../images/ball.jpg");  
  background-repeat: no-repeat;  
  background-position: 5cm 5cm  
}
```

Після виконання цього прикладу ми побачимо, що зображення розташовано на 5 см зліва та на 5 см зверху.

В якості значень властивості `background-position` можна використовувати значення у процентах, ключові слова (`right`, `left`, `top`, `bottom`).

```
body.body5 {  
  background-image: url("../images/ball.jpg");  
  background-repeat: no-repeat;  
  background-position: 100% 25%  
}
```



Властивості `background`, що були розглянуті, можна записувати у скороченому вигляді.

Розглянемо фрагмент код:

```
background-color: #ff0000;  
background-image: url("../images/ball.jpg");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: 5cm 5cm
```

Для того, щоб властивість `background` можна було записати однією стрічкою, необхідно розташувати його властивості у такому порядку:

```
[background-color] | [background-image] | [background-repeat] |  
[background-attachment] | [background-position].  
background: #ff0000 url("../images/ball.jpg") no-repeat fixed 5cm 5cm;
```

Якщо властивість відсутня, то вона автоматично отримує значення за замовчуванням. Наприклад, якщо властивості `background-attachment` та `background-position` відсутні, то вони отримують свої значення за замовчуванням.

Управління шрифтом

В старих версіях HTML шрифт оформлявся за допомогою тегу ``, але цей тег потрібно було додавати кожного разу, коли було необхідно

встановити шрифт, від цього збільшувався розмір веб-сторінки, незручно було змінювати властивість шрифту.

Щоб задати сімейство шрифтів використовується властивість font-family. В цій властивості завжди вказується ряд шрифтів, розділених комою, наприкінці списку вказується сімейство шрифтів. При застосуванні шрифтів до веб-сторінки завжди задається основний шрифт, а потім альтернативний. У разі відсутності на комп'ютері користувача заданих шрифтів, то веб-сторінка, як мінімум буде відображена шрифтом, що входить до цього сімейства.

Існує п'ять сімейств шрифтів:

Шрифт	Сімейство шрифтів
Times New Roman Georgia	serif(з засічками)
Arial Candara	sans-serif (без засічок)
Courier Courier New	monospace (моноширинний)
Bradley Hand ITC Edwardian Script ITC	cursive (рукописний)
Ωεβδινγσ	fantasy (декоративний)

Шрифти сімейства serif найкраще підходять для основного тексту сторінки. Засічки допомагають направляти увагу читача уздовж рядка

Шрифти сімейства sans-serif використовуються для оформлення заголовків, панелей посилань та посилань. Шрифти без засічок звертають на себе більше уваги, але погано підходять для довгого читання.

У моноширинних шрифтах усі символи мають однакову ширину. Слід зазначити, що у моноширинних шрифтах збільшується між символний інтервал, як зліва так і з права символу (цей інтервал є невід'ємною частиною символу). Моноширинні шрифти допускаються тільки для створення якихось особливих ефектів оформлення — наприклад, у даній роботі моноширинним

шрифтом набрано фрагменти коду HTML та CSS. У моноширинному шрифті всякий символ має одну і ту ж ширину.

У файлі style.css створимо для оформлення абзацу таке правило:

```
p.ex6 {font-family: Arial, Candara, Century-Gothic, sans-serif}
```

```
<html>
<head>
<title>Приклад №6</title>
<link rel="stylesheet" type="text/css" href="../style/style.css" />
</head>
<body>
  <p class="ex6"> Шрифт цього абзацу відноситься до сімейства
  sans-serif</p>
</body>
</html>
```

У цьому прикладі основним шрифтом абзацу є шрифт Arial у разі його відсутності буде загрузено наступний, у разі відсутності усіх шрифтів абзац буде оформлено шрифтом сімейства sans-serif.

Властивість font-style визначає стиль шрифту з обраного сімейства може мати наступні значення:

- normal – звичайний шрифт;
- italic – курсивний шрифт (більш декоративний шрифт з нахилом вправо);
- oblique – нахилений шрифт (звичайний шрифт нахилений вправо).

У файлі style.css створимо таке правило:

```
h1.ex7 {
  font-family: Arial, Candara, Century-Gothic, sans-serif;
  font-style: normal;
}
h2.ex7 {
```

```

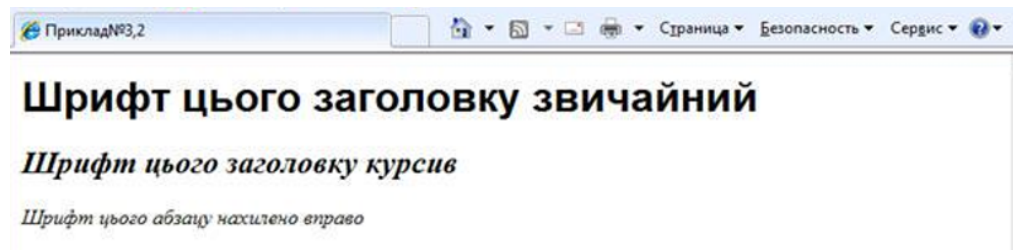
font-family: Times New Roman, Georgia, serif;
font-style: italic;
}
p.ex7{
font-family: Times New Roman, Georgia, serif;
font-style: oblique;
}

```

```

<html>
<head>
<title>Приклад №7</title>
<link rel="stylesheet" type="text/css" href="../style/style.css" />
</head>
<body>
<h1 class="ex7">Шрифт цього заголовку звичайний</h1>
<h2 class="ex7">Шрифт цього заголовку курсив</h2>
<p class="ex7"> Шрифт цього абзацу нахилено вправо</p>
</body>
</html>

```



Властивість `font-variant` використовується для вибору між варіантами `normal` і `small-caps`.

Параметр `normal` задає звичайні літери.

Параметр `small-caps` задає малі заголовні букви (`upper case`) замість букв нижнього регістру.

У файлі `style.css` створимо таке правило:

```

p.ex8 {
  font-family: Arial, Candara, Century-Gothic, sans-serif;
  font-style: normal;
  font-variant: small-caps;
}
<html>
<head>
<title>Приклад №8</title>
<link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
<p class="ex8">Абзац оформлено малими прописними літерами</p>
</body>
</html>

```

Властивість `font-weight` описує, наскільки товстим, або "важким", повинен відображатись шрифт. Шрифт може бути `normal` або `bold`. Деякі браузері підтримують навіть числові значення 100-900 (у сотнях) для опису ваги шрифту.

Іноді при значенні "500" або "600" браузері відображають "проміжну" насиченість між `normal` і `bold`.

У файлі `style.css` створимо таке правило:

```

p.ex9 {
  font-family: Arial, Candara, Century-Gothic, sans-serif;
  font-style: normal;
  font-variant: normal;
  font-weight: 600
}

<html>
<head>
<title>Приклад №9</title>
<link rel="stylesheet" type="text/css" href="../style/style.css" />

```

```
</head>
<body>
<p class="ex9">Товщина шрифту абзацу дорівнює 600</p>
</body>
</html>
```

Властивість `font-size` задає розмір шрифту для відображення вмісту елемента. Щоб задати розмір шрифту використовуються різні одиниці виміру (наприклад, пікселі або відсотки).

```
h1.ex10{
  font-family: Arial, Candara, Century-Gothic, sans-serif;
  font-style: normal;
  font-size: 30px;
}
h2.ex10{
  font-family: Times New Roman, Georgia, serif;
  font-style: italic;
  font-size: 100pt;
}
h3.ex10{
  font-family: Times New Roman, Georgia, serif;
  font-style: italic;
  font-size: 120%;
}
p.ex10{
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
  font-family: arial, sans-serif;
}
<html>
<head>
<title>Приклад №10</title>
```

```

<link rel="stylesheet" type="text/css" href="../style/style.css" />
</head>
<body>
<h1 class="ex10">Шрифт нормальний, розмір 30px</h1>
<h2 class="ex10"> Шрифт курсив, розмір 100pt </h2>
<h3 class="ex10"> Шрифт курсив, розмір 120% </h3>
<p class="ex10"> Шрифт напівжирний, курсив, 30px</p>
</body>
</html>

```

Властивості font, що були розглянуті, можна записувати у скороченому вигляді. Розглянемо фрагмент коду оформлення абзацу:

```

p{
font-style: italic;
font-weight: bold;
font-size: 30px;
font-family: arial, sans-serif;
}

```

Використовуючи скорочений запис, код можна спростити:

```

p{
font: italic bold 30px arial, sans-serif;
}

```

Порядок розташування властивостей такий:

font-style | font-variant | font-weight | font-size | font-family.

Управління текстом

Властивість text-indent задає відступ першого рядка при відображенні блокових елементів. Відступ може бути, як позитивним так і негативним і задається одним з наступних способів:

Значення	Дія
<розмір>	визначає, що відступ першого рядка фіксований

<Відсоток>	визначає, що відступ обчислюється щодо ширини блоку, що вміщується.
------------	---

Додаємо для Прикладу 10 властивість `text-indent` (задані раніше властивості залишаємо без змін):

```
p.ex10{
text-indent: 10%;
}
```

Модифікуємо для Приклад 10 властивість `text-indent`:

```
p.ex10{
text-indent: 3pt;
}
```

Властивість `text-align` задає вирівнювання тексту при відображенні блокових елементів. Воно може набувати наступних значень:

Значення	Дія
left	Вирівнювання по лівому краю.
right	Вирівнювання по правому краю.
center	Вирівнювання по центру.
justify	Вирівнювання за шириною.

Додаємо для Прикладу 10 властивість `text-align` (задані раніше властивості залишаємо без змін):

```
p.ex10{
text-align: justify;
}
```

Властивість `text-decoration` прикрашає текст при відображенні елементів. Воно може набувати наступних значень:

Значення	Відображення
none	Звичайний текст.
underline	Підкреслений текст.
overline	Надкреслений текст.
line-through	Перекреслений текст.
blink	Блимаючий текст.

Додаємо для Прикладу 10 властивість `text-decoration` (задані раніше властивості залишаємо без змін):

```
p.ex10{
text-indent: 3pt;
text-align: justify;
text-decoration: line-through;
}
```

Властивість `letter-spacing` задає інтервал між буквами при відображенні тексту. Його значення задається одним з наступних способів:

Значення	Відображення
normal	Звичайний текст.
<розмір>	Задає інтервал в додаток до стандартного інтервалу. Це значення може бути негативним.

Додаємо для Прикладу 10 властивість `letter-spacing` (задані раніше властивості залишаємо без змін):

```
p.ex10{letter-spacing: 2em;}
```


Властивість text-transform задає перетворення тексту при відображенні елементів. Воно може набувати наступних значень:

Значення	Дія	Приклад
none	Без перетворення.	Це приклад.
capitalize	Робити першу літеру кожного слова прописною.	Це Приклад.
uppercase	Виводити текст прописними літерами.	ЦЕ ПРИКЛАД.
lowercase	Виводити текст рядковими літерами.	це приклад.

До Прикладу 10 додати властивість text-transform з чотирма варіантами значень, результатом виконання яких є наведені приклади.

Різновиди селекторів

1. *Групування селекторів.* При розробці стилю сайту виникає ситуація, коли використовуються параметри, що повторюються. Для того, щоб ці параметри не повторювалися використовується групування селекторів за певними властивостями.

без групування	з групуванням
<pre> H1 { font-family: Arial, Helvetica, sans-serif; font-size: 160%; color: #000aff; } H2 { font-family: Arial, Helvetica, sans-serif; font-size: 135%; </pre>	<pre> H1, H2 { font-family: Arial, Helvetica, sans-serif; } H1 { font-size: 160%; color: #000aff; } H2 { font-size: 135%; </pre>

<pre>color: #aaaaaa; }</pre>	<pre>color: #aaaaaa; }</pre>
<pre>h4{ font-weight: bold; text-align: center; } #a h4{ font-style: italic; } #b h4{ font-style: italic; }</pre>	<pre>h4{ font-weight: bold; text-align: center; } #a h4, #b h4{ font-style: italic; }</pre>

2. *

```
* {margin: 0; padding: 0; }
```

Цей CSS-селектор виділяє кожен елемент на сторінці. Багато розробники використовують його для того, щоб скинути у всіх елементів значення `margin` і `padding`. На перший погляд це зручно, але все-таки в робочому коді так краще не робити. Цей CSS-селектор занадто сильно вантажить браузер.

* Також можна використовувати для виділення дочірніх елементів.

```
#container * {border: 1px solid black; }
```

В даному випадку виділяються всі дочірні елементи `#container`. Знову ж, намагайтеся не зловживати ним.

сумісність

3. XY

```
li a {text-decoration: none; }
```

CSS-селектор дочірніх елементів зустрічається найчастіше. Якщо вам потрібно виділити елементи певного типу з множини елементів, використовуєте цей селектор. Наприклад, вам потрібно виділити всі

посилання, які знаходяться в елементі li. У цьому випадку використовуйте цей селектор.

4. $X + Y$

```
ul + p {color: red; }
```

Виділяє наступний елемент. Він буде вибирати *тільки* перший елемент типу Y, який йде відразу після елемента X. У прикладі текст першого абзацу після кожного ul буде червоного кольору.

5. $X > Y$

```
div # container > ul {border: 1px solid black; }
```

Різниця між стандартними $X Y$ і $X > Y$ полягає в тому, що даний CSS-селектор буде вибирати тільки безпосередні дочірні елементи. Наприклад, розглянемо наступний код.

```
<div id = "container">
  <ul>
    <li> Приклад
    <ul>
      <li> Дочірній елемент </ li>
    </ ul>
  </ li>

  <li> Приклад </ li>
  <li> Приклад </ li>
  <li> Приклад </ li>
</ ul>
</ div>
```

CSS-селектор `#container > ul` вибере тільки ul-елементи, які є безпосередніми дочірніми елементами div з `id = container`. Він не вибере, наприклад, ul-и, які є дочірніми елементами перших li.

Тому можна отримати вигоду в продуктивності використовуючи даний CSS-селектор. Справді, це особливо рекомендується при роботі з jQuery або іншими бібліотеками, які обирають елементи на основі правил CSS-селекторів.

6. $X \sim Y$

```
ul ~ p {color: red; }
```

Цей CSS-селектор дуже схожий на $X + Y$, однак, є менш суворим. При використанні $ul + p$ буде вибрати тільки перший елемент, що йде за X . В даному випадку будуть вибрані всі елементи p , що йдуть за ul .

7. $X [title]$

```
a [title] {color: green; }
```

В CSS-селекторах також можна використовувати атрибути. Наприклад в даному прикладі ми виділили всі посилання, які мають атрибут `title`. Решта посилання залишаться не порушеними.

8. $X [href = "foo"]$

```
a [href = "http: // everstudent .ua"] {color: # ffde00; }
```

Всі посилання, які посилаються на `everstudent.ua` будуть золотими. Всі інші посилання залишаться без змін.

Зверніть увагу на лапки. Не забудьте так само робити в jQuery і інших JavaScript бібліотеках, в яких елементи вибираються по CSS-селекторів.

9. $X [HREF * = "everstudent.ua"]$

```
a [href * = "everstudent"] {color: # 1f6053; }
```

Зірка позначає, що шукане значення має з'явитися *будь-де* в атрибуті. Таким чином, CSS-селектор охоплює `everstudent.ua`, `http://everstudent.ua/portfolio` і т.д.

Але що робити, якщо посилання веде на якийсь сторонній і не пов'язаний ресурс, в адресі якого присутній `everstudent`? Тоді потрібно використовувати `"^"` або `"&"`, для посилання на початок і кінець рядка відповідно.

10. $X [href ^ = "http"]$

```
a [href ^ = "http"] {  
background: url (path / to / external / icon.png) no-repeat;  
padding-left: 10px;  
}
```

Ви ніколи не замислювалися, як деякі веб-сайти можуть відображати маленький значок поряд із зовнішніми посиланнями? Я впевнений, що ви бачили їх раніше, вони добре запам'ятовуються.

"^" - найбільш часто використовується у регулярних виразах символ. Він використовується для позначення початку рядка. Якщо ми хочемо охопити всі теги, у яких href починається з http, нам потрібно використовувати CSS-селектор, наведений вище.

Зверніть увагу, що ми не шукаємо "http: //". Це не правильно, оскільки не враховуються адреси, що починаються з https: //

А якщо ми хочемо задати стиль тільки для посилань, що ведуть на фотографію? Потрібно шукати *кінець* рядка.

11. *X [href \$ = ".jpg"]*

```
a [href $ = ".jpg"] {color: red; }
```

Знову ж таки, ми використовуємо символ регулярного виразу "\$" для позначення кінця рядка. В даному прикладі ми шукаємо посилання, які посилаються на jpg-файли, або uрl-и, в кінці у яких міститься ".jpg".

12. *X: checked*

```
input [type = radio]: checked {border: 1px solid black; }
```

Цей псевдоклас виділяє тільки елементи перемикач або прапорець. Причому ті, які відзначені / обрані.

13. *X: after*

Псевдокласи: before та after генерують контент навколо обраного елемента.

```
.clearfix: after {  
  content: "";  
  display: block;  
  clear: both;  
  visibility: hidden;  
  font-size: 0;  
  height: 0;  
}  
.clearfix {  
  * display: inline-block;
```

```
_height: 1%;  
}
```

Цей приклад використовує `after`, щоб додати пробіл після елемента і заборонити його обтікання. *X: `hover`*

```
div: hover {background: #e3e3e3; }
```

Цей CSS-селектор застосовує стиль до елемента, коли на нього наводиться курсор миші.

Цей CSS-селектор часто використовують для того, щоб поставити `border-bottom` на посилання, коли на них наводять мишкою.

```
a: hover {border-bottom: 1px solid black; }
```

`border-bottom: 1px solid black;` виглядає краще, ніж `text-decoration: underline;`

Посилання

Псевдокласи. Властивості гіперпосилання засобами CSS можна визначати по-різному, залежно від того, відвідали вже посилання, чи активне воно, чи знаходиться покажчик миші над посиланням. Це дозволяє додати цікаві ефекти на ваш веб-сайт. Щоб оформити гіперпосилання засобами CSS, потрібно використовувати так звані псевдокласи.

Псевдоклас дозволяє враховувати різні стани або події при визначенні властивостей `html`-тега. У гіперпосилання є декілька станів та подій.

Псевдоклас `: link` використовується для посилань на сторінки, які користувач ще не відвідував.

```
a: link {  
color: #0000ff;  
font-weight: normal;  
}
```

Псевдоклас: `active` використовується для активних посилань.

```
a: active {  
color: #00bfff;
```

```
background-color: ffd700;
}
```

У цьому прикладі у посилання буде змінено колір шрифту та колір фону.

Псевдоклас: `visited` використовується для посилань на сторінки, які відвідав користувач.

```
a: visited {
color: #0000ff;
font-weight: 400;
}
```

Псевдоклас: `hover` використовується для посилань, над котрими знаходиться вказівник миші.

```
a: hover {
font-weight: normal;
text-trasform: uppercase;
letter-spacing: 10px;
}
```

У цьому прикладі шрифт посилання буде нормальної товщини, текст буде відображатися прописними літерами, відстань між літерами буде складати 10px.

Видалення підкреслювання посилань. Видалити підкреслювання посилань дуже просто. Для видалення підкреслювання достатньо встановити властивість `text-decoration` зі значенням `none`.

```
a: link {
color: #0000ff;
font-weight: normal;
text-decoration: none;
}
```

Одиниці вимірювання

px - прив'язані до абсолютних пікселі і тому не потрібні mm, cm, pt і pc. Використовується для максимально конкретного і точного завдання розмірів.

- 1mm (мм) = 3.8px
- 1cm (см) = 38px
- 1pt (типографний пункт) = 4/3 px
- 1pc (типографська піка) = 16px

Не відповідають реальним см., мм. чи пунктам. 1 см=2.54 дюйми, 1 дюйм=72 pt.

em - задає розмір щодо шрифту батька, можна відносно конкретних символів: "x" (ex) і "0" (ch), використовується там, де потрібно спростити масштабування компоненти.

rem - задає розмір щодо шрифту <html>, використовується для зручності глобального масштабування: елементи які планується масштабувати, задаються в rem, а JS змінює шрифт у <html>.

% - Відносно такої ж властивості батька (як правило, але не завжди), використовується для ширин, висот і так далі, без нього нікуди, але потрібно знати, щодо чого він вираховується відсотки.

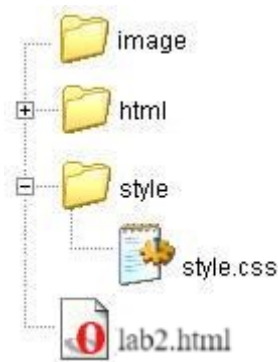
vw, vh, vmin, vmax - щодо розміру екрана.

Додаткові теоретичні відомості

1. <https://css.in.ua/html/tags>
2. <https://css.in.ua/css/selectors>
3. <http://css.in.ua/css/properties>
4. <https://html-css.co.ua/>

2.3 Хід роботи

1. У робочій папці дисципліни на вебхостингу front_end_lastname створіть папку lab2_lastname. У папці lab2_lastname створіть наступну структуру об'єктів:



Файл `index.html` із папки `front_end_lastname` повинен містити посилання на стартові сторінки всіх виконаних Вами робіт. Зокрема для цієї роботи у ньому створіть посилання на файл `lab2.html` із папки `lab2_lastname` і в зворотньому напрямку.

2. Ознайомтеся з теоретичними відомостями до виконання роботи і реалізуйте 10 наведених у них прикладів.

Кожен приклад записуйте різними файлами у папку `html` під відповідною назвою (Наприклад: Приклад 1 у `example1.html`, і т.д.). У файлі `lab2.html` наведені посилання на відповідні приклади необхідно модифікувати відповідно до розділу «Ідентифікація і групування елементів» із теоретичних відомостей. Зовнішні стилі записуйте у файл `style.css`, який повинен бути приєднаним до кожного із `html`-файлів.

Приклади реалізуйте по порядку, присвоюючи при потребі клас чи ідентифікатор, таким чином щоб усі стилі містилися у файлі `style.css`, а реалізовані приклади відповідали наведеним у теоретичних відомостей.

3. Реалізуйте будь-які вісім різновидів селекторів із пункту «Різновиди селекторів»

4. Здайте роботу.

2.3. Контрольні питання

1. Для чого служать аркуші стилів?
2. У чому полягає різниця між зовнішніми та внутрішніми аркушами стилів?
3. Назвіть види селекторів і призначення кожного з них.
4. Які правила записів внутрішніх стилів?
5. Яка область застосування аркушів стилів?

Лабораторна робота №3. Карта зображення. Робота з таблицями. Динамічні ефекти.

Мета: навчитися використовувати карту зображень, створювати таблиці та динамічні об'єкти.

Тривалість: 2 год.

3.1 Теоретичні відомості

1. <https://css.in.ua/html/tags>
2. <http://css.in.ua/css/properties>
3. <https://html-css.co.ua/html/tag-table/>
4. Стилізація таблиць
5. Iframe
6. <http://css.in.ua/html/tag/audio>
7. <http://css.in.ua/html/tag/video>
8. <http://css.in.ua/html/tag/map>

3.2 Хід роботи

Варіант – порядковий номер у списку групи. Завдання виконуйте у різних файлах.

Завдання 1.

Знайдіть або створіть картинку, на якій зображено мінімум три різні об'єкти згідно вашого варіанту, які з допомогою тегу <http://css.in.ua/html/tag/map> зробіть посиланнями на їх опис у <https://uk.wikipedia.org/>. Форма областей: круг, трикутник, прямокутник. Координати точок можна шукати з допомогою <https://www.photopea.com/> (додаток Google, безкоштовний аналог Adobe Photoshop): Вікно/інформація.

Початок роботи із додатком Photopea: перейдіть на сайт <https://www.photopea.com/>. Ви можете працювати у онлайн-версії додатку або встановити десктопну версію додатка, клікнувши на Більше-Встановити Photopea у веб-версії додатку photopea.com.

Детальніше [тут](#).

Теми відповідно до варіантів:

- | | | |
|-----------------------|-------------------------|-------------------------|
| 1. Птахи | 13. Знаки зодіаку | 22. Музичні інструменти |
| 2. Риби | 14. Корисні копалини | 23. Автомобілі |
| 3. Звірі | 15. Комп'ютерна техніка | 24. Жіночий одяг |
| 4. Комахи | 16. Жіночий одяг | 25. Чоловічий одяг |
| 5. Гриби | 17. Чоловічий одяг | 26. Знаки зодіаку |
| 6. Автомобілі | 18. Дитячий одяг | 27. Корисні копалини |
| 7. Деревя | 19. Взуття | 28. Автомобілі |
| 8. Фрукти | 20. Ювелірні аксесуари | 29. Побутова техніка |
| 9. Овочі | 21. Спортивні аксесуари | 30. Комп'ютерна техніка |
| 10. Ягоди | | |
| 11. Лікарські рослини | | |
| 12. Планети | | |

Завдання 2.

Зобразіть з допомогою стилів таблиці із файлу [table.jpg](#)

Див. [Стилізація таблиць](#)

Завдання 3.

Створіть [таблицю](#) із зображеннями згідно наступних вимог:

Мої одногрупники			
1.jpg	2.jpg	3.jpg	4.jpg
[Коротка біографія студентів]			

Ширина таблиці відповідає ширині веб-сторінки. Задайте стилі для границь і фону комірок.

Зліва від заголовка першого рядка розмістити анімацію, яка знаходиться у папці [Аніме](#) або будь-яку іншу.

У другому рядку малюнки *.jpg – це фотографії однакового розміру (4x3 см., 150 dpi) чотирьох однокрупників (2 зверху, 2 знизу по списку журналу) із зазначеними під зображеннями прізвищами.

Фото знаходяться в папці photo.

При кліку на кожну із фотографій однокрупників у третьому рядку повинна відображатись відповідна біографія, з наступною інформацією: П.І.Б., дата народження, посилання на електронну скриньку, улюблену мелодію. Примітка: для ефекту зміни біографій створіть 4 однакові сторінки, але з різним вмістом третього рядка. Цю задачу можна зробити з допомогою тегу <iframe> (див. приклад), але його використання небажане (дізнайтесь чому).

Під таблицею задайте наступні елементи:

- аудіо - програвач (тег audio),
- відео - програвач (тег video),
- посилання на скачування відповідного аудіо і відео файлу.

Онлайн-конвертером: <https://video.online-convert.com/>.

Контрольні питання:

1. Які ви знаєте форми областей для створення карти зображень?
2. Назвіть теги для створення елементів та їх елементів.
3. Чому не рекомендоване використання тегу <iframe>?

Лабораторна робота №4. Аркуші стилів. Тег div та span. Блочні та рядкові елементи. CSS-позиціонування

Мета: опанувати принципи блочних та рядкових елементів.

Тривалість: 2 год.

4.1 Теоретичні відомості

1. Блочні елементи, властивість display
2. Блокові HTML теги
3. Рядкові HTML теги
4. CSS-позиціонування

4.2 Хід роботи

Завдання 1.

Наберіть HTML-документ за допомогою будь-якого із HTML-редакторів, виділіть селектори типу, селектори класу, селектори id. З'ясуйте в чому різниця між використанням селекторів. Визначте область дії селекторів, де й чому не спрацювали аркуші стилів.

```
<html>
<head>
  <title>Untitled Document</title>
  <meta charset="UTF-8">
  <style type="text/css">
    h2 {
      font-size: 16pt;
      color: #0033FF;
      font-family: Geneva, Arial
    }
    h2.curs {
      font-size: 18pt;
      font-style: italic;
```

```
font-family: "Courier New", Courier, mono;
color: #46dcf9
}
h2.normal {
font-size: 16pt;
font-style: normal;
color: #FF0000
}
#bgmagenta{
text-indent: 20px;
background-color: #1166FF;
color: #FFFF00
}
#fishka{
background-color: #CC99FF;
color: #3300FF;
font-family: serif;
font-size: 20pt;
font-style: oblique
}
#bgyellow{
background-color: #FFFF00;
color: #0000FF;
font-family: Verdana, Arial
}
div {
position: absolute;
width: 150px;
height: 150px;
padding: 20px;
border: dashed 3px;
font-size: 15pt;
}
div.text {
```

```

position: absolute;
font-family: "serif";
background-color: #99FFFF;
color: red;
font-style: oblique;
top: 600px;
width: 850px;
height: 150px;
padding: 20px;
}
#first {
top: 850px;
left: 50px;
background: #ccf0ff;
border-color: #0cf;
}
#second {
top: 950px;
left: 150px;
background: #fff0cc;
border-color: #fc0;
font-size: 10pt;
}
</style>
</head>
<body bgcolor="#FFCC99">
  <h2>Приємно на пари з ранку приходити</h2>
  <h2 class="curs"> Приємно бачити добрі очі екзаменатора</h2>
  <h2 id="bgmagenta"> Приємно зайти в деканат після сесії</h2>
  <p id="bgmagenta"> Приємно не мати "хвостів", <br> бути успішним
  і <br id="fishka"> упевненим у завтрашньому дні</p>
  <h2 class="curs"> Приємно вчасно захищати лабораторні
роботи...</h2><br>

```

```
<div class="text" align="center "> Життя світле й безхмарне, коли тобі
20 років і попереду <p class="normal"> море надій, </p> <hr id="bgyellow"
size="8"> усе тіло наповнене енергією й готове до <p id="bgmagenta">
здійснень, щастя, любові </p></div>
```

```
<h3 class="curs"> Життя повне приємних моментів</h1>
```

```
<h3 id="fishka" align="center"> Тільки варто небагато
попрацювати</h3>
```

```
<hr id="bgmagenta" align="center" width="200" size="5">
```

```
<hr id="bgyellow" align="center" width="150" size="5">
```

```
<h2 id="bgyellow" align="center"> Мрії прогульника</h2>
```

```
<h2 class="normal"> Том 2. Повне зібрання студентських снів</h2>
```

```
<div id="first">Перший блок</div>
```

```
<div id="second">Другий блок</div>
```

```
</body>
```

```
</html>
```

Завдання 2.

Вставте будь який текст, який складається із трьох абзаців. Оформіть запропонований текст за допомогою аркушів стилів. З допомогою відступу зліва для тегу <div>, кожен абзац зробіть із зсувом вправо на 60 пунктів відносно попереднього, задайте різний колір тексту, фону, типу рамки і абзацного відступу.

Завдання 3.

Розробити web-сторінки, кожна з яких демонструє:

- усі значення властивості *position*
- властивість *float*
- властивість *clear*
- властивість *z-index*

Завдання 4.

Створіть із запропонованого тексту HTML – документ. Тест оформіть за допомогою стилів розміщених в зовнішньому файлі відповідно до наведених нижче вимог.

З допомогою тегу <div>, наведений нижче текст розташуйте у дві колонки з допомогою двох різних блоків. Над блоками розмістити заголовок «Програмування Internet».(див. структуру нижче)

Відстань між блоками — 20 пікселів, відстань між текстом і рамкою — 10 пікселів;

ширина першого блоку — 600 пікселів, другого — 400 пікселів;

відступ зверху — 100 пікселів;

позиціонування — абсолютне;

колір рамки червоний, тип рамки — подвійний;

фоновий малюнок — bg_menu.gif, повтор по осі x, по центру блоку (для першого блоку) і по верху блоку — (для другого блоку), фіксоване;

колір фону блоку — вирахувати із кольору останніх пікселів зображення bg_menu.gif;

висота блоків — 120%;

передбачити наявність смуг прокрутки у блоці, якщо текст не поміщається в блок;

абзацний відступ — 15 пікселів.

Вимоги до заголовків: 4-ий рівень, вирівняний по центру, напівжирний (в стилях задати один раз для всіх заголовків).

Окремо задати тип шрифту курсив для заголовків всередині блоку (задати одним правилом для двох заголовків методом групування параметрів).

Для всієї сторінки задайте білий фоновий колір, фоновий малюнок gamka.gif, який вирівняний по лівому краю сторінки, повторюється по осі Y. Порівняйте як поводить себе зображення при значення fixed і scroll властивості background-attachment при прокручуванні сторінки.

Програмування Internet

Історія виникнення Internet

У середині 60-років у самий розпал холодної війни міністерство оборони США планувало створити мережу для управління, яка б допомогла вижити в умовах ядерної війни. Стандартні телефонні мережі вважалися недостатньо надійними, оскільки вихід з ладу одного з центральних комутаторів може паралізувати цілий регіон (телефонна мережа має деревоподібну топологію).

Для вирішення завдання було залучено агентство ARPA (Advanced Research Project Agency). Це агентство було створено у відповідь на запуск у СРСР у 1957 році штучного супутника землі. Агентство не мало своїх лабораторій і вчених, і його бюджет був незначним (за масштабами Пентагону). ARPA вирішувало проблеми, видаючи гранти університетам і компаніям, чиї пропозиції виявлялися перспективними. Ними була досліджена можливість побудови мереж на основі перемикачів пакетів. Потім була побудована така мережа, що складається з субсетей і окремих ЕОМ. Субсеті склалися з IMP (Interface Message Processor), побудованих на міні-ЕОМ, з'єднаних каналами передачі даних.

Вже на цьому рівні передбачалася динамічна маршрутизація пакетів - і вихід з ладу окремого сайту або каналу призводив до того, що пакети починали рухатися в обхід пошкодженої ділянки. Кожен вузол складався з IMP і ЕОМ, з'єднаних коротким

Технологія конструювання програмного забезпечення

Технологія

конструювання ПЗ – це система інженерних принципів для створення економного ПЗ, яке надійно і ефективно працює на реальних комп'ютерах. Розрізняють методи, засоби і процеси технології конструювання ПЗ.

Методи дозволяють відповісти на питання, як створити програмний виріб, і охоплюють круг задач, пов'язаних з плануванням розробки, оцінкою термінів і витрат, з аналізом вимог, проектуванням, кодуванням і тестуванням програм, з супроводом програмного виробу. Метод програмної інженерії – це структурний підхід до створення ПЗ, націлений на створення ефективного продукту найбільш прибутковим (рентабельним) шляхом. Практично всі методи побудовані на ідеї створення графічних моделей системи з подальшим їх використанням в якості

<p>кабелем. EOM могла послати IMP повідомлення довжиною 8063 біта. IMP поділяв це повідомлення на кадри довжиною 1008 біт і пересилав їх адресату. Мережа працювала за схемою "запам'ятати і переслати", при якій пакет спочатку записується цілком у буфер і тільки потім передається далі.</p> <p>У 1968 році був організований тендер на створення експериментальної мережі. Тендер виграла компанія BBN. Мережа була побудована на базі міні-EOM DDP-316 з пам'яттю 12К 16-бітових слів. Машини були з'єднані за допомогою виділених ліній з пропускною здатністю 56 Кбіт / с. Програмне забезпечення було створено в 1969 році силами студентів-випускників місцевого університету. Програми базувалися на технології сокетів 4.2BSD. Саме тоді остаточно сформувалася ідеологія TCP / IP.</p> <p>До 1983 року мережа <i>ARPANET</i> містила вже більше 200 вузлів, а стек протоколів TCP / IP набув офіційного статусу. Тоді ж були введені в лад перша DNS (Domain Name System) сервери.</p> <p>У 1991 році конгрес США прийняв закон про створення мережі NREN (National Research and Education Network - національна мережа для науки і освіти) з каналами, розрахованими на швидкість передачі в діапазоні гігабіт / с. Таким чином, можна вважати, що Інтернету більше 30 років, а першого офіційного документу Інтернет (<i>RFC</i>) - більше 35.</p>	<p>специфікації, або архітектури системи. Методи повинні включати в себе наступні компоненти:</p> <ul style="list-style-type: none"> - Опис моделей системи і їх нотацій; - Правила, які накладають обмеження на використання моделей в системі; - Рекомендації, які характеризують хороші прийоми проектування в даному методі; - Керівництво до дії – опис дій, яким можна слідувати під час створення моделей і подальшого їх використання. <p>Немає ідеальних методів, усі вони застосовуються лише у тих чи інших випадках. Наприклад, об'єктно-орієнтовані методи добре підходять для інтерактивних систем, але не для систем реального часу.</p>
---	---

На щастя, ядерної війни не відбулося і, мабуть, Інтернет став одним з небагатьох позитивних результатів холодної війни. Історія розробки базових принципів та алгоритмів Інтернет показує, як при обмежених засобах можна вирішити глобальну проблему.

Перший документ RFC (Request for Comments) регламентує стек протоколів **TCP/IP**, побачив світ у квітні 1969 року. За перший рік було підготовлено лише 27 RFC. Далі активність у цій сфері почала зростати. Складалося ядро протоколів. До кінця 1982 були створені крім IP, базові поштові протоколи RFC-82122, ARP і TCP. У 1983 році оформився протокол DNS (Domain Name Service), який потрібен для перетворення імені мережевого об'єкта в IP-адресу. З цього моменту відкрилася можливість створювати мережні додатки типу telnet (віддалений доступ) і FTP (File Transfer Protocol). Темп підготовки документів RFC по роках відображений на рис. 1.1.

4.3 Контрольні питання

1. Для чого служать аркуші стилів?
2. У чому полягає різниця між зовнішніми та внутрішніми аркушами стилів?
3. Назвіть види селекторів і призначення кожного з них.
4. Які правила записів внутрішніх стилів?
5. Яка область застосування аркушів стилів?

Лабораторна робота №5. HTML. Семантичні елементи

Мета: навчитися створювати семантичні елементи та задавати їх стилі.

Термін: 2 год.

5.1 Теоретичні відомості

1. Семантичні елементи HTML5
2. CSS-властивість content

5.2 Хід роботи

Завдання 1.

Опрацюйте теоретичний матеріал, вказаний у посиланнях розділу “Теоретичні відомості”. Продумайте структуру та розробіть веб-сторінку (сторінки) на тему згідно Вашого варіанту, використовуючи всі семантичні елементи HTML5, перелічені у теоретичних відомостях.

Для заповнення вмісту блоків сайту з допомогою властивості content, додайте спеціальні символи, текст, зображення, блочний вміст, значення атрибуту, лапки; використайте форматування лапок: властивість quotes

Теми :

- | | |
|-----------------------|-------------------------|
| 1. Птахи | 13. Знаки зодіаку |
| 2. Риби | 14. Корисні копалини |
| 3. Звірі | 15. Комп'ютерна техніка |
| 4. Комахи | 16. Жіночий одяг |
| 5. Гриби | 17. Чоловічий одяг |
| 6. Автомобілі | 18. Дитячий одяг |
| 7. Дерева | 19. Взуття |
| 8. Фрукти | 20. Ювелірні |
| 9. Овочі | 21. Спортивні аксесуари |
| 10. Ягоди | 22. Музичні інструменти |
| 11. Лікарські рослини | 23. Автомобілі |
| 12. Планети | 24. Жіночий одяг |

- | | |
|----------------------|-------------------------|
| 25. Чоловічий одяг | 28. Автомобілі |
| 26. Знаки зодіаку | 29. Побутова техніка |
| 27. Корисні копалини | 30. Комп'ютерна техніка |

5.3 Контрольні запитання

1. Що таке семантичні елементи?
2. Для чого призначені семантичні елементи?
3. Як задати метадані документа?
4. Як задати розділи документа?
5. Які теги слугують для групування вмісту?

Лабораторна робота №6. CSS3. Стилзація елементів. Переходи, трансформації

Мета: навчатися стилізувати елементи, створювати переходи та трансформації.

Термін: 2 год.

6.1 Теоретичні відомості

1. Стилзація елементів
2. CSS Gradients
3. CSS3-оформлення тексту
4. CSS3-тінь блоку
5. CSS властивість transform

6.2 Хід роботи

Завдання 1.

Стилізуйте елементи попередньої роботи, використовуючи:

- о властивості зовнішнього контуру outline,
- о властивості resize (зміна розміру блоків),
- о властивості cursor (стилзація курсору),
- о властивості caret-color (колір каретки вставки),
- о CSS-градиент: linear-gradient() та radial-gradient(),
- о CSS3-рамку:
 - Заокруглення кутів за допомогою border-radius
 - Рамки-зображення border-image
 - Ширина рамки-зображення border-image-width
 - Ресурс рамки-зображення border-image-source
 - Елементи рамки-зображення border-image-slice
 - Повторення рамки-зображення border-image-repeat
 - Зміщення рамки-зображення border-image-outset
 - Градієнтна рамка
- о CSS3-оформлення тексту (text-decoration-....)

- о CSS3-тінь блоку (text-shadow).

Завдання 2.

Забезпечте плавні зміни властивостей деяких елементів попереднього завдання (при необхідності додайте нові елементи), використовуючи наступні властивості:

1. transition-property;
2. transition-duration;
3. transition-timing-function;
4. transition-delay;
5. короткий запис переходу transition;
6. плавний перехід кількох властивостей.

Завдання 3.

Створіть 2D-трансформацію для деяких елементів попередніх завдань (при необхідності додайте нові елементи), використовуючи:

- функції 2D-трансформації transform: matrix(a, c, b, d, x, y); друга функція – згідно варіанту:

№	Функція
1.	translate(x,y)
2.	translateX(n)
3.	translateY(n)
4.	scale(x,y)
5.	scaleX(n)
6.	scaleY(n)
7.	rotate(кут)
8.	skew(x-кут,y-кут)
9.	skewX(кут)

10.	skewY(кут)
-----	------------

- точку трансформації transform-origin;
- множинні трансформації.

Завдання 4.

Створіть 3D-трансформацію (об'єкти для трансформації оберіть згідно тематики роботи 5), використовуючи функції 3D-трансформації transform: matrix3d; друга функція – згідно варіанту:

№	Функція
1.	translate3d(x,y,z)
2.	translateZ(z)
3.	scale3d(x,y,z)
4.	scaleZ(z)
5.	rotate3d(x,y,z,кут)
6.	rotateX(кут)
7.	rotateY(кут)
8.	rotateZ(кут)
9.	perspective(n)
10.	translate3d(x,y,z)

6.3. Контрольні запитання

1. Як створити градієнт з допомогою CSS3?
2. Як створити рамку з допомогою CSS3?
3. Як задати тінь блоку з допомогою CSS3?
4. Як створити переходи з допомогою CSS3?
5. Як створити 2D -трансформації з допомогою CSS3?
6. Як створити 3D -трансформації з допомогою CSS3?

Лабораторна робота №7. CSS. Блочна верстка веб-сторінок. Властивість display. CSS Grid

Мета: опанувати можливості властивості display.

Тривалість: 2 год.

7.1 Теоретичні відомості

1. Блочні елементи, властивість display
2. Верстка на CSS Grid

7.2 Хід роботи

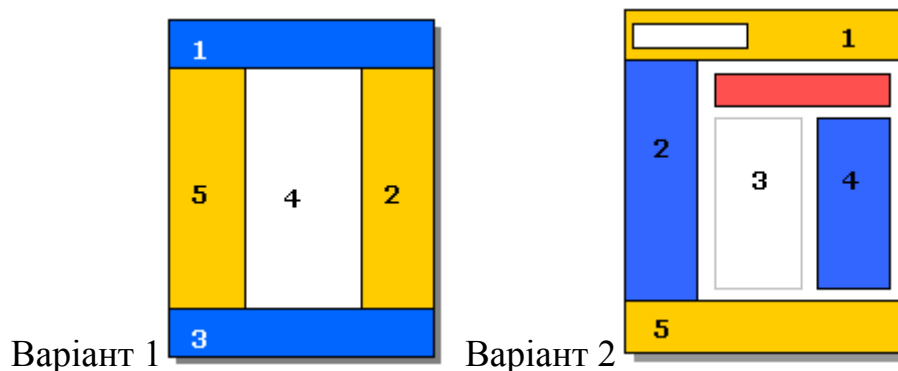
Завдання 1

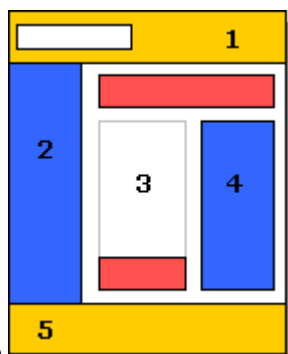
З допомогою вкладених блоків (властивість display: inline-block), а також інших стилів розробити інформаційну Web-сторінку, відповідно до свого варіанту (№ варіанту — це номер у списку групи, якщо Ваш номер від 1 до 10; і остача від ділення на 10 номеру у списку групи, якщо номер 11 і більше, у 20-го 10-й варіант), кольорова гама довільна, дати назву латиницею *Прізвище.html*).

Висота середнього рядка 70%. Блоки наповнити довільним вмістом, в одному з них розмістити великий за об'ємом текст. Передбачити появу смуг прокрутки для випадку, коли контент перевищує висоту.

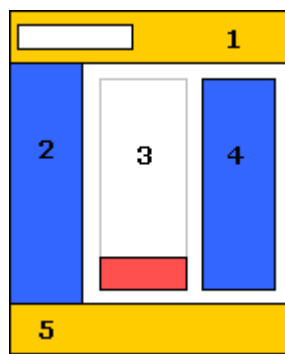
Один із блоків поверніть на деякий кут: rotate(кут)

Для заповнення блоків використайте будь-які три спецсимволи HTML (HTML-код).

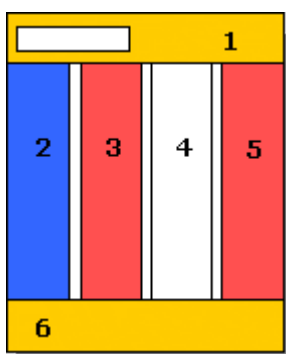




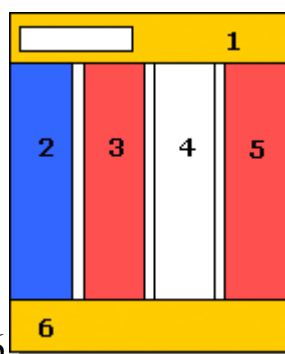
Варіант 3



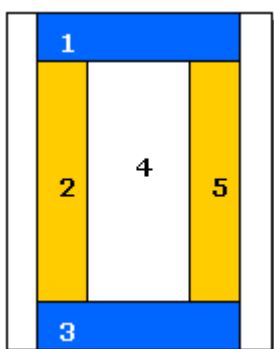
Варіант 4



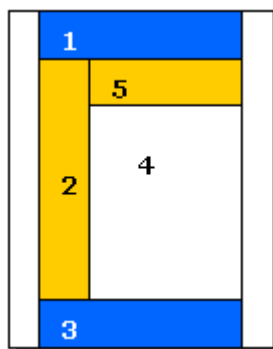
Варіант 5



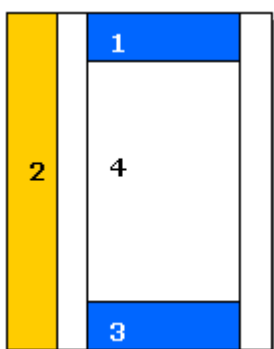
Варіант 6



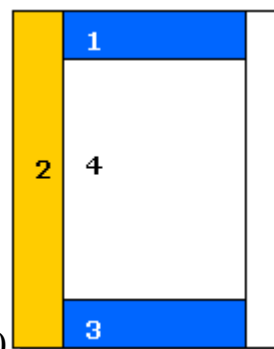
Варіант 7



Варіант 8



Варіант 9



Варіант 10

Завдання 2.

Використовуючи метод верстки CSS Grid, а також інші стилі розробити інформаційну Web-сторінку, відповідно до свого варіанту (див. попереднє завдання, кольорова гама довільна).

З використанням правила `@media` (`min-width`, `max-width`, `print`), задати і продемонструвати різні стилі в залежності від роздільної здатності пристрою

виводу (зробити мінімум для двох варіантів: для монітору ПК і смартфона) та при виведенні сторінки на друк. Адаптивну зміну розміру та розташування блоків, стилі елементів елементів задати на свій розсуд, але з обов'язковою зміною структури сторінки.

Блоки наповнити довільним вмістом, в одному з них розмістити великий за об'ємом текст. Передбачити появу смуг прокрутки для випадку, коли контент перевищує висоту.

7.3 Контрольні запитання

1. Які ви знаєте блочні елементи?
2. Які ви знаєте рядкові елементи?
3. Які ви знаєте значення властивості `display`?
4. Які переваги верстки на CSS Grid?

Лабораторна робота №8. CSS. Метод верстки Flexbox. Правила. Функції.

Мета: навчитися верстати веб-сторінки методом верстки Flexbox, задавати різні стилі в залежності від роздільної здатності та типу пристрою виводу та приєднювати зовнішні шрифти.

Тривалість: 2 год.

8.1 Теоретичні відомості

1. Основні поняття Flexbox
2. CSS правила
3. CSS функції

8.2 Хід роботи

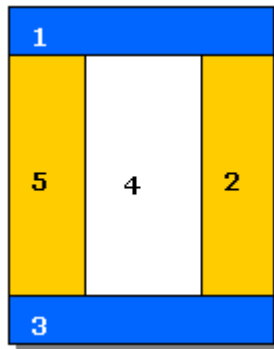
Завдання 1.

Використовуючи ТІЛЬКИ метод верстки Flexbox, а також інші стилі розробити інформаційну Web-сторінку, відповідно до свого варіанту (№ варіанту — це остача від ділення на 10 номеру у списку групи, кольорова гама та наповнення довільні).

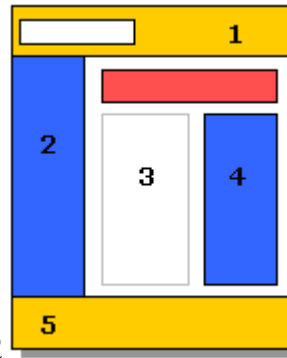
З використанням правила @media (min-width, max-width, print), задати і продемонструвати різні стилі в залежності від роздільної здатності пристрою виводу (зробити як мінімум, для монітору ПК і смартфона) та при виведенні сторінки на друк.

Блоки наповнити довільним вмістом, в одному з них розмістити великий за об'ємом текст. Передбачити появу смуг прокрутки для випадку, коли контент перевищує висоту.

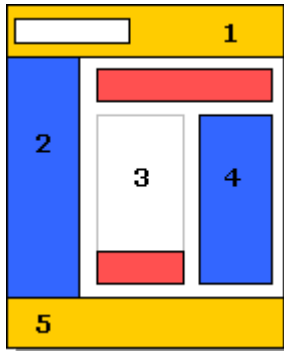
Варіанти структури веб-сторінки:



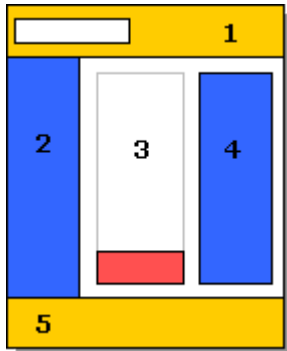
Вариант 1



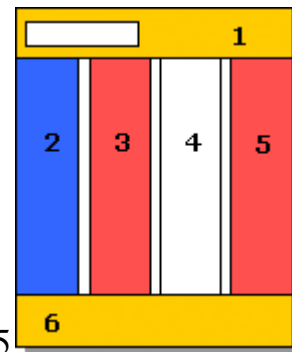
Вариант 2



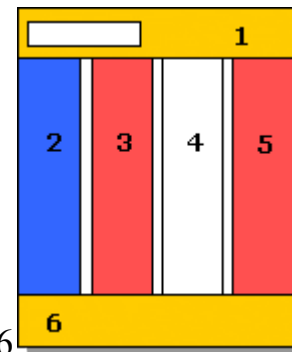
Вариант 3



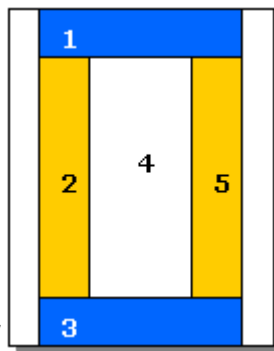
Вариант 4



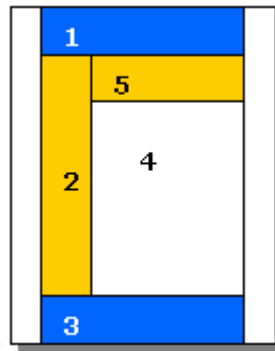
Вариант 5



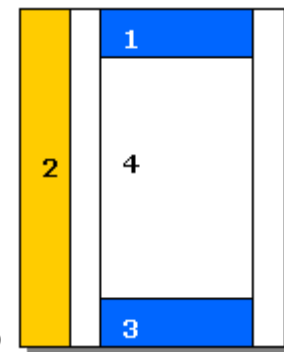
Вариант 6



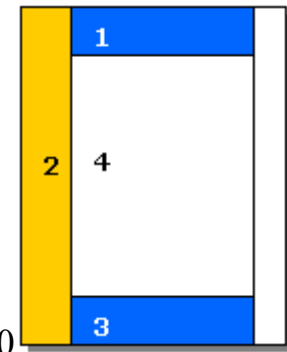
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Завдання 2.

Продемонструйте виконання функцій `attr()` та `calc()`.

Завдання 3

Для тексту сторінки із завдання 1 підключіть як мінімум два нестандартних шрифта, один з допомогою правила `@import`, інший з допомогою правила `@font-face`.

Потрібні шрифти можете завантажити із сайту <https://fonts.google.com/> чи знайти на інших ресурсах.

На сайті <https://fonts.google.com/>: у полі Search fonts введіть назву потрібного шрифту або виберіть із запропонованого переліку і клікніть на Download family (1). Для встановлення потрібного шрифту на ПК, розархівуйте завантажені файли і викличте контекстне меню для файлу із розширенням ttf, далі Встановити.

Для того, щоб шрифт заданий на вебсторінці, яка розміщена на вебхостингу, коректно відображався на будь-якому пристрої, незалежно від встановлених на ньому шрифтів, підключіть до вебсторінки потрібні шрифти. Для цього клікніть на “+ Select this style” біля необхідного Вам вигляду шрифту, далі скопіювати і вставити в заголовок веб-сторінки необхідний код (2)

The screenshot displays the Google Fonts interface for the 'Caveat' font family. At the top right, a blue button labeled 'Download family' is circled in red and marked with a red '1'. Below the font name, there are tabs for 'Select styles', 'Glyphs', 'About', and 'License'. The 'Styles' section features a text input field containing 'Almost before we knew it, we had left the ground' and a size selector set to '64px'. To the right, a panel shows the font's styles: 'Regular 400' and 'Medium 500'. Below this, there is a section for embedding the font, with the '@import' option selected and circled in red, marked with a red '2'. The corresponding CSS code is shown below, with the `@import` line underlined in red.

Приклад підключення шрифту:

```
<style>
```

```
@import
url('https://fonts.googleapis.com/css2?family=Caveat&display=swap');
</style>
```

Приклад підключення шрифту з допомогою правила `@font-face` (див. відео [Підключаємо шрифти](#)).

```
@font-face {
  font-family: 'MyWebFont';
  src: url('myfont.woff2') format('woff2'),
       url('myfont.woff') format('woff'),
       url('myfont.ttf') format('truetype');
}
body {
  font-family: 'MyWebFont', Fallback, sans-serif;
}
```

Даний спосіб є кращим з точки зору швидкості завантаження сторінки.

8.3 Контрольні запитання

1. Назвіть особливості методу верстки Flexbox?
2. Для чого використовується правило `@media`?
3. Для чого використовується правило `@import`?
4. Для чого використовується правило `@font-face`?

Лабораторна робота №9. Верстка вебсторінок за допомогою фреймворка Bootstrap

Мета: навчитися створювати адаптивні веб-сторінки за допомогою фреймворка Bootstrap.

Тривалість: 2 год.

9.1 Теоретичні відомості

Bootstrap - це css і js-фреймворк, по суті, набір файлів з готовим написаним кодом. Мета розробників практично будь-якого фреймворка - спростити розробку сайту собі і іншим, кому буде доступний інструмент. Bootstrap є повністю безкоштовний, так що ви можете використовувати його будь-яким чином, редагувати вихідний код і як завгодно переробляти фреймворк під себе.

Початок роботи Bootstrap

Для того, щоб підключити файли фреймворку до html-документу, просто підключіть його на пряму із офіційного сайту getbootstrap.com, як показано у прикладі нижче або встановіть як показано на офіційному сайті.

Короткий огляд цих файлів:

1. bootstrap.css і bootstrap.min.css - не стиснута і стиснута версії css-коду фреймворка. До робочого проекту рекомендується підключати стиснутий файл, таким чином ви поліпшите швидкість завантаження. Але якщо ви плануєте дивитися чи змінювати код у файлі, підключайте нестиснуту версію.

2. bootstrap.js і bootstrap.min.js - файл зі скриптами на мові JavaScript.

3. папка fonts і в ній файли glyphs - це іконочні шрифти Bootstrap. У ньому близько 200 ікон. Для більшості випадків вам їх вистачить, іноді потрібне підключення інших.

Приклад підключення стислої версії css-коду фреймворка і трьох блоків, які будуть відображатися однаково на всіх пристроях. Ми розділимо екран на три рівні частини:

```

<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <div class="row">
      <div class="col-sm-4">Перша колонка</div>
      <div class="col-sm-4">Друга колонка</div>
      <div class="col-sm-4">Третя колонка</div>
    </div>
  </body>
</html>

```

Детальніше із Bootstrap можна ознайомитися тут:
<https://getbootstrap.com>

9.2 Хід роботи

1. Створити елементи сайту згідно цього:
<https://www.youtube.com/watch?v=c9B4TPnak1A>
2. З використанням фреймворка Bootstrap, зверстати адаптивну головну сторінку інтернет-магазину згідно свого варіанту. Адаптивність перевірити шляхом зменшення розміру екрану. Контент та стилістику обрати самостійно.

Обов'язково на сайті мають бути:

- панель навігації;
- випадаюче меню;
- вітальний блок розділений на три колонки різної ширини;
- слайдер(Carousel);
- кнопки різних типів;
- форми;
- зображення з коротким описом і кнопкою для переходу на детальніший опис (Cards);
- інші елементи.

Варіанти:

- | | | |
|-------------------------|---------------------------|-------------------------|
| 1. Домашні птахи | 12. Двоколісний транспорт | 21. Музичні інструменти |
| 2. Їжа | 13. Меблі | 22. Транспорт |
| 3. Домашні тварини | 14. Комп'ютерна техніка | 23. Кухонна техніка |
| 4. Гриби | 15. Жіночий одяг | 24. Побутова техніка |
| 5. Транспорт | 16. Чоловічий одяг | 25. Автомобілі |
| 6. Деревя | 17. Дитячий одяг | 26. Комп'ютерна техніка |
| 7. Фрукти | 18. Взуття | 27. Птахи |
| 8. Овочі | 19. Солодощі | 28. Їжа |
| 9. Ягоди | 20. Спортивні аксесуари | 29. Домашні тварини |
| 10. Косметика | | 30. Транспор |
| 11. Кондитерські вироби | | |
| 31. т | | |

9.3 Контрольні запитання

1. Bootstrap - що це?
2. Що таке Bootstrap - сітка і де її можна подивитися?
3. Від чого залежать розміри Bootstrap - сітки
4. Яка різниця між файлами bootstrap.css і bootstrap.min.css?

Лабораторна робота №10. Вебформи. Псевдоелементи. Псевдокласи.

Мета: ознайомтеся з основними елементами форми та їх атрибутами та із псевдокласами і псевдоелементами.

Тривалість: 2 год.

10.1 Теоретичні відомості

1. Вебформи

10.2 Хід роботи

Завдання 1.

Ознайомтеся із основними елементами вебформи та їх атрибутами.

Використовуючи вебформу, створити вебсторінку (-ки), яка містить (при створенні елементів форми використовувати максимальну кількість тегів та атрибутів):

1. Демонстрацію однорядкового текстового поля веб-форми.
2. Форму, що використовує багаторядкову текстову область.
3. Демонстрацію значень password, email, range властивості type тегу input.
4. Форму, що використовує два незалежні перемикача єдиного вибору (radio) і кнопку відправки даних.
5. Приклад активізації за замовчуванням одного з варіантів вибору, що надаються перемикачем.
6. Демонстрацію використання прапорців для множинного вибору варіантів із запропонованого переліку.
7. Приклад використання випадаючого списку єдиного вибору та групування елементів списку (optgroup).
8. Демонстрація списку множинного вибору.
9. Приклад форми з можливістю скидання в початковий стан(
<input type="reset" value="Очистить форму" />).

10. Демонстрація використання семантично нейтральній кнопки і зв'язки її з обробником події. (`<input type="button" value="Click me!" onClick='alert("Oh, you have been clicked me!!!")'>`).

11. Згрупуйте деякі елементи, використовуючи тег `<fieldset>`

Завдання 2

Продемонструйте виконання наступних псевдоелементів: `: : after`, `: : before`, `: : first-letter`, `: : first-line`

Завдання 3

Задайте різні стилі для посилань, використовуючи псевдокласи: `: active`, `: hover`, `: link`, `: visited`

Завдання 4

Продемонструйте виконання псевдокласів `: valid`, `: invalid` та інших згідно варіанту:

№	Псевдокласи		
1.	<code>: checked</code>	<code>: not(селектор)</code>	<code>: read-write</code>
2.	<code>: disabled</code>	<code>: nth-child(n)</code>	<code>: required</code>
3.	<code>: empty</code>	<code>: nth-last-child(n)</code>	<code>: root</code>
4.	<code>: enabled</code>	<code>: nth-last-of-type(n)</code>	<code>: target</code>
5.	<code>: first-child</code>	<code>: nth-of-type(n)</code>	<code>: checked</code>
6.	<code>: first-of-type</code>	<code>: only-child</code>	<code>: disabled</code>
7.	<code>: last-child</code>	<code>: out-of-range</code>	<code>: first-child</code>
8.	<code>: focus</code>	<code>: only-of-type</code>	<code>: empty</code>
9.	<code>: in-range</code>	<code>: optional</code>	<code>: enabled</code>
10.	<code>: last-of-type</code>	<code>: read-only</code>	<code>: first-of-type</code>

10.3 Контрольні запитання

1. Назвіть основні елементами форми.
2. Назвіть атрибутами тегу `input` та їх призначення
3. Назвіть, які ви знаєте псевдокласи та їх призначення
4. Назвіть, які ви знаєте псевдоелементи та їх призначення

Лабораторна робота №11. Верстка сайтів на основі PSD-макетів.

Мета: навчитися верстати сайти на основі PSD-макетів.

Термін: 2 год.

11.1 Теоретичні відомості

1. Лекція по додатку Photopea
2. Підключення шрифтів

Якщо ви шукаєте шрифт для кирилиці, потрібно це вказати обравши Language.

11.2 Хід роботи

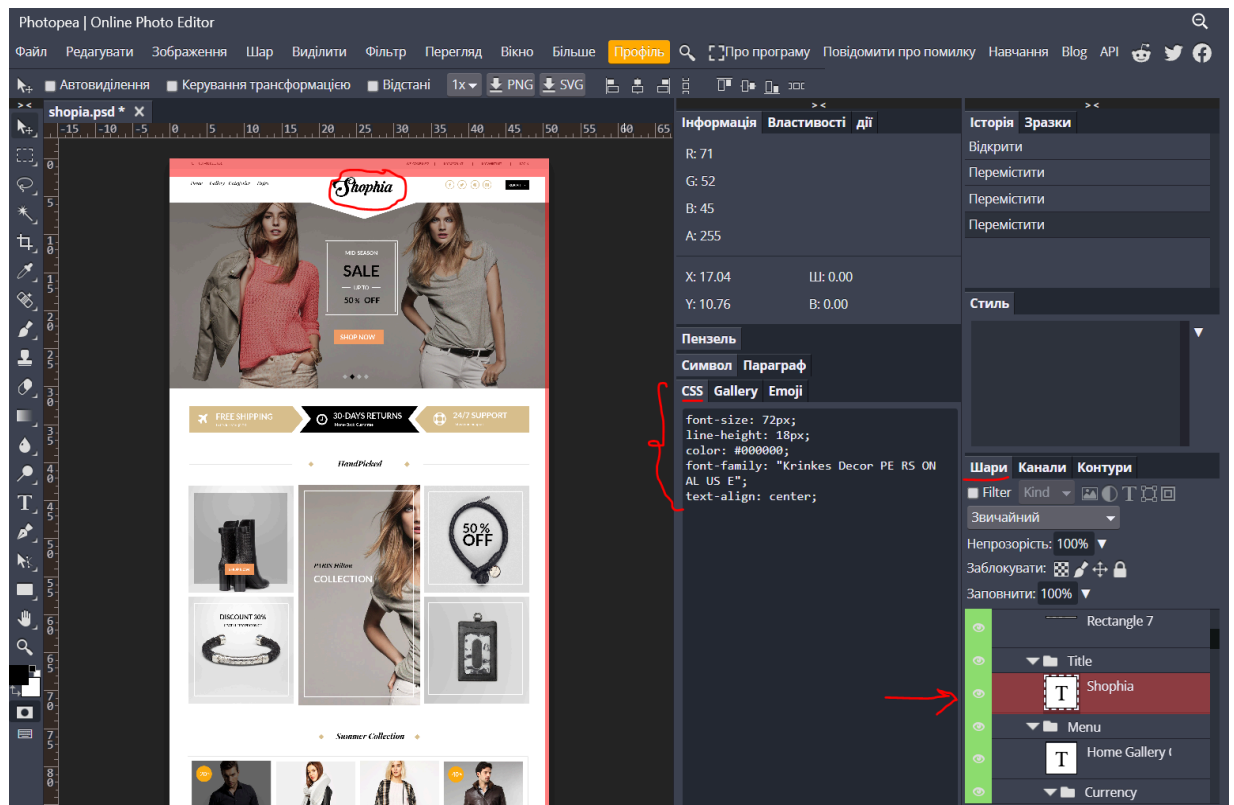
Завдання 1

В папці **PSD** оберіть варіант макету згідно порядкового номера за списком групи, відкрийте його з допомогою графічного редактора, наприклад photopea.com та виконати верстку усіх сторінок макету, тип верстки - адаптивна: з використанням правила @media (min-width, max-width), задати і продемонструвати різну верстку сторінок в залежності від роздільної здатності пристрою виводу (зробити мінімум для двох варіантів пристроїв: для монітору ПК (згідно макету заданого у варіанті) і смартфона (верстака на свій розсуд на основі елементів і стилістики, заданих у макеті згідно варіанту).

Для встановлення photopea.com на ПК зайдіть у веб версії програми в пункт меню Більше-> встановити Photopea.

Щоб переглянути стилі елементів макету, клікніть на них на панелі шарів і перейдіть на панель CSS. Щоб вона відображалася, клікніть на Вікно->Більше->біля CSS має стояти позначка. Якщо позначка є, а панель CSS не бачите, клікніть на кнопку розгортання панелей “< >”. “> <” - Згортає панелі.

Щоб увімкнути/вимкнути будь-який елемент вікна, перейдіть в меню Вікно.



11.3 Контрольні запитання

1. Що таке PSD-макет?
2. Для чого використовують правило @media?
3. Які особливості адаптивної верстки?
4. Які особливості гнучкої верстки?
5. Які особливості фіксованої верстки?

Список літератури

1. Український веб-довідник
2. CSS Tutorial
3. Електронний HTML і CSS довідник українською мовою
4. Блочні елементи, властивість display
5. Стилізація таблиць
6. Семантичні елементи HTML5
7. Основні поняття Flexbox
8. Верстка на CSS Grid
9. Bootstrap
10. Вебформи
11. Підключення шрифтів

Додатки. Реєстрація на вебхостингу та налаштування FTP-з'єднання для роботи з ним за допомогою файлових менеджерів і редактора вихідного коду.

Додаток А. Етапи реєстрації облікового на вебхостингу <https://infinityfree.net/>:

app.infinityfree.net/register

роллейбус (фир... Как читаются англ... Побудова бази зна... jQuery Form Submi... PHP and jQuery Co... Ивано-Франковск...

Sign up for a free account

Email address
marichkadu@gmail.com

Password

Confirm Password

I've read and agree to the terms of service.

app.infinityfree.net/accounts

Hosting Accounts

←

Ads by Google

Stop seeing this ad Why this ad? ⌵

✔ Success!
Your email address has been verified! You can now create your hosting account.

Your Accounts

USERNAME	LABEL	STATUS
No accounts yet. Create an account now.		

Active Accounts: 0 / 3

+ Create Account

app.infinityfree.net/accounts/create/step1

Create a Hosting Account

Step 1. Choose a domain name Step 2. Enter additional information Step 3. Done

Domain Type

Subdomain Custom Domain

Subdomain: dutchak Domain Extension: .infinityfreeapp.com

You can add more domains after your account has been created.

Search Domain

app.infinityfree.net/accounts/create/step2

Create a Hosting Account

Step 1. Choose a domain name Step 2. Enter additional information Step 3. Done

Account Label: Website for dutchak.infinityfreeapp.com


Account Username: (generated automatically)


Account Password: [REDACTED]

I'm not a robot reCAPTCHA

Create Account

Create a new Hosting Account

 **Write With Confidence**
Real-time suggestions whenever you write. Try Grammarly today Grammarly [Learn More](#)

 **Success!**
Your account has been created with username epiz_29636855!

Create a Hosting Account

Step 1. Choose a domain name Step 2. Enter additional information Step 3. Done

[View in Client Area](#) [Open Control Panel](#)

Hosting Accounts



Write With Confidence

Check your grammar, spelling, and punctuation instantly with Grammarly Grammarly

Learn More

Your Accounts

USERNAME	LABEL	STATUS	
epiz_29636655	Website for dutchak.infinityfreeapp.com	Active	Manage

Active Accounts: 1 / 3

[+ Create Account](#)

[Control Panel](#)

[File Manager](#)

Account Details

Username	epiz_29636655
Password	***** Show/Hide
Status	ACTIVE
Label	Website for dutchak.infinityfreeapp.com
Main Domain	qs2oypiy.epizy.com
Website IP	185.27.134.216
Hosting Volume	vol6_1
Created on	2021-09-05

FTP Details

FTP Username	<u>epiz_29636655</u>
FTP Password	***** Show/Hide
FTP Hostname	<u>ftpload.net</u>
FTP Port (optional)	21

MySQL Details

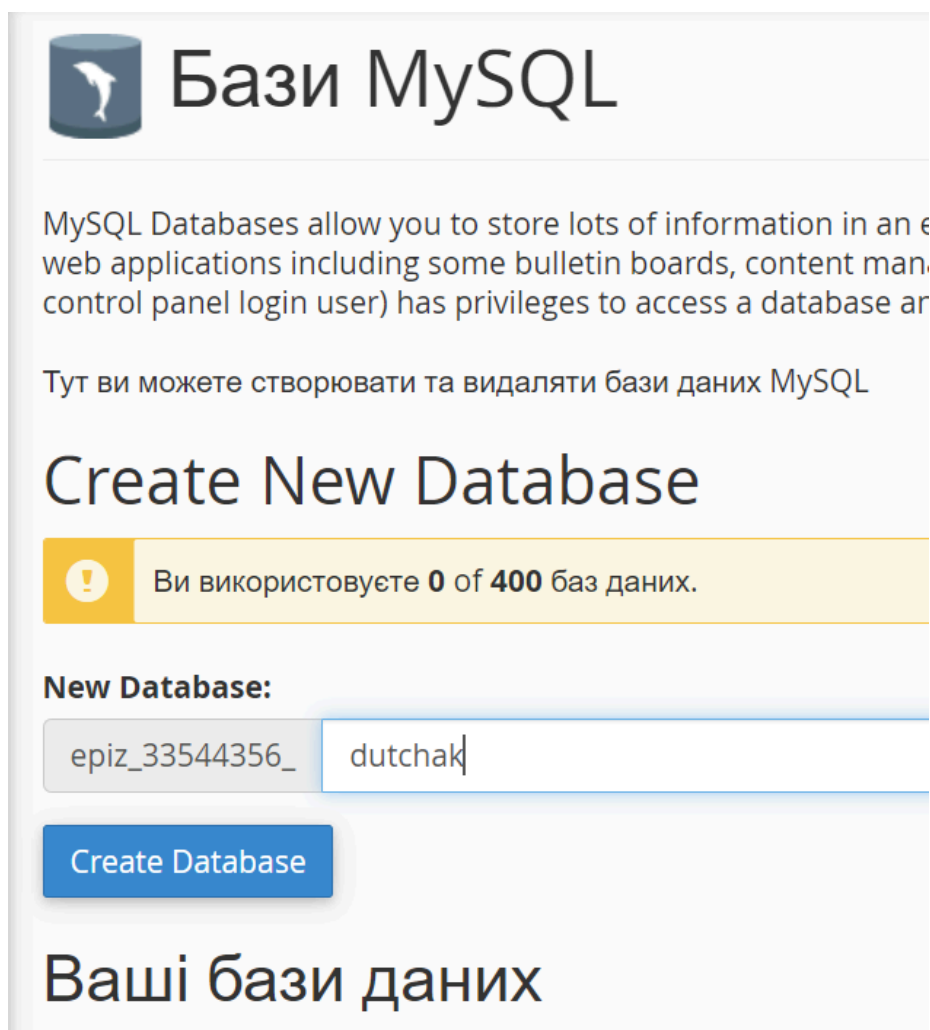
MySQL Username	epiz_29636655
MySQL Password	***** Show/Hide
MySQL Hostname	sql200.epizy.com
MySQL Port (optional)	3306
Database Name	epiz_29636655_XXX (create this in the control panel)

Domains and Subdomains

DOMAIN	TYPE	FILE MANAGER
<u>dutchak.infinityfreeapp.com</u>	Subdomain	Files

1 total domains [View All Domains](#)

Для створення бази даних, перейдіть у Control panel -> Бази MYSQL,
далі:



The screenshot shows the MySQL Databases management interface. At the top, there is a header with the MySQL logo and the text "Бази MySQL". Below the header, there is a brief description of MySQL Databases and a note about user privileges. The main heading is "Create New Database". A yellow warning box indicates that the user is using 0 out of 400 databases. Below this, there is a form for creating a new database. The form has a label "New Database:" and two input fields. The first field contains "epiz_33544356_" and the second field contains "dutchak". A blue "Create Database" button is positioned below the input fields. At the bottom of the visible area, the text "Ваші бази даних" is displayed.

Бази MySQL

MySQL Databases allow you to store lots of information in an e web applications including some bulletin boards, content man control panel login user) has privileges to access a database ar

Тут ви можете створювати та видаляти бази даних MySQL

Create New Database

Ви використовуєте **0** of **400** баз даних.

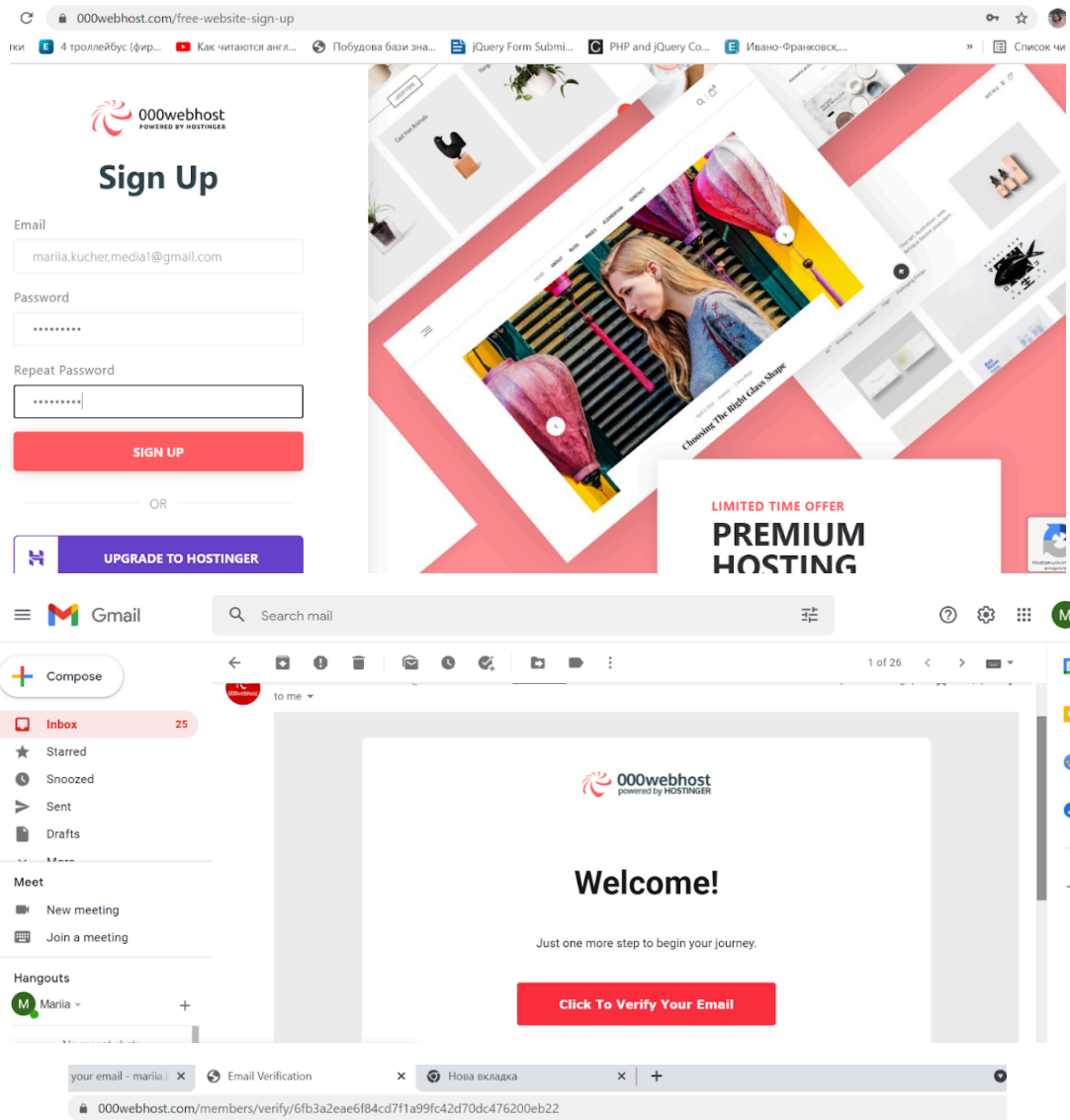
New Database:

epiz_33544356_ dutchak

Create Database

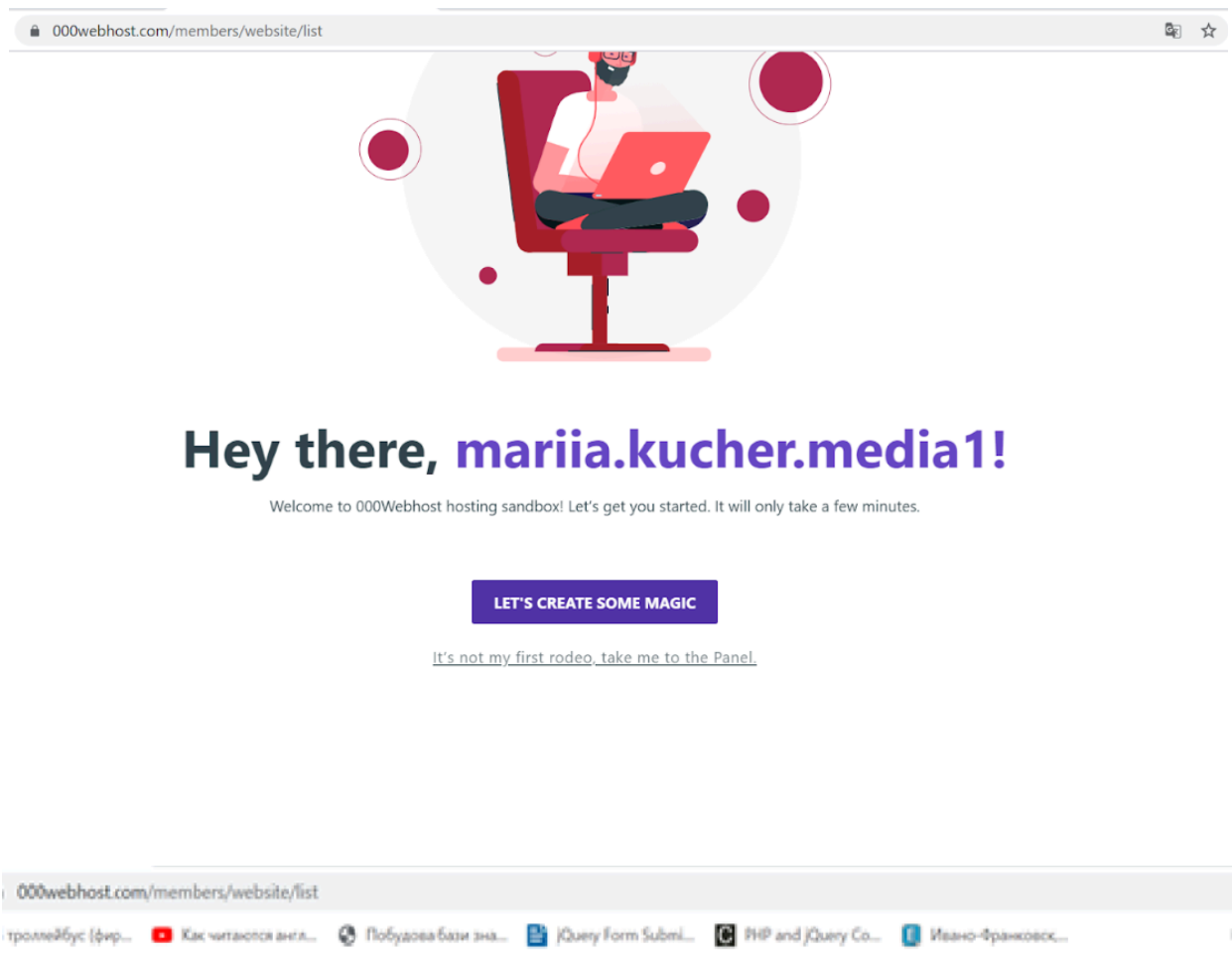
Ваші бази даних

Додаток Б. Етапи реєстрації на вебхостингу <https://www.000webhost.com/>

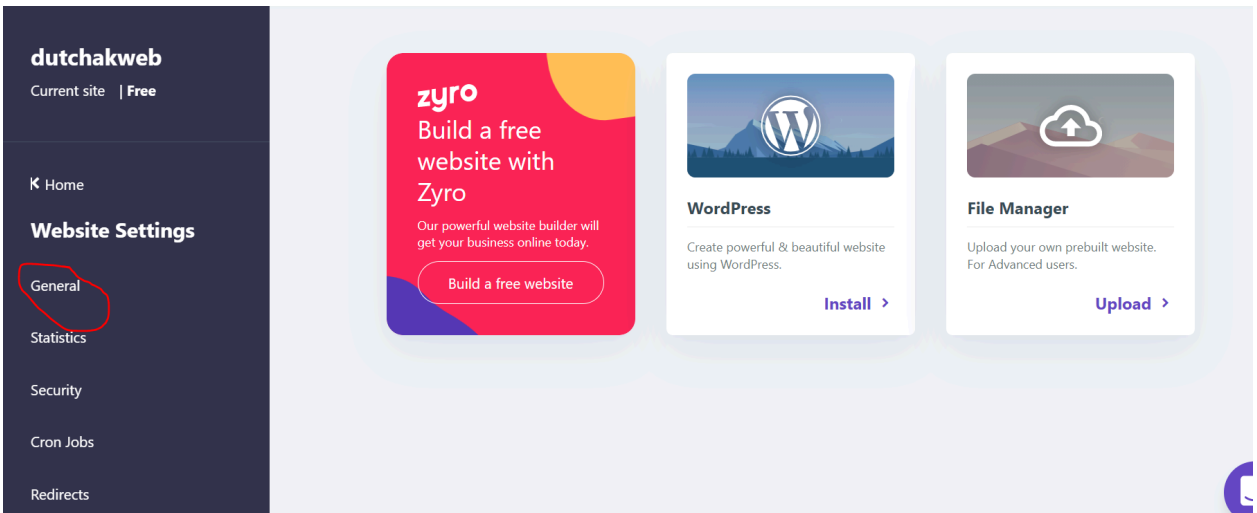
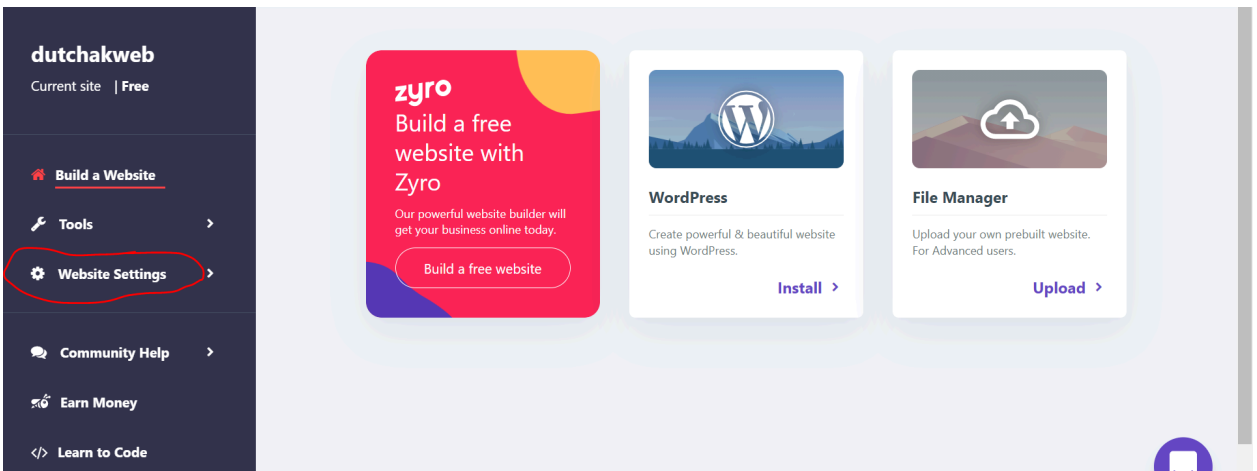
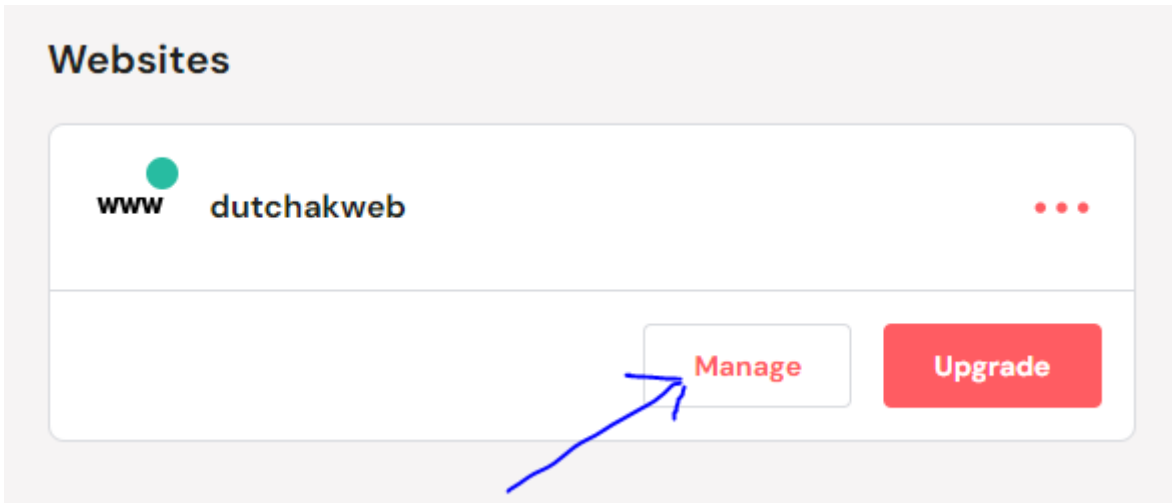


Email verified!

LOG IN



Десь на цьому етапі пропонує вибір типу сайту, то обирайте перше, щось типу “Learn development web”



dutchakweb  link to site

Dashboard

Tools

Website Settings

General

Security

Cron Jobs

Redirects

Logs

Backups

Community Help

General Settings

home - dutchakweb - General Settings

FTP Details

Here you can set preferences to manage access to your website files. By disabling this feature you will not be able to access the Web File Manager.

FTP transfer



Host Name

files.000webhost.com

Port

21

Username

dutchakweb

Password

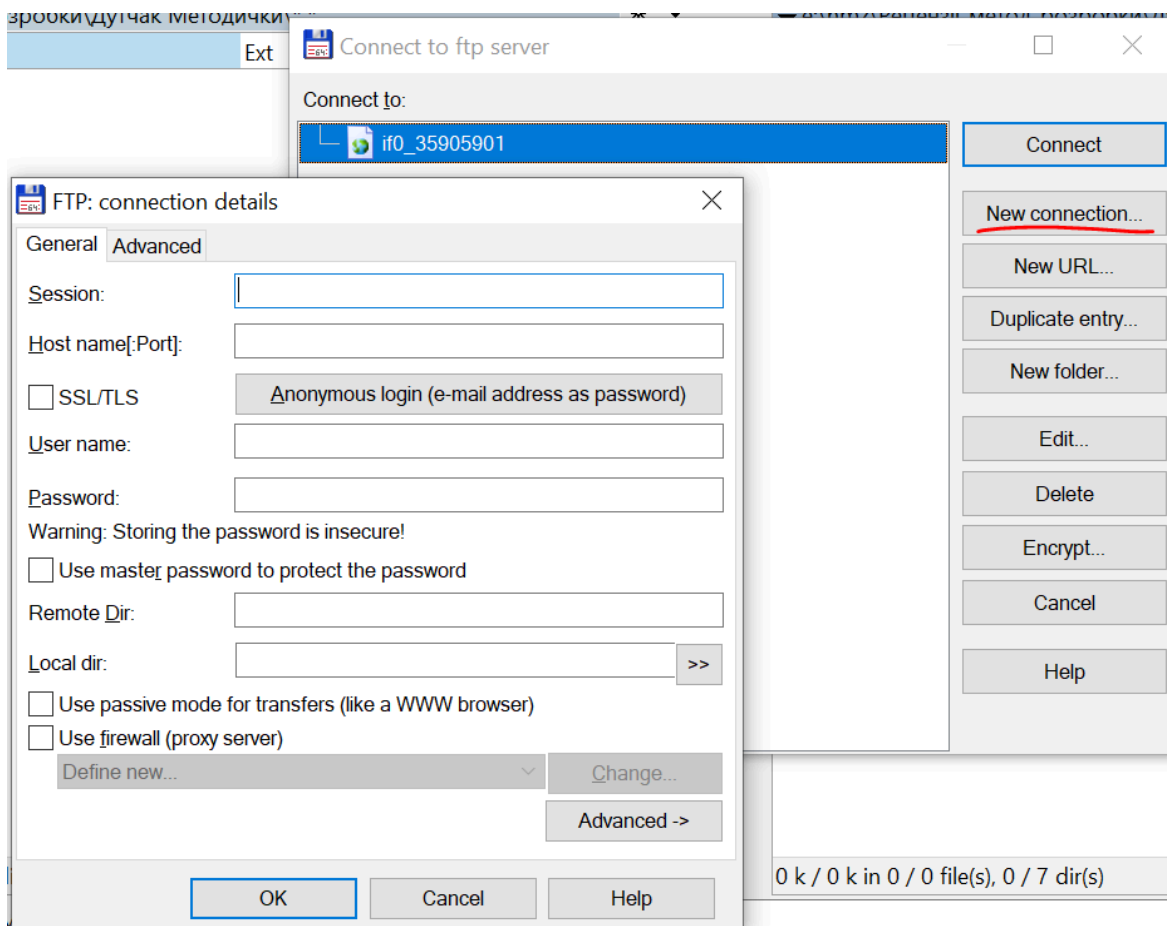
Same as your website password

Password

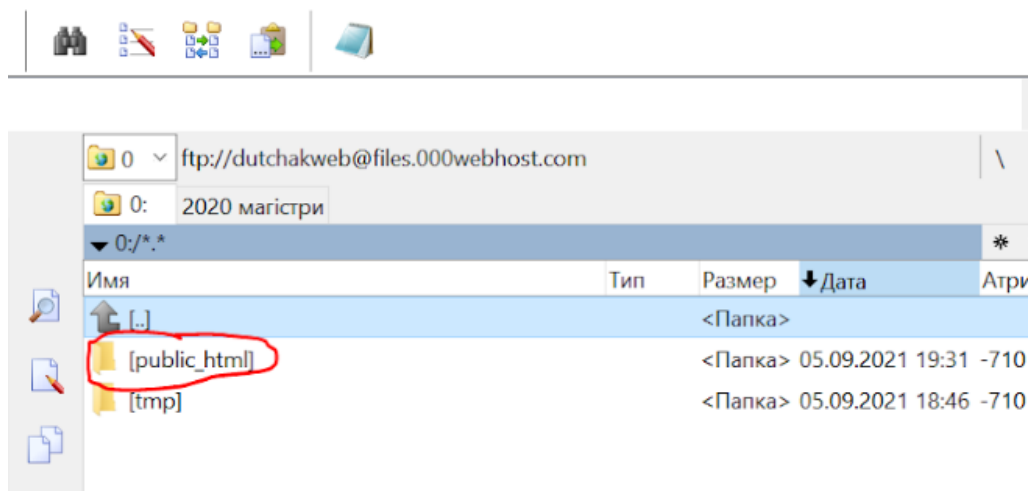
Change your website password here. This is also FTP password.

Додаток В. Налаштування FTP-з'єднання у програмах Total Commander, Файловий провідник та Far Manager

У програмі Total Commander заходите в меню програми «Мережа/З'єднання з ftp-сервером» (Ctrl+F), далі «Нове з'єднання» і ввести налаштування сервера (із закладки «Website Settings/General» на <https://000webhost.com>, Account Details на <https://infinityfree.net>).



Далі обираємо щойно створене з'єднання і клікаємо на «З'єднати». Після з'єднання із ftp-сервером, перейти у папку public_html і туди скопіювати необхідні web-файл. Файл index.html (index.php тощо) є файлом стартової сторінки сайти.



Файловий провідник

Якщо не вдалося з'єднатися із FTP через Total Commander, можете приєднатися через програму Файловий провідник. У адресному рядку потрібно вказати:

`ftp: //user_name@host_name`

Наприклад:

`ftp: //epiz_33544356@ftpupload.net`

`ftp: //dutchakweb@files.000webhost.com`


Налаштування FTP-єднання у файловому менеджері Far Manager


У файловому менеджері Far Manager: Alt+F1 (або F2) -> Net Box -> Shift+F4 і ввести налаштування сервера (із закладки "Website Settings/General" на `https: //000webhost.com`, Account Details на `https: //infinityfree.net`).


Щоб з'єднатися Alt+F1 (або F2) -> Net Box -> далі обираєте потрібну назву.

Far Manager також дозволяє напряду редагувати файли, розміщені на віддаленому сервері, для редагування клікаєте F4.

Додаток Г. Налаштувати VS Code для роботи із FTP-сервером хостингу.

Запустіть програму **VS Code**. Викличте меню додатків  Extensions

або Ctrl+Shift+X і встановіть розширення  SFTP SFTP/FTP sync Natizyskunk. (якщо встановлений, ще раз не потрібно встановлювати)


Клікніть зліва вверху на  Explorer або Ctrl+Shift+E, далі Open Folder відкрийте папку дисципліни. Після того, як вона з'явиться у Вашому робочому просторі, виділіть її, далі клікнути Ctrl+Shift+P і ввести sftp: config.

У файлі налаштувань sftp.json внести свої дані:

```
{
  "name": "My Server",
  "host": "host_name",
  "protocol": "ftp",
  "port": 21,
  "username": "your Username",
  "password": "your Password",
  "remotePath": "/",
  "uploadOnSave": true,
  "useTempFile": true,
  "openSsh": true,
  "downloadOnOpen": true,
  "ignore": []
}
```

Якщо Ви працюєте не за власним ПК, пароль можна не вносити, тоді його потрібно буде вводити в діалогове вікно. Або вказати пароль, а в кінці роботи видалити.

Збережіть і закрийте файл sftp.json.

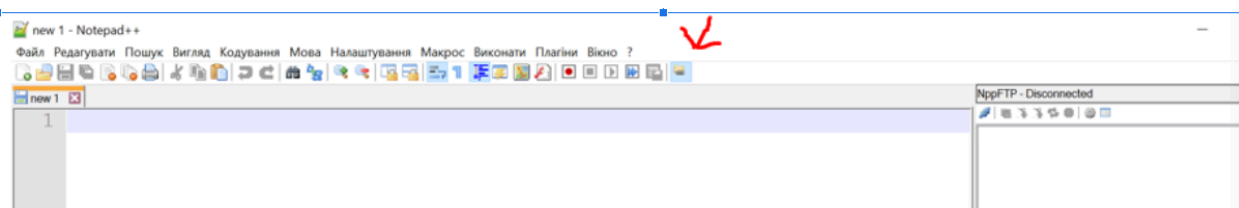
Якщо все зроблено вірно, зліва появиться піктограма SFTP , перехід на яку дозволяє переглядати і редагувати файли на Вашому FTP-сервері. Відредагуйте будь-який файл із Вашого хостингу, після чого кореневий каталог Вашого хостингу і файл, який Ви редагували, появиться в локальній папці дисципліни.

Якщо редагування файлу недоступне, у контекстному меню вибрати Edit in local. Зберегти зміни.

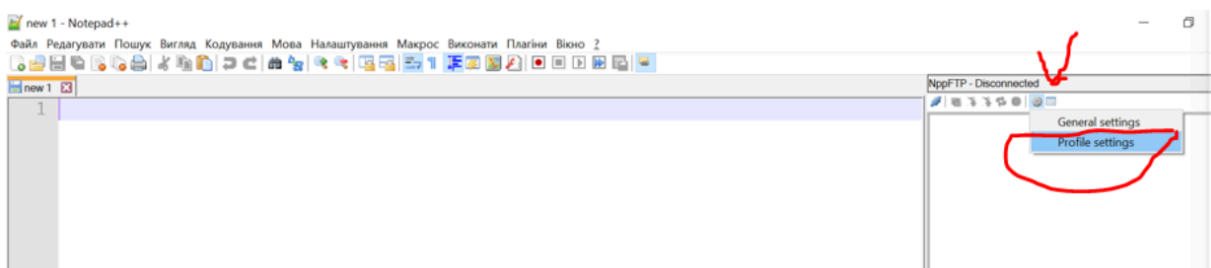
Щоб в локальній папці дисципліни появились всі файли і папки хостингу, їх потрібно скопіювати з хостингу (наприклад, з допомогою ТС) в кореневий каталог хостингу локальної папки. Тоді ці папки будуть повністю аналогічні і зміни в одній з них будуть автоматично вноситися в іншу. Крім того, Ви будете мати змогу контролювати для них версії і відправляти на віддалений репозиторій.

Додаток Д. Налаштування ftp-з'єднання за допомогою програми Notepad++

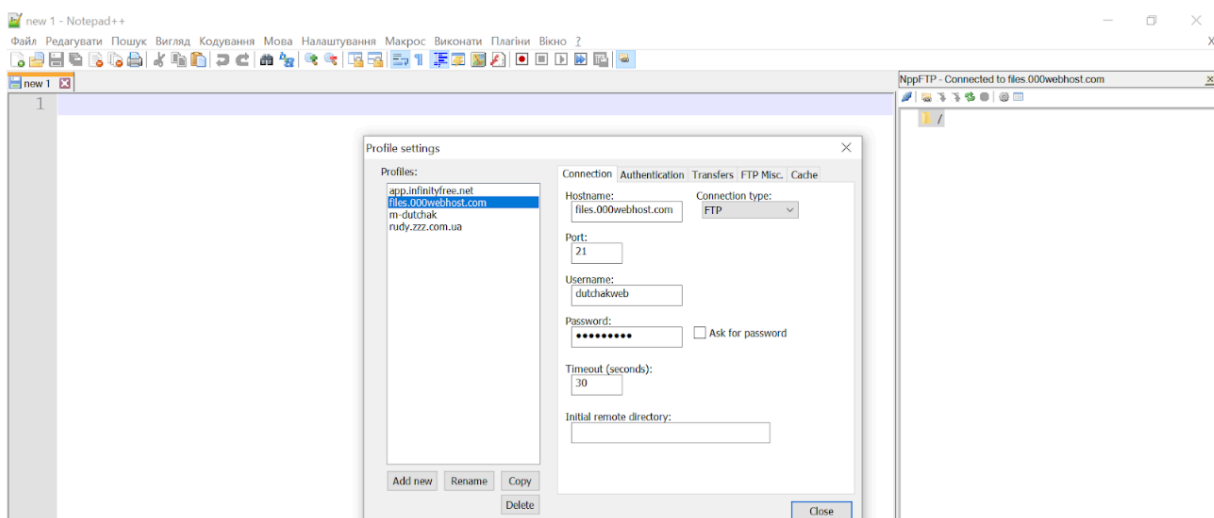
Запускаємо Notepad++ і в панелі інструментів клікаємо мишкою на значок "Show NppFTP Window" (див наступні скрині). Якщо відсутня дана можливість, потрібно завантажити плагін NppFTP1 (файл NppFTP.dll із папки bin копіюємо в c: \Program Files\Notepad++\plugins\)



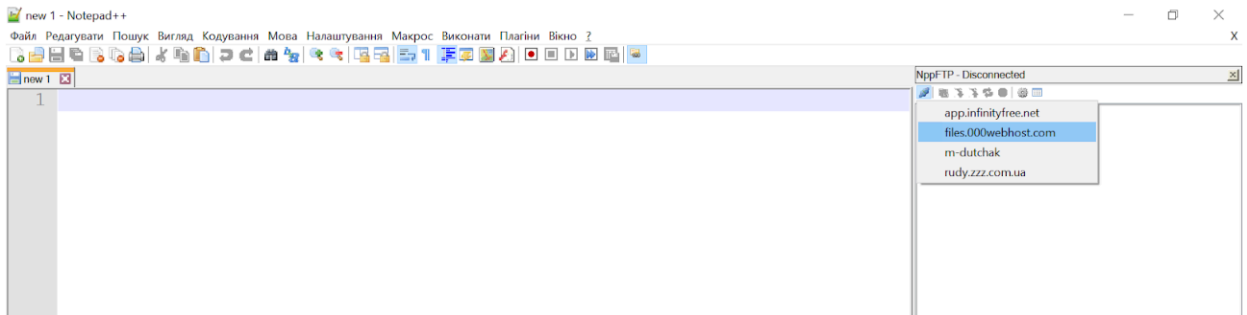
У вікні вбудованого FTP-клієнта тиснемо на "шестерю" ("Setting") і вибираємо "Profile settings" ("Налаштування профілю").



У вікні профілю заповнюємо стандартні дані для ftp-з'єднання хостингу:

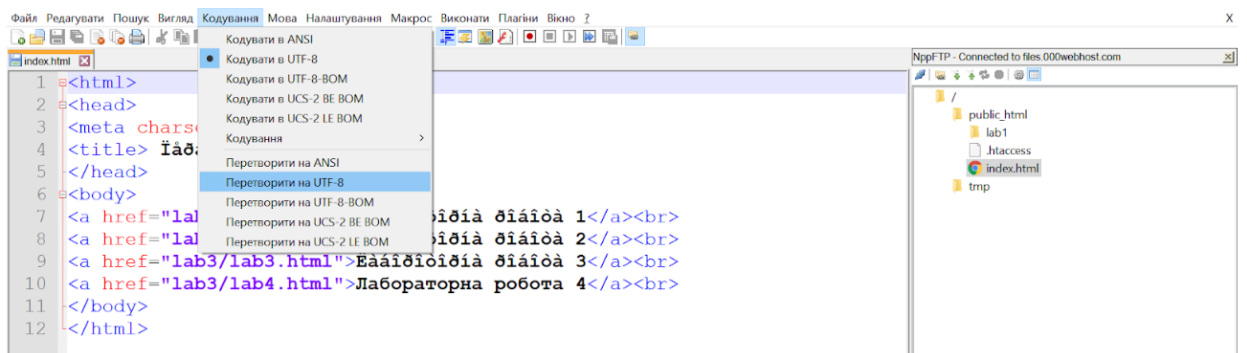
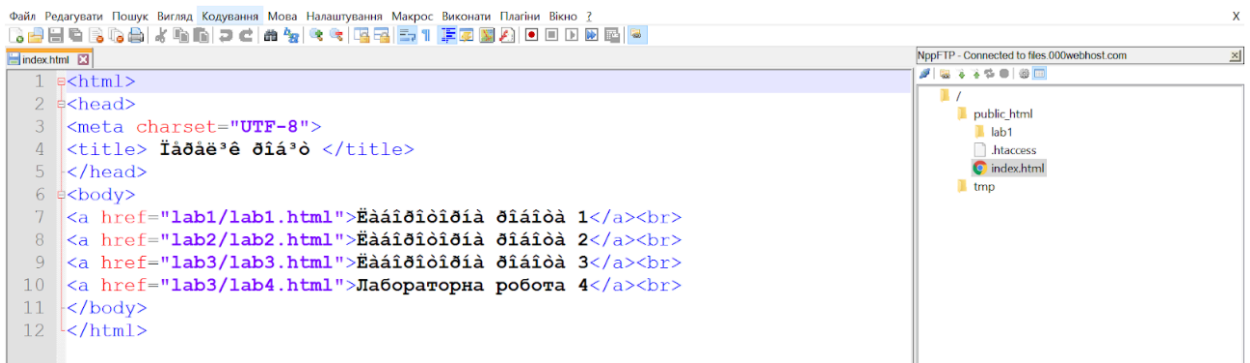


Потім обираємо створене з'єднання і переходимо на хостинг через ftp-з'єднання:



Дальше знаходимо скопійовані на вебхостинг html-файли і продовжуємо виконувати лабораторні роботи. Щоб кирилиця коректно відображалася, перед копіюванням на вебхостинг конвертуйте web-файли і задати кодування UTF-8:

Кодування-Перетворити на UTF-8.

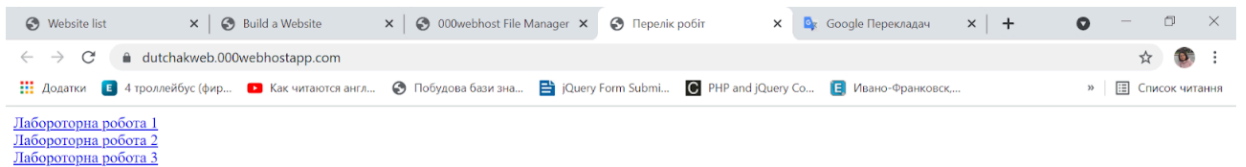


```

1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <title> Перелік робіт </title>
5 </head>
6 <body>
7 <a href="lab1/lab1.html">Лабораторна робота 1</a><br>
8 <a href="lab2/lab2.html">Лабораторна робота 2</a><br>
9 <a href="lab3/lab3.html">Лабораторна робота 3</a><br>
10 </body>
11 </html>

```

Для перегляду файлу index.html через браузер, в адресному рядку введіть Website Name, яке вказане в налаштуванні хостингу в закладці “Website Settings/General”.



Код web-документу можна переглядати викликавши контекстне меню сторінки, далі «Переглянути джерело сторінки» (Ctrl+U)

