

Міністерство освіти і науки України
Державний заклад «Південноукраїнський національний
педагогічний університет імені К.Д. Ушинського»
Фізико-математичний факультет

Брескіна Лада Валентинівна
Кобякова Людмила Миколаївна
Шувалова Ольга Ігорівна

**WEB-ПРОГРАМУВАННЯ. МОВА HTML. CSS СТИЛІ. ОПРАЦЮВАННЯ
ПОДІЙ ОБ'ЄКТІВ БРАУЗЕРА**

методичні рекомендації
для самостійної підготовки і виконання лабораторних робіт
навчального курсу ОК 15 «Інформатика»
для здобувачів першого (бакалаврського) рівня вищої освіти
1 року навчання спеціальності 014.09 Середня освіта (Математика)

Одеса 2021

Друкується за рішенням Вченої ради Державного закладу
«Південноукраїнський національний педагогічний університет
імені К.Д. Ушинського» (протокол ____ від _____ 2021 року).

Брескіна Л.В., Кобякова Л.М., Шувалова О.І.

Web-програмування. Мова HTML. CSS стилі. Опрацювання подій об'єктів браузера: методичні рекомендації для самостійної підготовки і виконання лабораторних робіт з курсу «Інформатика». Одеса. Університет Ушинського, 2021. 102с.

Рецензенти:

Максимов Максим Віталійович, доктор технічних наук, професор, завідувач кафедри комп'ютерних технологій автоматизації Одеського національного політехнічного університету.

Бойко Ольга Павлівна, кандидат фізико-математичних наук, доцент кафедри прикладної математики та інформатики Південноукраїнського національного педагогічного університету імені К. Д. Ушинського.

Методичні рекомендації «**Web-програмування. Мова HTML. CSS стилі. Опрацювання подій об'єктів браузера**» розроблено для самостійної підготовки і виконання лабораторних робіт в відповідності до програми другого змістового модуля навчальної дисципліни ОК 15 «Інформатика» для студентів 1 курсу спеціальності 014.04 Середня освіта (Математика). В рекомендаціях представлено 10 лабораторно-практичних робіт, що за програмою навчальної дисципліни розраховано на 20 годин аудиторної роботи та 50 годин самостійної роботи студентів. Перша лабораторна охоплює питання практичного застосування теоретичних основи комп'ютерних мереж через опанування питань роботи з хостингом мережі Інтернет, заведення доменного імені, формування URL адреси, роботи з сервісом Git, тощо. В лабораторних роботах 2, 3, 8 вивчаються базові об'єкт гіпертекстового документа. В лабораторних роботах 4, 5, 6, 7 представляються пояснення і завдання до опанування CSS стилів сучасного гіпертекстового документа. Правилам підключення JavaScript програм та опрацюванню подій об'єктів браузера присвячуються 9 та 10 роботи

Ключові слова: Web-програмування сторони клієнта, мова HTML, CSS стилі, мова JavaScript, комп'ютерні мережі, інформатика, хостинг, домен, URL адреса.

Зміст

Навчально-тематичний план дисципліни.....	3
Вступ.....	4
1.Лабораторна робота №1. Комп'ютерна мережа Інтернет. Служба WWW. Мова HTML. Базова структура гіпертекстового документа. Хостинг.....	6
2.Лабораторна робота №2. Мова HTML. Частина body. Базові об'єкти. Унікальна ідентифікація об'єктів.....	14
3.Лабораторна робота №3. Мова HTML. Частина body. Таблиця..	21
4.Лабораторна робота №4. CSS стилі. Правила підключення. Селектори.....	27
5.Лабораторна робота №5. CSS стилі. Стилi блоку.....	35
6.Лабораторна робота №6. CSS стилі. Стилi шрифту і тексту.....	47
7.Лабораторна робота №7. CSS стилі. Стилi налаштування кольорової гами гіпертекстового документа.....	56
8.Лабораторна робота №8. Мова HTML. Форми.....	62
9. Лабораторна робота №9. Мова JavaScript. Підключення до гіпертекстового документа. Оброблення подій об'єктів браузера.	75
10. Лабораторна робота №10. Мова JavaScript. Функції. Керування стилями.....	80
Список використаних джерел.....	89
Додаток А. Хостинг. Етапи реєстрації на zzz.com.ua.....	91
Додаток Б. Сервис github.com і його використання як хостингу...	98
Додаток В. Таблиці кодів символів гіпертекстового документа...	103

Навчально-тематичний план дисципліни

ОК 15 «Інформатика»

для здобувачів першого (бакалаврського) рівня вищої освіти

1 року навчання спеціальності 014.09 Середня освіта (Математика)

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7
Змістовий модуль 1. Соціальна інформатика. Офісні технології						
Тема 1. ІС. Апаратне забезпечення.	26	1				25
Тема 2. ІС. Програмне забезпечення.	27	1		16		10
Тема 3. Обробка графічних та відео-файлів.	38	1		12		25
Тема 4. Безпека використання сучасних інформаційних систем. Адміністрування власного інформаційного простору.	11	1				10
Разом за змістовим модулем 1.	102	4		28		70
Змістовий модуль 2. Соціальна інформатика. Робота з групою в мережі						
Тема 5. Хмарні сервіси. SaaS.	8	2		4		2
Тема 6. WEB-сервер.	16	2		4		10
Тема 7. Вступ до WEB-програмування боку клієнту. HTML.	30	2		8		20
Тема 8. Вступ до WEB-програмування боку клієнту. CSS.	24	2		4		18
Разом за змістовим модулем 2.	78	8		20		50
Усього годин	180	12		48		120

Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
Змістовий модуль 2. Соціальна інформатика. Робота з групою в мережі		
8	Публікація матеріалів у власному блозі та у соціальних мережах.	4
9	Хостінг. Використання GitHub в якості хостінгу. Публікація власних розробок.	4
10	Мова HTML. Структура документу. Основні теги.	4
11	Робота з графікою за допомогою HTML.	4
12	CSS-стилі. Підключення. Синтаксис. Властивості.	4
Усього в змістовому модулі 2.		20

Вступ

Навчальна дисципліна ОК 15 «Інформатика» для студентів 1 курсу спеціальності 014.04 Середня освіта (Математика) за навчальним планом підготовки фахівця освітнього рівня «бакалавр» охоплює 6 кредити загальною кількістю 180 годин. Аудиторні заняття - 60 годин (лекції — 12 годин, лабораторно-практичні роботи — 48 години), 120 годин відведено на самостійну роботу студентів. Зміст навчальної дисципліни розділено за програмою на два змістових модуля, перший з яких охоплює соціальну інформатику і офісні технології, а другий присвячено питанням мережевих технологій та Web-програмуванню сторони клієнта.

Передумови для вивчення дисципліни: для опанування курсу «Інформатика» в першому семестрі навчання студентом достатньо володіти компетентностями, здобутими у шкільному курсі інформатики.

Метою курсу є удосконалення компетентностей студентів в галузі соціальної інформатики:

- 1) **інтегральної компетентності** - здатності розв'язувати задачі та практичні проблеми, пов'язані із збереженням, передаванням та обробкою даних; представляти результати власної діяльності, що передбачає застосування сучасних інформаційних технологій, зокрема хмарних технологій;
- 2) **загальних компетентностей** - здатності до аналізу та синтезу при обробці даних, до абстрактного мислення; до роботи в команді та автономно; здатності застосовувати набуті знання в практичних ситуаціях; здатності вільно спілкуватися державною мовою (усно та письмово) і до професійного самовдосконалення за допомогою сучасних інформаційних технологій;
- 3) **фахових компетентностей** - здатності до вибору апаратного забезпечення для своєї професійної діяльності, використання різних операційних систем та відповідних офісних пакетів; спроможність обробляти аудіо та відео файли; представляти власні інформаційні матеріали у мережі з різними рівнями доступу; застосовувати сучасні інформаційні технології для вирішення дидактичних задач.

Очікувані результати навчання дисципліни є формування:

знань теорії інформаційних систем та пов'язаної з інформаційними технологіями термінологією; основних підходів щодо формування та редагування текстів, електронних таблиць, презентацій, графічних та відео-файлі і представлення їх у глобальній мережі за допомогою мережевих технологій;

вмінь створювати власні блоги та статичні сайти і публікувати на них підготовлені тексти, електронні таблиці, презентації та інфографіку; використовувати сервіси глобальної мережі Internet для реалізації своїх професійних компетентностей.

Методичні рекомендації «Web-програмування. Мова HTML. CSS стилі. Опрацювання подій об'єктів браузера» розроблено в відповідності до програми лабораторно-практичних робіт другого змістового модуля навчальної дисципліни ОК 15 «Інформатика» для студентів 1 курсу спеціальності 014.04 Середня освіта (Математика). В рекомендаціях представлено 10 лабораторно-практичних робіт, що за програмою навчальної дисципліни розраховано на 20 годин аудиторної роботи та 50 годин самостійної роботи студентів.

Зміст лабораторно-практичних робіт спирається на базові знання зі шкільної програми та формує підґрунтя для подальшого опанування мережевих технологій. Перша лабораторна охоплює питання практичного застосування теоретичних основи комп'ютерних мереж через опанування питань роботи з хостингом мережі Інтернет, заведення доменного імені, формування URL адреси, роботи з сервісом Git, тощо. В лабораторних роботах 2, 3, 8 вивчаються базові об'єкт гіпертекстового документа. В лабораторних роботах 4, 5, 6, 7 представляються пояснення і завдання до опанування CSS стилів сучасного гіпертекстового документа. Правилам підключення JavaScript програм та опрацюванню подій об'єктів браузера присвячуються 9 та 10 роботи.

Лабораторна робота №1

Комп'ютерна мережа Інтернет. Служба WWW. Мова HTML. Базова структура гіпертекстового документа. Хостинг.

Мета. Формування предметних компетентностей щодо створення базової структури гіпертекстового документа. Програмне забезпечення. Хостинг. Базові поняття комп'ютерних мереж.

Хід роботи:

1. Ознайомитись з поняттями:
 - a. Стек протоколів TCP/IP (Таблиця 1)
 - b. IP-протокол, IP-адреса, технологія клієнт-сервер
 - c. Служба мережі Інтернет WWW (Таблиця 2)
 - d. Доменне ім'я, URL адреса
 - e. Мова HTML. Редактори
 - f. Мова HTML. Базова структура гіпертекстового документа
 - g. Термінологія об'єктно-орієнтованого програмування до вивчення мови HTML
 - h. Ієрархія об'єктів браузера
2. Виконати Задачу 1.1
3. Виконати Задачу 1.2

Теоретичний матеріал.

Служба Word Wide Web.

Взаємодія між двома вузлами мережі Інтернет підпорядкована набору правил. Стек протоколів TCP/IP описує ці правила. Виділяють базових 4 рівня роботи стеку протоколів TCP/IP, що відповідає моделі взаємодії відкритих систем (Таблиця 1) [1, С. 364]

Таблиця 1. Рівні роботи стеку протоколів TCP/IP

Стек протоколів TCP/IP	Модель взаємодії відкритих систем OSI														
<table><tr><td>I. Прикладний рівень</td></tr><tr><td>(http/https, ftp, smtp/pop3)</td></tr><tr><td>II. Транспортний рівень</td></tr><tr><td>(TCP, UDP)</td></tr><tr><td>III. Рівень міжмережевої взаємодії IP</td></tr><tr><td>IV. Рівень мережевих інтерфейсів</td></tr><tr><td>(ЛОКАЛЬНІ МЕРЕЖІ)</td></tr></table>	I. Прикладний рівень	(http/https, ftp, smtp/pop3)	II. Транспортний рівень	(TCP, UDP)	III. Рівень міжмережевої взаємодії IP	IV. Рівень мережевих інтерфейсів	(ЛОКАЛЬНІ МЕРЕЖІ)	<table><tr><td>7. Прикладний рівень</td></tr><tr><td>6. Представницький рівень</td></tr><tr><td>5. Сеансовий рівень</td></tr><tr><td>4. Транспортний рівень</td></tr><tr><td>3.Мережевий рівень</td></tr><tr><td>2.Канальний рівень</td></tr><tr><td>1.Фізичний рівень</td></tr></table>	7. Прикладний рівень	6. Представницький рівень	5. Сеансовий рівень	4. Транспортний рівень	3.Мережевий рівень	2.Канальний рівень	1.Фізичний рівень
I. Прикладний рівень															
(http/https, ftp, smtp/pop3)															
II. Транспортний рівень															
(TCP, UDP)															
III. Рівень міжмережевої взаємодії IP															
IV. Рівень мережевих інтерфейсів															
(ЛОКАЛЬНІ МЕРЕЖІ)															
7. Прикладний рівень															
6. Представницький рівень															
5. Сеансовий рівень															
4. Транспортний рівень															
3.Мережевий рівень															
2.Канальний рівень															
1.Фізичний рівень															

IP протокол - це Інтернет протокол, що забезпечує доставку пакетів між IP - адресами вузлів. **IP - адреса** однозначно ідентифікує кожне з'єднання в мережі Інтернет. IP - адреса - це 4-байтне число (або 6-байтне число, це ми не розглядаємо). Приклад: 127.191.26.15 (255.255.255.255) - максимально допустиме значення адреси.

В мережі Інтернет реалізується технологія **клієнт - сервер**, що позначає те, що зв'язок між двома клієнтами відбувається через сервер.

Вузол - сервер відрізняється від **вузла-клієнта** в основному тільки програмним забезпеченням.

Служба мережі Інтернет WWW (Таблиця 2)

Таблиця 2. Служба World Wide Web

Програмне забезпечення серверу	Програмне забезпечення клієнта	Протоколи	Адресація
Хостинг (дисковий простір серверу)	Браузер (Opera, Mozilla)	https http (прикладни	IP адреса статична (постійна) пов'язана з хостингом (завжди!)

на якому розміщуються файли сайту) Web-сервер - програмне забезпечення сторони серверу необхідне для обробки запитів до файлів сайту.	FireFox, Google Chrome, Microsoft Edge та інші)	й рівень) TCP (транспортний рівень)	Служба DNS поєднує IP адресу хостингу з URL адресою сайту URL адреса незмінна для сайту (зазвичай) IP адреса змінюється при перенесенні сайту з одного хостингу на інший З URL-адреси відкривається index.html файл на хостингу (зазвичай поширення файла може мінятися, ім'я index)
---	---	--	--

Доменне ім'я реєструється в службі DNS імен. Має чітку структуру:

- домен 1 рівня (може мати однорівневу **.com, .ua, .org, .net**; або дворівневу структуру - **.com.ua, .edu.ua, .od.ua**), що вибирається з існуючих.
- домен 2 рівня (що унікально ідентифікує ресурс в домені 1 рівня).
Приклади: **google.com, pdpu.edu.ua**
- субдомен, або вкладені домени. Приклади: **atl.pdpu.edu.ua, library.pdpu.edu.ua, scienceandeducation.edu.ua**

URL адреса - це доменне ім'я, що пов'язане з конкретним протоколом прикладного рівня. Приклад: **https://pdpu.edu.ua** альтернативне ім'я **www.pdpu.edu.ua** (URL адреса сайту), **ftp://mirwebsistem.com** (URL адреса ftp сервера), **@gmail.com, @pdpu.edu.ua** (URL адреса поштового сервера)

Мова HTML - мова розмітки гіпертекстового документа. Гіпертекстовий документ - це текст, що структуровано на основі об'єктів HTML. Об'єкти створюються тегами в довільному простому текстовому редакторі, інтерпретуються браузером, як візуальні компоненти.

Для створення HTML-файлу можна використати найпримітивніший текстовий редактор. Наприклад, Блокнот (NotePad) для ОС Windows. Бажано, щоб у середовищі редактора здійснювалось виділення кольором тегів, атрибутів і їх значень. Наприклад, у середовищі редактора gedit для ОС Linux працює таке виділення. Рекомендуємо встановити простий редактор AkelPad для ОС Windows, який забезпечить кольорові виділення тегів, атрибутів і їх значень в

гіпертекстовому документі. Є і спеціалізовані редактори для створення гіпертекстового документа, що працюють у форматі WYSIWYG (What You See Is What You Get, «що бачиш, то і отримаєш»). Наприклад, редактори Kompozer чи Sublime Text. Але для вивчення безпосередньо мови HTML нам необхідний простий редактор. Результуючий вигляд наших документів будемо переглядати за допомогою браузера.

Підсумково ми працюємо в редакторах:

ОС Windows - **Блокнот** (NotePad) | **AkelPad**. ОС Linux — **gedit**

Переглядаємо результати в **Браузері** (Google Ghrom, Opera, Mozilla FireFox....)

Базова структура гіпертекстового документа.

Розглядаючи базову структуру гіпертекстового документа слід зазначити наступне:

1. `<!doctype html>` - це обов'язковий перший рядок для кожного гіпертекстового документа, що створено на основі HTML 5.0
2. Бінарний (складається з 2-х тегів перший-відкриваючий, другий - закриваючий) тег `<html></html>` починає і закінчує гіпертекстовий документ.
3. Між тегами `<html></html>` розташовуються дві конструкції:
 - 1 - голова (`<head></head>`);
 - 2 - тіло (`<body></body>`)
4. В частині `<head></head>` розміщуються керуючі конструкції гіпертекстового документа. Перший документ ми будемо створювати без підключення CSS стилів і без підключення JavaScript програми. Обов'язковими тегами частини head є:
 - 1 - заголовок (title, що є підписом закладки в браузері)
 - 2 - метаінструкція, що вказує на кодування документу (зазвичай в гіпертекстовому документі використовують кодування UTF-8, що дає можливість відобразити на гіпертекстовій сторінці текст практично будь-якої мови, слід пам'ятати, що в цьому кодуванні для кожного символу відводиться 4 байта пам'яті (рис. 1.1)

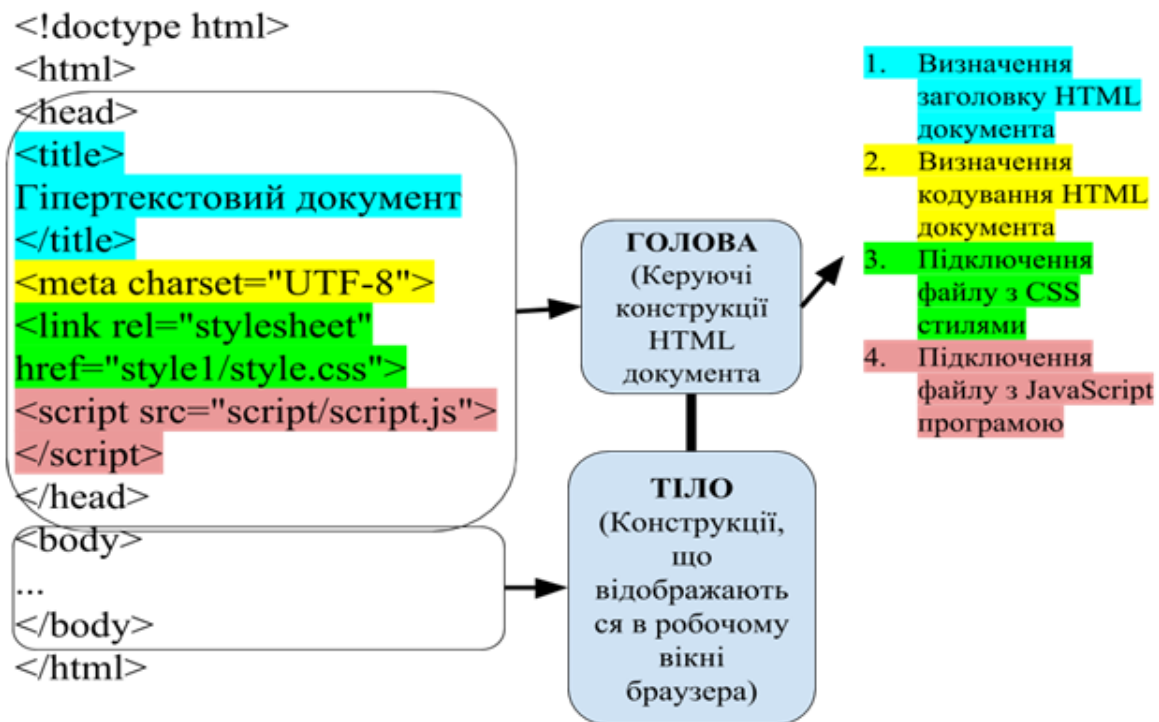


Рис.1.1. Базова структура гіпертекстового документа

В частині `body` гіпертекстового документа розташовуються данні, що відображаються в робочому вікні браузера. Тегами HTML частини `body` створюються екземпляри об'єктів браузера.

Термінологія об'єктно-орієнтованого програмування до вивчення мови HTML.

В основі об'єктно-орієнтованої мови програмування лежать два основних поняття: об'єкт (клас) та екземпляр об'єкта (об'єкт). Екземпляр об'єкта – це базове поняття в ООП, це конкретна реалізація. Основні властивості екземпляр об'єктаів: Інкапсуляція це здатність екземплярів об'єктаів скривати деякі способи обробки даних (методи) та самі дані від навколишнього цифрового середовища. Поліморфізм - це здатність екземпляром об'єкта використовувати методи батьківського класу. Слово «поліморфізм» можна перекласти як «багато форм». Тобто це можливість використання одного і того ж імені методу до екземплярів об'єктів різних об'єктів (класів). Спадкування - стосується здатності мови дозволяти будувати нові об'єкти (класи) на основі значень існуючих об'єктів (класів). По суті, спадкування дозволяє розширювати поведінку базового об'єкту (класу), наслідуючи основну функціональність батьківського об'єкта

(класу) у похідному об'єкті (класі).

Ієрархія об'єктів браузера та правила однозначної ідентифікації об'єкта document і вкладених об'єктів гіпертекстового документу у мовах HTML, CSS, JavaScript (Рис.1.2)

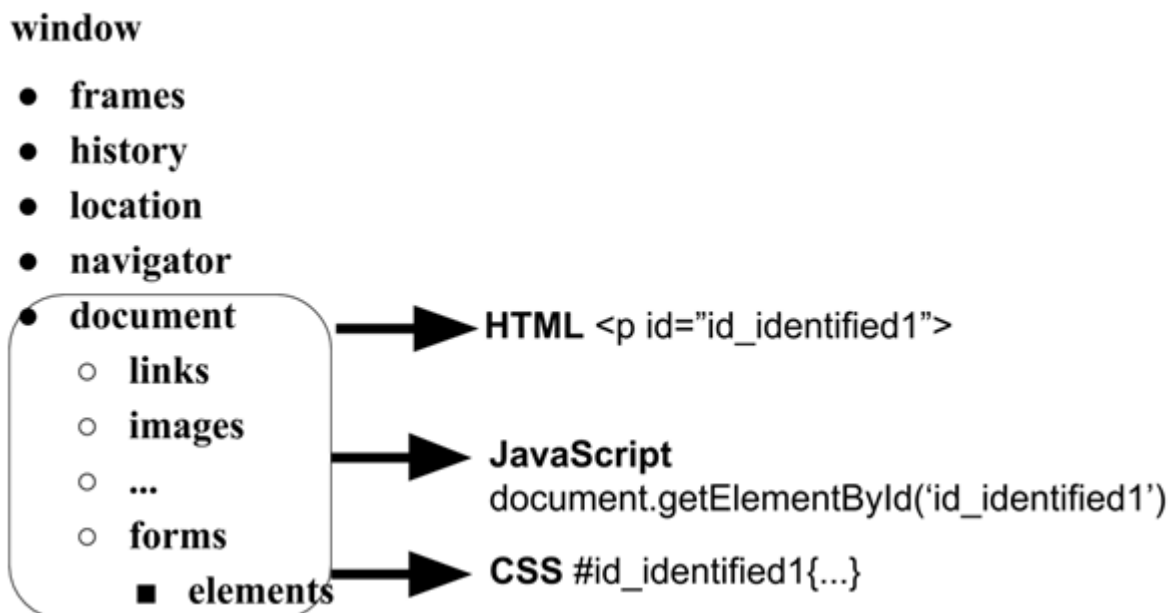


Рис.1.2. Ієрархія об'єктів браузера та однозначна їх ідентифікація в мовах HTML, CSS, JavaScript

Робота з хостингом мережі Інтернет

Задача 1.1 (професійний хостинг). Розглянемо приклад роботи хостингу zzz.com.ua (безкоштовна реєстрація терміном на 5 днів, річна вартість хостингу 100 грн (дата даних 12.05.2021)). Як альтернативою можна скористатись безкоштовними послугами хостингу на ho.ua на пів року (дата даних 14.06.2021)

- 1-Зареєструватись, завести безкоштовне доменне ім'я при реєстрації
- 2-Завести ftp клієнта для віддаленої роботи з файлами
- 3-Увійти в FTP-клієнт і створити файл index.html
- 4-Розглянути альтернативний варіант розміщення гіпертекстового документа на хостингу.

Розв'язок:

- 1 - для проходження безкоштовної реєстрації має бути зареєстровано поштову скриньку. При реєстрації на першому етапі вибирається доменне ім'я (рис.А.1).

Наступний крок - це заповнення реєстраційної форми, отримати код підтвердження на вашу e-mail адресу, введення коду і отримання безкоштовного хостингу на 7 днів, для подальшої роботи на рік слід зробити оплату (рис.А.1-А.5)

2- робота з Панеллю управління хостингом на поточному етапі передбачає вміння зареєструвати домени і субдомени (рис.А.6) та заводити клієнта FTP (рис.А.7)

3 - Входимо в FTP клієнт хостинга (рис.8), відкриваємо папку з ім'ям нашого домену (рис.А.9), переходимо в режим створення першого файлу (рис.А.10), створюємо файл index.html (рис.А.11)

4 - Альтернативним способом роботи з хостингом є створення файлу index.html на локальному комп'ютері в редакторі Блокнот (рис.1.3) і подальше його завантаження на хостинг (рис.А.12)

Тепер з URL адреси (<http://+домен>) ми бачимо наш простенький сайт (рис.А.13) (рекламу видно через неоплачений хостинг)

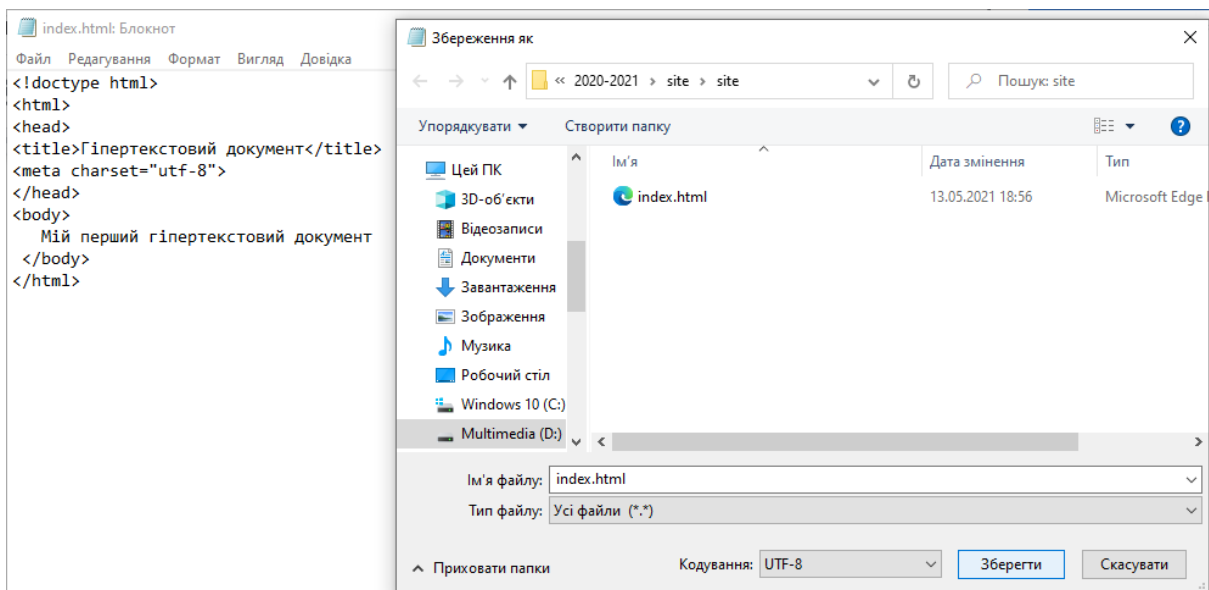


Рис. 1.3. Створення гіпертекстового документа index.html в блокноті

Задача 1.2 (безкоштовний хостинг з обмеженими можливостями)

1-Зареєструватись на сервісі github.com.

2-Створити репозиторій.

3-Налаштувати доступ з URL адреси.

4-Створити перший гіпертекстовий документ index.html

Розв’язок:

- 1 - Хід реєстрації стандартний з двох етапів: заповнення реєстраційної форми і підтвердження реєстрації з електронної пошти. Входимо в сервіс.
- 2 - Переходимо в розділ “New repository” (рис.Б.1). Створюємо ім’я репозиторія латинськими літерами, ставимо галочку для створення readme файла (рис.Б.2).
- 3 - Після успішного створення репозиторію потрапляємо в розділ Code для створення нових файлів в репозиторію (рис.Б.3). Для налаштування URL адреси для доступу до файлів репозиторію необхідно перейти в розділ Settings (рис.Б.4) і налаштувати GitHub Pages (рис.Б.5). Після отримання URL адреси (рис.Б.6) переходимо в розділ Code і завантажуюмо index.html файл з локального комп’ютера (рис.Б.7). Обов’язково підтверджуємо завантаження “Commit Changes” (рис.Б.8). Переглядаємо результуючу сторінку з URL адреси (рис.Б.9).

Лабораторна робота №2

Мова HTML. Частина body. Базові об'єкти. Унікальна ідентифікація об'єктів.

Мета. Формування предметних компетентностей по роботі з візуальними об'єктами гіпертекстового документа.

Хід роботи:

1. При ознайомленні з теоретичним матеріалом покроково створити гіпертекстову сторінку на основі прикладів - задач від задачі 2.1 до задачі 2.1.5
 - a. Вивчаємо об'єкт блок `<div> </div>` (Задача 2.1)
 - b. Вивчаємо об'єкт заголовок `<h1></h1>` (Задача 2.1.1)
 - c. Вивчаємо об'єкт image `` (Задача 2.1.2)
 - d. Вивчаємо об'єкт гіперпосилання `<a href="https://www.google.com/">Пошук Гугл</a>` (Задача 2.1.3)
 - e. Вивчаємо об'єкт списки `<ol></ol>` - нумерований список, `<ul></ul>` - нелінійний список (Задача 2.1.4)
 - f. Вивчаємо об'єкт абзац (параграф) `<p></p>` (Задача 2.1.5)
2. Виконати індивідуальне завдання

Теоретичний матеріал.

Вивчаємо об'єкт блок `<div> </div>` (в браузері не відображається, об'єднує будь-які об'єкти, що вкладено в конкретний блок)

- Правило унікальної ідентифікації блоків `<div id="container"> </div>`
- Правило ієрархічної вкладеності блоків

```
<div id="container1">
```

```
<div id="container2">
```

```
<div id="container3"> </div>
```

```
</div>
```

```
</div>
```

Задача 2.1. Побудуємо гіпертекстову сторінку, що містить 1 зовнішній блок з унікальним іменем container. В цей блок вкладемо 4 блоки з унікальними іменами: 1 - header; 2 - menu; 3 - content; 4 – footer.

Розв'язок - (Таблиця 2.1).

Таблиця 2.1. Розв'язок задачі 2.1

<pre><!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/style.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html></pre>	<pre><!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/style.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html></pre>
---	---

Вивчаємо об'єкт заголовок <h1></h1> гіпертекстового документа.

- В гіпертекстовому документі визначаються заголовки 6 рівнів від <h1> - найбільший до <h6> - найменший
- Правило унікальної ідентифікації заголовків <h1 id="zag1"> </h1>

Задача 2.1.1. Вставити в блок header заголовок першого рівня зі слоганом сайту.

Розв'язок - (Таблиця 2.1.1).

Таблиця 2.1.1 Розв'язок задачі 2.1.1

<pre><div id="header"> <h1>Слоган сайту</h1> </div></pre>	<pre><div id="header"> <h1>Слоган сайту</h1> </div></pre>
---	---

Вивчаємо об'єкт image гіпертекстового документа .

Задача 2.1.2. Вставити в блок header перед заголовком першого рівня логотип сайту (зображення).

Розв'язок:

1 етап - знайти зображення

2 етап - розмістити (зберегти) його в тій самій папці, в котрій зберігається файл index.html (рис. 2.1)

3 етап - скопіювати повне ім'я файла, а саме ім'я і поширення. (Приклади: pic1.png, pic1.jpg, pic1.jpeg)

ltimedia (D:) > 2019-2020 > Shuvalova > 2020-2021 > гіпертекстовий документ			
Ім'я	Дата змінення	Тип	Розмір
 images123.png	04.03.2021 13:40	Файл PNG	9 КБ
 index.html	04.03.2021 13:46	Chrome HTML Do...	1 КБ

Рис. 2.1 Приклад збереженого в папці файла

- Тег `` створює об'єкт image.
- В атрибуті src вказуємо путь до файла з картинкою.
- В атрибуті alt описуємо текстово графічний об'єкт.
- Для визначення унікального ідентифікатора цього об'єкта необхідно додати значення атрибута id. Приклад:

``

Розв'язок - (Таблиця 2.1.2).

Таблиця 2.1.2 Розв'язок задачі 2.1.2

<code><div id="header"> <h1>Слоган сайту</h1> </div></code>	<code><div id="header"> <h1>Слоган сайту</h1> </div></code>
---	---

Вивчаємо об'єкт гіперпосилання гіпертекстового документа.

Задача 2.1.3. Вставити в блок меню три гіперпосилання:

№	підпис гіперпосилання	URL адреси переходу за гіперпосиланням
1.	Пошук Гугл	https://www.google.com/
2.	Університет Ушинського	https://pdpu.edu.ua/

3.	Бібліотека Університету Ушинського	https://library.pdpu.edu.ua/
----	------------------------------------	---

Розв'язок – (таблиця 2.1.3)

- Тег `<a href="https://www.google.com/">Пошук Гугл</a>` створює об'єкт гіперпосилання №1 з таблиці-завдання.
 - В атрибуті href вказуємо URL адреси переходу за гіперпосиланням.
 - Підпис гіперпосилання розташувати треба між відкриваючим і закриваючим тегом "a"
- `<a ...>` `</a>`.
- Для визначення унікального ідентифікатора цього об'єкта необхідно додати значення атрибута id. Приклад:

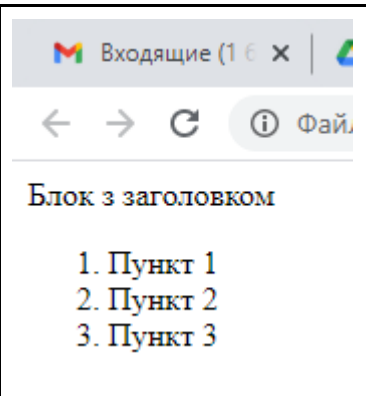
`<a href="https://www.google.com/" id="link1">Пошук Гугл</a>`

Таблиця 2.1.3 Розв'язок задачі 2.1.3

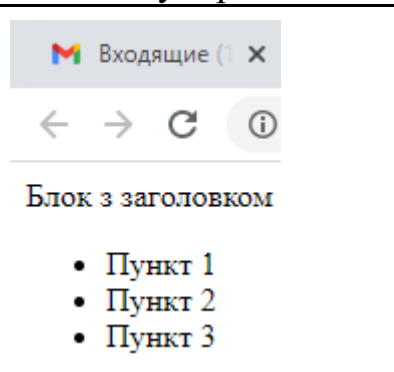
<pre><div id="menu"> Пошук Гугл Університет Ушинського Бібліотека Університету Ушинського </div></pre>
<pre><div id="menu"> Пошук Гугл Університет Ушинського Бібліотека Університету Ушинського </div></pre>
Слоган сайта <u>Пошук Гугл Університет Ушинського Бібліотека Університету Ушинського</u> Блок контент Блок футер

Структуруємо дані гіпертекстової сторінки з використанням списків `<ol></ol>` - нумерований список (таблиця 2.1.4.1), `<ul></ul>` - нелінійний список (таблиця 2.1.4.2), вкладений (ієрархічний список) (таблиця 2.1.4.3).

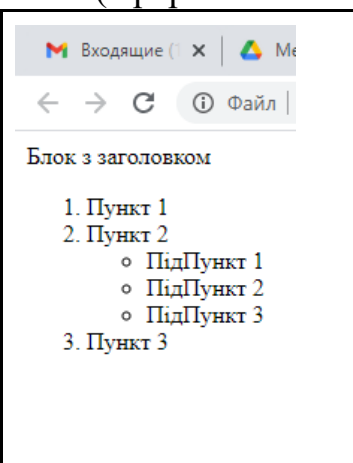
Таблиця 2.1.4.1 Приклад нумерованого списку

<pre> Пункт 1 Пункт 2 Пункт 3 </pre>	<pre> Пункт 1 Пункт 2 Пункт 3 </pre>	 Входящие (1 x ...) ← → ↻ ⓘ Файл Блок з заголовком 1. Пункт 1 2. Пункт 2 3. Пункт 3
--	--	---

Таблиця 2.1.4.2 Приклад нелінійного списку

<pre> Пункт 1 Пункт 2 Пункт 3 </pre>	<pre> Пункт 1 Пункт 2 Пункт 3 </pre>	 Входящие (1 x ...) ← → ↻ ⓘ Блок з заголовком • Пункт 1 • Пункт 2 • Пункт 3
--	--	--

Таблиця 2.1.4.3 Приклад вкладеного (ієрархічного) списку

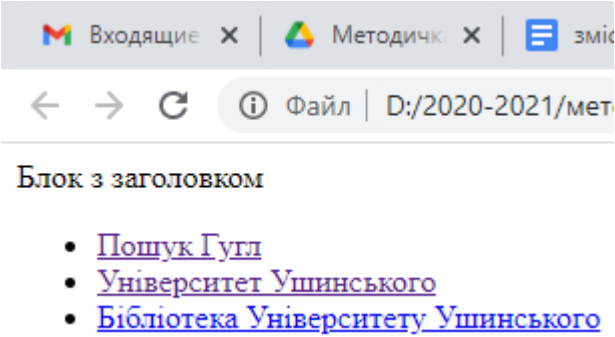
<pre> Пункт 1 Пункт 2 ПідПункт 1 ПідПункт 2 ПідПункт 3 Пункт 3 </pre>	<pre> Пункт 1 Пункт 2 ПідПункт 1 ПідПункт 2 ПідПункт 3 Пункт 3 </pre>	 Входящие (1 x ...) ← → ↻ ⓘ Файл Блок з заголовком 1. Пункт 1 2. Пункт 2 ◦ ПідПункт 1 ◦ ПідПункт 2 ◦ ПідПункт 3 3. Пункт 3
---	---	---

Задача 2.1.4. Структурувати гіперпосилання задачі 2.1.3 з використанням списку.

Розв’язок (таблиця 2.1.4).

Таблиця 2.1.4. Розв'язок задачі 2.1.4

<pre> <div id="menu"> Пошук Гугл Університет Ушинського Бібліотека Університету Ушинського </div> </pre>



- Для визначення унікального ідентифікатора цього об'єкта необхідно додати значення атрибута id. Приклад:

```
<ul id="sp1">...</ul>
```

Структуруємо дані гіпертекстової сторінки з використанням абзаців (параграфів) <p></p>

Задача 2.1.5. Вставити в блок content три абзаца з довільним текстом

Розв'язок: (таблиця 2.1.5).

Таблиця 2.1.5. Розв'язок задачі 2.1.5

<pre><div id="content"> <p>Параграф 1</p> <p>Параграф 2</p> <p>Параграф 3</p> </div></pre>	<pre><div id="content"> <p>Параграф 1</p> <p>Параграф 2</p> <p>Параграф 3</p> </div></pre>	Параграф 1 Параграф 2 Параграф 3
--	--	---

- Для визначення унікального ідентифікатора цього об'єкта необхідно додати значення атрибута id. Приклад:

```
<p id="pg1">...</p>
```

Індивідуальне завдання.

1. Створити сторінку, що містить 6 блоків
2. Перший блок зовнішній, включає в себе 5 вкладених блоків.
3. Другий блок включає логотип сайту (рисунок `<img ...>`), назву сайту та слоган сайту (заголовки `h1`, `h3`) (Придумати індивідуальний слоган. В якості назви використати власне прізвище, ім'я, по батькові)
4. Третій блок включає список (нумерований чи нелінійний) з 3-х гіперпосилань на довільні ресурси мережі Інтернет, якими ви користуєтесь.
5. Четвертий блок містить ваші інтереси, що структуруються з використанням абзаців (щонайменше 3 абзаци).

Лабораторна робота №3.

Мова HTML. Частина body. Таблиця. Амперсанд послідовності

Мета. Формування предметних компетентностей по роботі з візуальними об'єктами гіпертекстового документа.

Хід роботи:

1. При ознайомленні з теоретичним матеріалом доопрацювати гіпертекстову сторінку попередньої (другої) лабораторної роботи на основі прикладів - задач від задачі 3.1 до задачі 3.2
 - а. Вивчаємо об'єкт таблиця `<table></table>` (Задача 3.1)
 - б. Амперсанд послідовності гіпертекстової сторінки. Об'єкт `<span></span>` (Задача 3.2)
2. Виконати індивідуальне завдання

Теоретичний матеріал.

Структуруємо дані гіпертекстової сторінки з використанням таблиць

`<table></table>`

`<table border="1"></table>` створюємо таблицю, скористаємось атрибутом `border="1"` для промальовування границь таблиці

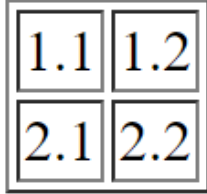
`<tr></tr>` створюємо рядок в таблиці. Рядки вкладаються в таблицю, їх кількість необмежена `<table border="1"><tr></tr></table>`

`<td></td>` створюємо комірку в рядку таблиці. Кількість комірок в одному рядку довільна, але ВСІ РЯДКИ ТАБЛИЦІ МАЮТЬ МІСТИТИ ОДНАКОВУ КІЛЬКІСТЬ КОМІРОК.

`<table border="1"><tr><td></td></tr></table>`

Приклад таблиці 2x2 (два рядка по дві комірки, в якості тексту вставимо нумерацію комірок) (таблиця 3.1.1)

Таблиця 3.1.1 Приклад таблиці 2x2

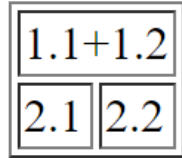
<pre><table border="1"> <tr> <td>1.1</td> <td>1.2</td> </tr> <tr> <td>2.1</td> <td>2.2</td> </tr> </table></pre>	<pre><table border="1"> <tr> <td>1.1</td> <td>1.2</td> </tr> <tr> <td>2.1</td> <td>2.2</td> </tr> </table></pre>	
--	--	---

Для об'єднання комірок в одному рядку таблиці використовується атрибут

colspan

Приклад таблиці з двома рядками, але в першому рядку 1 комірка, що охоплює 2, а в 2 рядку 2 комірки (таблиця 3.1.2)

Таблиця 3.1.2 Приклад таблиці 2x2 з об'єднанням комірок в 1 рядку

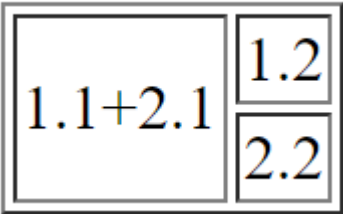
<pre><table border="1"> <tr> <td colspan="2">1.1+1.2 </td> </tr> <tr> <td>2.1</td> <td>2.2</td> </tr> </table></pre>	<pre><table border="1"> <tr> <td colspan="2">1.1+1.2 </td> </tr> <tr> <td>2.1</td> <td>2.2</td> </tr> </table></pre>	
--	--	--

Для об'єднання комірок в одному стовпчику (в сусідніх рядках) таблиці

використовується атрибут **rowspan**

Приклад таблиці з двома рядками, в першому рядку 2 комірка, де перша комірка охоплює дві комірки по вертикалі (себе і комірку з нижнього рядка), а в 2 рядку 1 комірка.

Таблиця 3.1.3 Приклад таблиці 2x2 з об'єднанням комірок в 1 стовпчику

<pre><table border="1"> <tr> <td rowspan="2">1.1+2.1 </td> <td>1.2</td> </tr> <tr> <td>2.2</td> </tr> </table></pre>	<pre><table border="1"> <tr> <td rowspan="2">1.1+2.1 </td> <td>1.2</td> </tr> <tr> <td>2.2</td> </tr> </table></pre>
	

Для об'єднання комірок в одному стовпчику (в сусідніх рядках) таблиці і в одному рядку використовується одночасно два атрибута **rowspan** і **colspan**

Приклад таблиці,

Задача 3.1. Вставити в блок content після третього абзаца таблицю, в котрій 2 рядка по 3 комірки в кожному. Перша комірка першого рядка охоплює 2 комірки по вертикалі і 2 комірки по горизонталі

Розв'язок: (таблиця 3.1)

Таблиця 3.1 Розв'язок задачі 3.1

<pre><table border="1"> <tr> <td rowspan="2" colspan="2"> 1.1+1.2+2.1+2.2 </td> <td>1.3</td> </tr> <tr> <td>2.3</td> </tr> </table></pre>	<pre><table border="1"> <tr> <td rowspan="2" colspan="2"> 1.1+1.2+2.1+2.2 </td> <td>1.3</td> </tr> <tr> <td>2.3</td> </tr> </table></pre>
---	---

1.1+1.2+2.1+2.2	1.3
	2.3

Для визначення унікального ідентифікатора цього об'єкта необхідно додати значення атрибута id. Приклад:

```
<table border="1" id="tb1">...</p>
```

Слід пам'ятати, що кожен рядок і кожна комірка в межах однієї таблиці є самостійними об'єктами, що можуть бути унікально ідентифікованими.

Приклад:

```
<table border="1">
<tr id="row1">
<td id="col1">
1.1+1.2+2.1+2.2
</td>
</tr></table>
```

Амперсанд послідовності гіпертекстової сторінки служать для додавання символів згідно їх коду. Загальний вигляд для символу захисту прав (copyright):

За десятковим номером: ©

За шістнадцятковим номером: ©

По імені: ©

Таблиці кодів слід переглянути в додатку до методичних рекомендацій (ДодатокВ, таблиця 1)

Задача 3.2. Вставити в блок footer значок copyright з підписом Університет Ушинського 2021 трьома різними способами: 1- амперсанд послідовність зі словесним кодом, 2 - амперсанд послідовність з десятковим кодом символа, 3 - амперсанд послідовність з шістнадцятковим кодом символа.

Розв'язок: (таблиця 3.2)

В цій задачі скористаємось тегом
 для переведення строки.

Таблиця 3.2 Розв’язок задачі 3.2

<pre><div id="footer"> &copy;Університет Ушинського 2021
 &#169;Університет Ушинського 2021
 &#xA9;Університет Ушинського 2021 </div></pre>	<pre>©Університет Ушинського 2021 ©Університет Ушинського 2021 ©Університет Ушинського 2021</pre>
<pre><div id="footer"> &copy;Університет Ушинського 2021
 &#169;Університет Ушинського 2021
 &#xA9;Університет Ушинського 2021 </div></pre>	

Амперсанд послідовність не є об’єктом гіпертекстового документа, а є просто символом. Для формування об’єкта з довільного символу або групи символів слід скористатись тегом `<span></span>`. Відповідно унікальну ідентифікацію об’єкта можна задати в наступний спосіб:

```
<span id="ap1">&copy;</span>
```

На сторінці тег `span` не впливає на форматування.

Індивідуальне завдання.

Доопрацювати завдання попередньої (другої) лабораторної роботи

- П’ятий блок — це таблиця з довільною інформацією (вставте нумерацію комірок як в прикладах до лабораторної роботи), що структурована згідно з вашим варіантом, що наведено на рисунку (рис. 3.1)
- Шостий блок включає амперсанд послідовність `©` та ПІБ розробника з поточним роком (рядок захисту прав).

Варіант 1	Варіант 2	Варіант 3																		
<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td colspan="2"></td></tr></table>							<table><tr><td colspan="2"></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>							<table><tr><td></td><td></td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"></td></tr></table>						
Варіант 4	Варіант 5	Варіант 6																		
<table><tr><td colspan="2"></td></tr><tr><td></td><td></td></tr><tr><td colspan="2"></td></tr></table>							<table><tr><td></td><td rowspan="3"></td></tr><tr><td></td></tr><tr><td colspan="2"></td></tr></table>						<table><tr><td rowspan="2"></td><td></td></tr><tr><td></td></tr><tr><td colspan="2"></td></tr></table>							

Рис 3.1. Індивідуальні завдання з побудови таблиць

Лабораторна робота №4

CSS стилі. Правила підключення. Селектори стилів.

Мета. Формування предметних компетентностей по підключенню стилів до об'єктів гіпертекстового документа

Хід роботи:

1. Ознайомитись з теоретичним матеріалом
2. Для закріплення базового правила підключення стилів створити гіпертекстову сторінку на основі прикладу - задачі 4.1
3. Створити документи з альтернативними способами підключення стилів (Таблиця 4.2)
4. Для закріплення правил визначення селекторів стилів програмно реалізувати приклад 4.2
5. Ознайомитись з роботою Інструмента розробника в браузері через поетапне визначення стилів (рис. 4.3)
6. Відповісти на контрольні запитання.

Теоретичний матеріал

Стилі - це властивості об'єктів гіпертекстового документа, що налаштовуються.

Об'єкти створюються за допомогою тегів.

Екземпляри об'єктів унікально ідентифікуються через визначення id.

Базовий підхід для визначення стилізації об'єктів гіпертекстового документа - це формування окремого документа з розширенням css.

Приклад 4.1 Підключити до документа index.html документ style.css, в котрому надалі будуть визначатись стилі об'єктів гіпертекстового документа.

Розв'язок:

1. В простому текстовому редакторі (Блокнот, AkelPad, gedit) створюємо документ на ім'я style.css (кодування UTF-8)
2. Розташовуємо файл style.css в одній папці з файлом index.html
3. В частині head файла index.html прописуємо команду підключення файла

зі стилями (таблиця 4.1.1)

Таблиця 4.1.1 Підключення файла зі стилями

index.html	style.css
<pre> <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html> </pre>	<pre> body{background-color: grey;} </pre>

Альтернативні методи підключення CSS стилів (Таблиця 4.1.2):

1. Визначення стилів в частині head
2. Визначення стилів безпосередньо в екземплярах об'єктів гіпертекстового документа

Таблиця 4.1.2. Альтернативні методи підключення CSS стилів

Визначення стилів в частині head	Визначення стилів в екземплярах об'єктів гіпертекстового документа
<pre> <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <style> body{background-color: grey;} </style> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> </div> </body> </html> </pre>	<pre> <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> </head> <body style="background-color: grey;"> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> </div> </body> </html> </pre>

Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html>	<div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html>
--	---

Правила визначення селекторів стилів в файлі зі стилями (style.css).

1. Стилі можна назначати для всіх екземплярів конкретного об'єкта. Це робиться через ім'я тега, що відповідає за створення об'єкта:

div, p, ul{color:red; border: 2px solid blue;} (Стилізація буде відноситись до всіх об'єктів div, p, ul гіпертекстової сторінки)

2. Стилі можна назначати конкретним екземплярам об'єктів. Це робиться через значення id цього екземпляра:

#header{color:red; border: 2px solid blue;} (Стилізація буде відноситись до одного екземпляра об'єкта <div id="header">.....</div>). Групова стилізація теж працює: #header, #content, #footer{color:blue;}

3. Для визначення однакових стилів різним екземплярам об'єктів використовуються селектори класів:

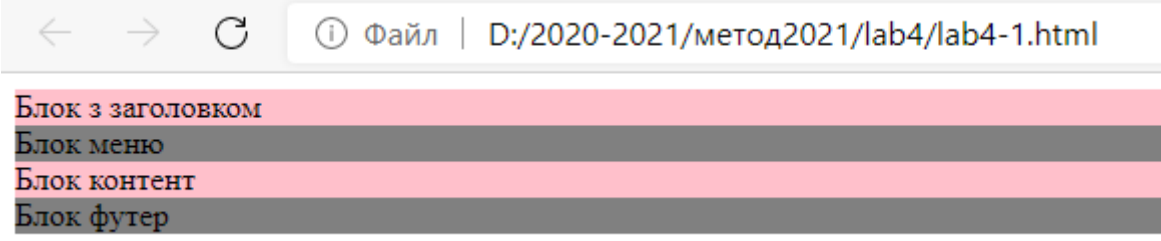
.class_style1 {color:red; border: 2px solid blue;} (Стилізація буде відноситись до всіх екземплярів об'єктів гіпертекстового документа, в котрих буде визначено атрибут class="class_style1" <div class="class_style1">.....</div> <p class="class_style1">.....</p>)

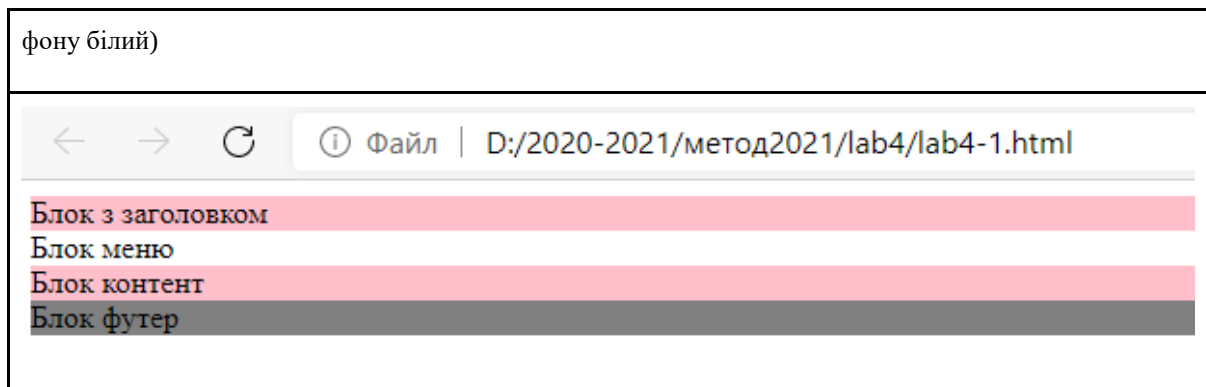
Для закріплення правил визначення селекторів стилів скористаємось стилем background-color: blue; (стиль, що визначає колір фону відповідного об'єкта)

Приклад 4.2.

1. До файла index.html підключити файл зі стилями style.css
2. Для всіх блоків визначити колір фону - сірий.
3. Налаштувати фон блоків header, content за допомогою селектору класу в рожевий колір.
4. Налаштувати фон блоку menu в білий колір

Таблиця 4.2. Розв'язок прикладу 4.2

index.html	style.css
<pre> <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style.css"> </head> <body> <div id="container"> <div id="header" class="clas_div1"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content" class="clas_div1"> Блок контент </div> <div id="footer"> Блок футер </div> </body> </html> </pre>	<pre> div{background-color: grey;} .clas_div1{background-color: pink;} #menu{background-color: white;} </pre>
Поетапне додавання стилізації:	
div{background-color: grey;} (всі об'єкти - блоки отримали колір фону сірий)	
	
div{background-color: grey;} .clas_div1{background-color: pink;} (об'єкти-блоки класу clas_div1 отримали колір фону рожевий)	
	
div{background-color: grey;} .clas_div1{background-color: pink;} #menu{background-color: white;} (екземпляр об'єкті з унікальною ідентифікацією меню отримали колір	



Пріоритети при визначенні стилів.

Розгляд поетапного додавання стилів дає нам можливість зрозуміти які з селекторів мають більший пріоритет. Якщо на один екземпляр об'єкта одночасно впливають стилі ідентифіковані селекторами - об'єкта, класа і ID, то пріоритетність від меншого до більшого розташовується в наступний спосіб:

1. Стилi об'єктiв.
2. Стилi класiв.
3. Стилi iдентифіковані ID

Змінити цей встановлений порядок пріоритетів можна з використанням конструкції `!important`;

Приклад:

```
.two { color: red !important; }
```

Для перегляду стилів екземплярів об'єктів на сторінці браузера є спеціальний інструмент - Інструмент розробника (рис. 4.1).

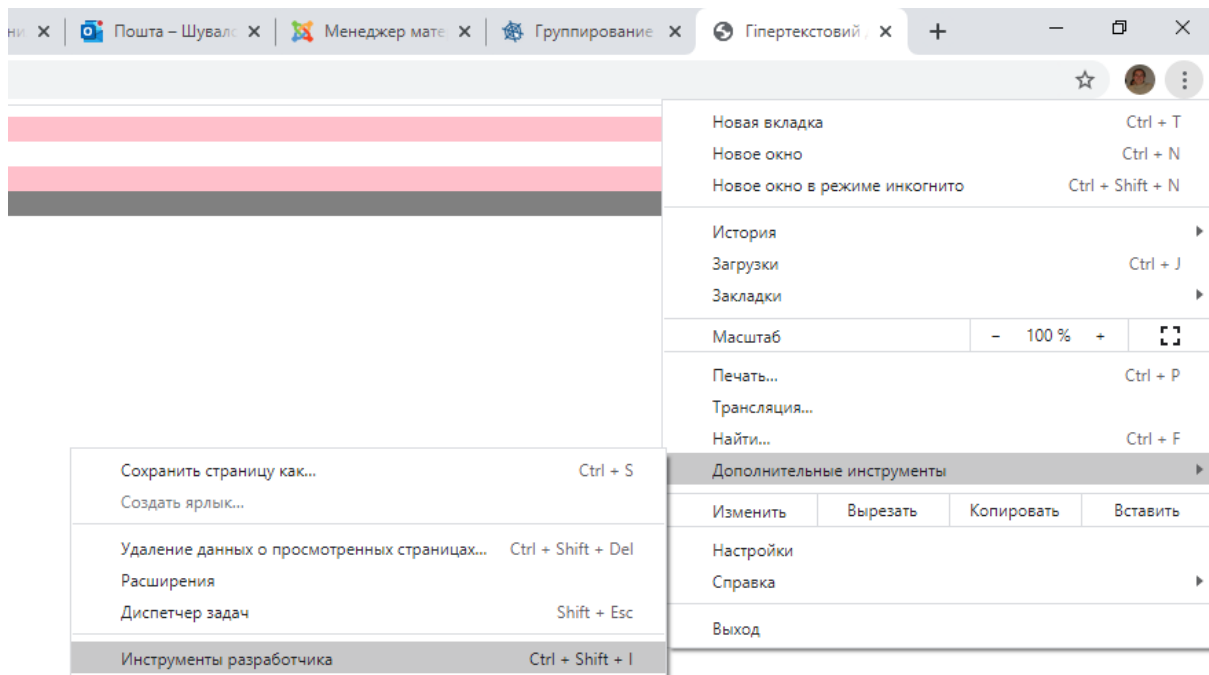
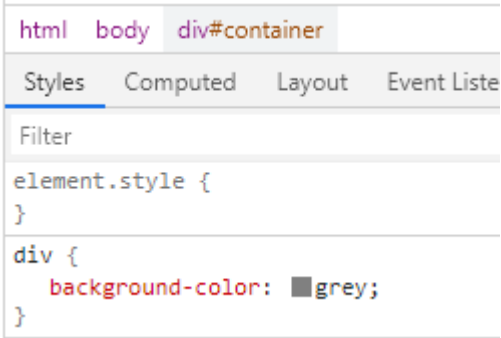
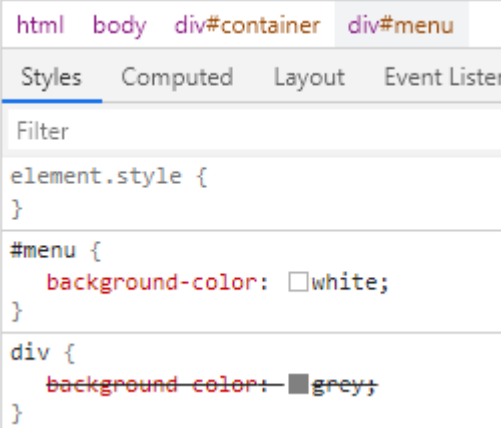
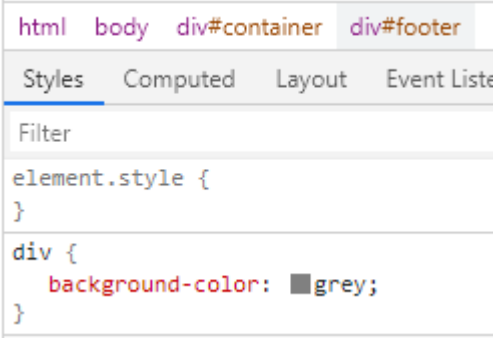


Рис.4.1. Відкриваємо інструменти розробника

Підтвердження пріоритетності стилізації в інструменті розробника. В якості приклада розглянемо таблицю з рисунками (таблиця 4.3). На рисунку 4.3.1 та 4.3.4 (рис.4.3.1, 4.3.4) визначено стилізацію всіх екземплярів об'єкта div (колір фону - сірий). На рисунку (рис.4.3.2) відмінено попередню стилізацію через назначення класу стилів, що визначає колір фону - рожевий. На рисунку (рис.4.3.3) відмінено попередню стилізацію через назначення стилів через унікальну стилізацію екземпляру об'єкта div (# - id), що визначає колір фону - білий.

Загальну пріоритетність стилів застосованих до одного екземпляра об'єкта можна навести списком. Наступні селектори мають більший пріоритет і відмінюють відповідну стилізацію за попереднім пріоритетом

Таблиця 4.3. Приклади роботи пріоритетів в стилізації

 Рис. 4.3.1. Стилї об'єкту container	 Рис. 4.3.2. Стилї об'єкта header
 Рис. 4.3.3. Стилї об'єкта menu	 Рис. 4.3.4. Стилї об'єкта footer

DOM модель гіпертекстового документа. Правила успадкування стилів

Для знайомства з DOM моделлю гіпертекстового документа слід відкрити

Інструменти розробника в браузері (рис. 4.1)

Батьківським об'єктом є екземпляр container з визначеним стилем.

Розглядаючи DOM модель гіпертекстового документа бачимо ієрархічну, вложену його структуру. В нашому прикладі в об'єкт body, вкладено екземпляр об'єкта div (container) в екземпляр об'єкта container вкладено 4 екземпляри об'єктів (header, menu, content, footer) (рис. 4.2).

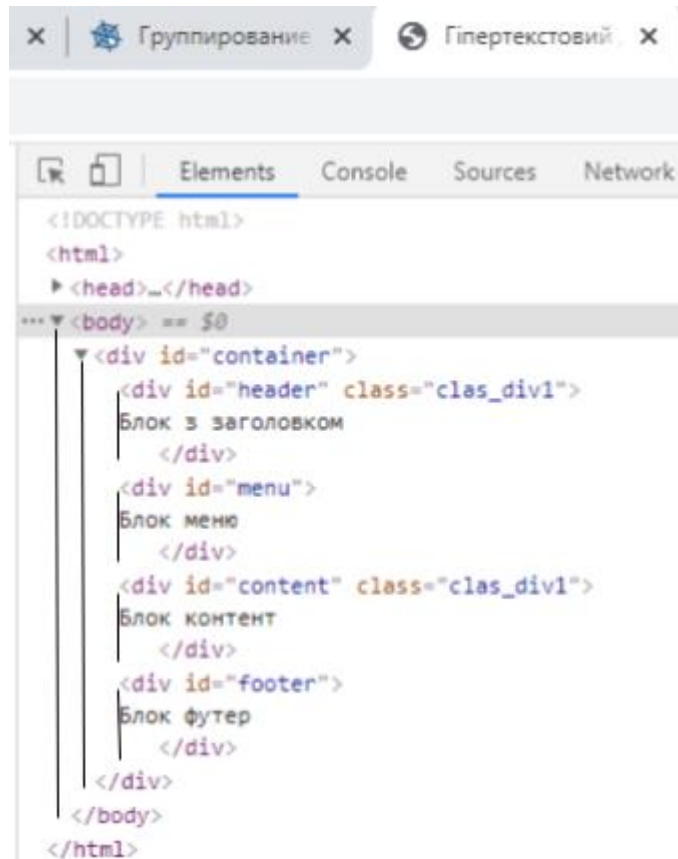


Рис. 4.2. DOM модель гіпертекстового документа

Принцип ієрархічного успадкування властивостей полягає в тому, що стилі, що назначено для батьківських об'єктів успадковуються вкладеними об'єктами всіх рівнів вкладеності. Ці стилі можна перевизначити в дочірніх об'єктах. Слід пам'ятати, що не всі стилі успадковуються. Для точного розуміння цього питання слід звертатись до описання кожного конкретного стиля [ДОВІДНИК ПО властивостям CSS](#).

Контрольні запитання

1. Правило підключення стилів.
2. Правила визначення селекторів стилів
3. Пріоритети при визначенні стилів. Використання конструкції !important
4. Інструмент розробника в браузері.
5. DOM модель гіпертекстового документа. Спадковість у стилізації.

Лабораторна робота №5.

CSS стилі. Стилi блоку.

Мета. Формування предметних компетентностей по блочному верстанню сайта.

Хід роботи:

1. Ознайомитись з теоретичним матеріалом “Одиниці вимірювання стилів” (Таблиця 5.1).
2. Ознайомитись з теоретичним матеріалом “Стилi Блоку” (Таблиця 5.2).
3. Поетапно реалізувати на комп’ютері розв’язок Задачі 5.1 (1-4 етап)
4. Виконати “Індивідуальну Задачу 5.2”
5. Розв’язати один варіант “Індивідуального завдання”.
6. Результати викласти на хостингу мережі Інтернет.

Теоретичний матеріал

1. Одиниці вимірювання стилів (Таблиця 5.1)

Одиниці вимірювання, що використовуються при налаштуванні стилів:

- 1 -одиниці вимірювання довжини, ширини, розміру шрифту
- 2 -одиниці визначення кольору

2. Стилi блоку (Таблиця 5.2)

- 1 - width
- 2 - height
- 3 - border
- 4 - padding
- 5 - margin
- 6 - float
- 7 - clear

Слід пам’ятати, що стиль width не охоплює товщину лівого і правого бордюра і розмір лівого і правого зовнішнього відступу у екземпляра об’єкта.

Для опанування правил блочного верстання сайту розв’яжемо наступну задачу.

Задача 5.1 У гіпертекстовому документі, що було створено до Задачі 2.1 розташувати блоки згідно макету, що наведено на рисунку (рис.5.1)



Рис. 5.1 Модель макету сайту для Задачі 5.1

Розв’язок (таблиця 5.1.1 – 5.1.4)

Для розв’язку цієї задачі скористуємось однією властивістю CSS стилізації, що не потрапляє до нашого списку блочних стилів -

background-color:red;

Етап 1. Налаштовуємо кольорову гамму всіх блоків (Таблиця 5.1.1)

Таблиця 5.1.1 Етап 1 розв’язку Задачі 5.1

5.1.html <pre><!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/5.1.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div></pre>	5.1.css <pre>#container{background- color:rgb(0,255,255);} #header{background-color:rgb(106,168,79);} #menu{background-color:rgb(142,124,195);} #content{background- color:rgb(234,153,153);} #footer{background-color:rgb(17,85,204);}</pre>
--	---

<pre> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html> </pre>	
	

Ми бачимо чотири різнокольорові блоки. Блок `id="container"` сховався під видимими блоками. Тут ми скористались селекторами стилів, що пов'язуються з унікальною ідентифікацією екземпляра об'єкта.

Етап 2. Налаштування ширини і висоти об'єктів (Таблиця 5.1.2).

На цьому етапі нам необхідно зробити деякі числові припущення:

Зовнішній блок `id="container"`: Ширина 100% ширини екрана, висоту не встановлюємо, вона буде відповідати сумарній ширині вкладених блоків

Блок з заголовком `id="header"`: Ширина 100% ширини екрана, висота 150px

Блок меню `id="menu"`: Висота 300px, ширина 25%

Блок контент `id="content"`: Висота 300px, ширина 75%

Блок футер `id="footer"`: Ширина 100% ширини екрана, висота 100px

Таблиця 5.1.2 Етап 2 розв'язку Задачі 5.1

5.1.html <pre> <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/5.1.css"> </pre>	5.1.css <pre> #container{ background-color:rgb(0,255,255); width:100%; } #header{ background-color:rgb(106,168,79); width:100%; height:150px; </pre>
--	--

<pre> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html> </pre>	<pre> } #menu{ background-color:rgb(142,124,195); width:25%; height:300px; } #content{ background-color:rgb(234,153,153); width:75%; height:300px; } #footer{ background-color:rgb(17,85,204); width:100%; height:100px; } </pre>
	

Етап 3. Поставити Блок з заголовком `id="menu"` та Блок контент `id="content"` поруч (Таблиця 5.1.3).

Для цього:

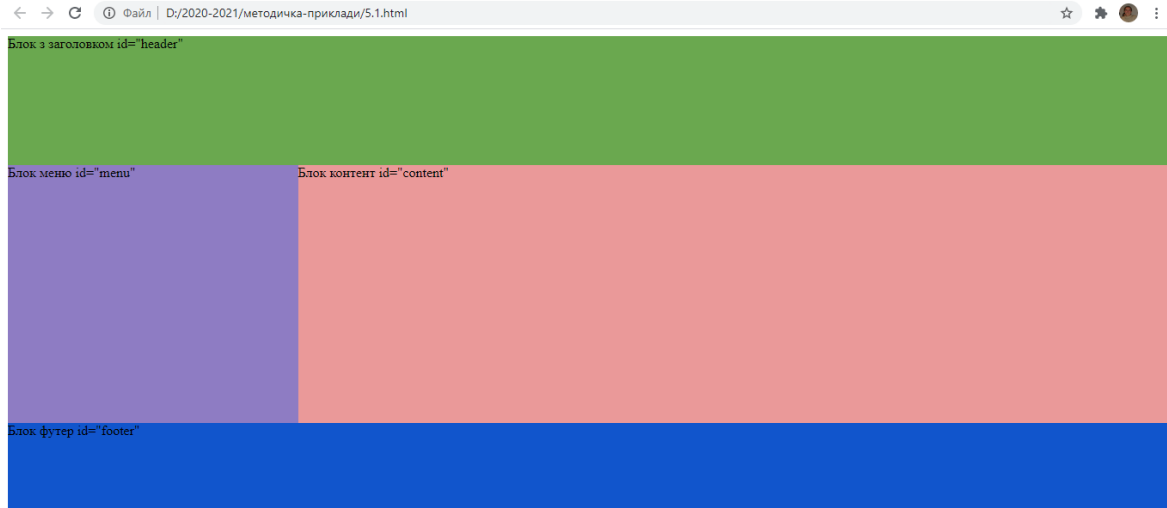
Блок меню `id="menu"`: `float:left` (обтікання по лівому краю наступного об'єкта)

Блок контент `id="content"`: `float:left` (обтікання по лівому краю наступного об'єкта)

Блок футер `id="footer"`: `clear:both;`

Таблиця 5.1.3 Етап 3 розв'язку Задачі 5.1

5.1.html <!doctype html>	5.1.css #container{
------------------------------------	-------------------------------

<pre> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/5.1.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком </div> <div id="menu"> Блок меню </div> <div id="content"> Блок контент </div> <div id="footer"> Блок футер </div> </div> </body> </html> </pre>	<pre> background-color:rgb(0,255,255); width:100%; } #header{ background-color:rgb(106,168,79); width:100%; height:150px; } #menu{ background-color:rgb(142,124,195); width:25%; height:300px; float:left; } #content{ background-color:rgb(234,153,153); width:75%; height:300px; float:left; } #footer{ background-color:rgb(17,85,204); width:100%; height:100px; clear:both; } </pre>
	

Етап 4. Налаштування відстані між частинами header, menu+content та footer (Таблиця 5.1.4)

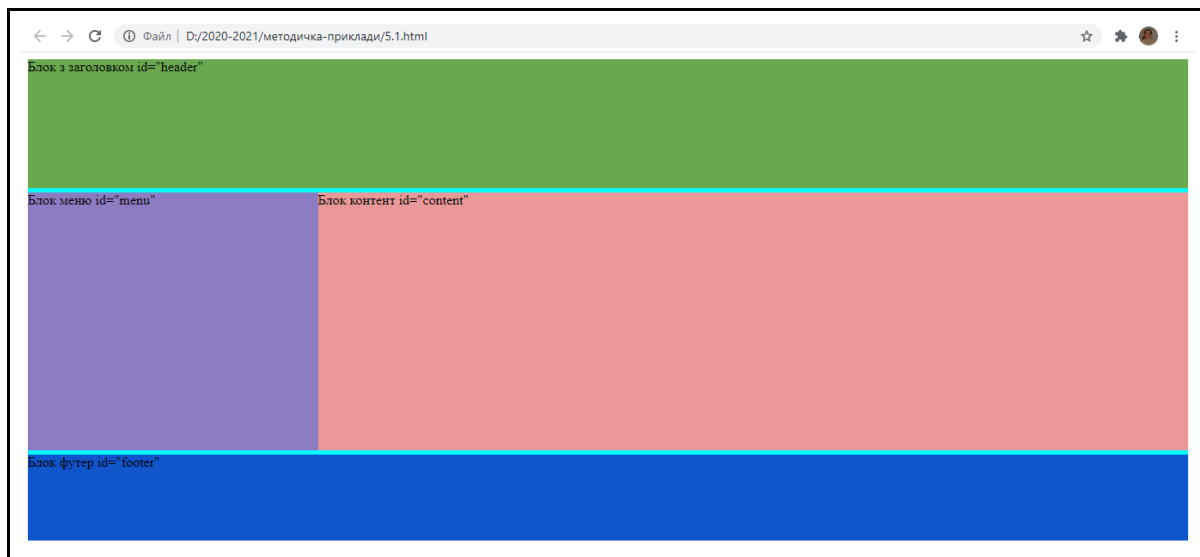
Для виконання цього етапу знов зробимо числові припущення

Зовнішній відступ зверху та знизу для блоків меню+контент 5 px

Зовнішній відступ оформимо як окремий клас стилів і підключимо до відповідних екземплярів об'єктів.

Таблиця 5.1.4 Етап 4 розв'язку Задачі 5.1

5.1.html <!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/5.1.css"> </head> <body> <div id="container"> <div id="header"> Блок з заголовком id="header" </div> <div id="menu" class="vidstup" > Блок меню id="menu" </div> <div id="content" class="vidstup" > Блок контент id="content" </div> <div id="footer"> Блок футер id="footer" </div> </div> </body> </html>	5.1.css #container{ background-color:rgb(0,255,255); width:100%; } #header{ background-color:rgb(106,168,79); width:100%; height:150px; } #menu{ background-color:rgb(142,124,195); width:25%; height:300px; float:left; } #content{ background-color:rgb(234,153,153); width:75%; height:300px; float:left; } #footer{ background-color:rgb(17,85,204); width:100%; height:100px; clear:both; } .vidstup{ margin-top:5px; margin-bottom:5px;}
---	--



Задачу розв'язано.

Таблиця 5.1 Одиниці вимірювання

Одиниці вимірювання довжини, ширини, розміру шрифту	em(ex)- маштабована одиниця шрифту, %- відсотки, px- піксели, in-дюйми, cm-сантіметри, mm-міліметри, pt- пункти , знаки, pc-піки 1in = 2.54cm = 25.4mm = 72pt = 6pc (ці одиниці не залежать від розміру екрану монітору і не рекомендовані для виводу на екран, а рекомендовані для друку) Одиниці виміру для виведення на екран em, px, % Одиниці em і ex залежать від шрифту і можуть бути різним для кожного елемента в документі. em це просто розмір шрифту. У елемент із шрифтом 2in, 1em означає 2in. Вираження таких розмірів, як поля та оббивка, в em означає, що вони пов'язані з розміром шрифту, і якщо користувач має великий шрифт (наприклад, на великому екрані) або маленький шрифт (наприклад, на портативному пристрої), розміри будуть відображатися пропорційно. Такі величини як 'text-indent: 1.5em' та 'margin: 1em' є надзвичайно поширені в CSS. font-size: 14px; font-size: 0.8em;(80% від поточного розміру шрифту) розміри шрифту
Одиниці визначення кольору	Назва: red, green, blue, black, yellow RGB 16 система числення: #fff(#ffffff)-білий #000(#000000)-чорний #555(#555555)-відтінок сірого кольору RGB 10 система числення: rgb(100%,100%,0%)-жовтий

	rgb(255, 255, 255)-білий rgb(0,0,0)-чорний rgb(100, 100, 100)-відтінок сірого кольору color: #FF00FF; color: rgb(100%,0%,100%); color:red;
Підключення сторонніх файлів	url(picture.gif) url(http://www.pix.org/lib1/pic278.gif)

Таблиця 5.2 Стили блоку

№	Стиль	Приклад	Варіанти застосування
1	width	width:20%; ширина в відсотках відраховується від ширини батьківського блоку width:200px;	
2	height	height:20%; висота в відсотках відраховується від ширини батьківського блоку height:200px;	
3	border	border:solid 2px black; встановлюється бордюр зі всіх сторін об'єкта solid - суцільний (dotted - точковий, dashed - рисочний) 2px - товщина бордюра black - колір бордюра	border-left:solid 2px black; border-right:solid 2px black; border-top:solid 2px black; border-bottom:solid 2px black;
4	margin	margin:2px; зовнішній відступ зі всіх сторін об'єкта margin:2px 3px; перше значення встановлює відступ від верхнього і нижнього краю, друге - від лівого і правого margin:2px 3px 2px; перше значення задає відступ від верхнього краю, друге - одночасно від лівого і	margin-left:2px; margin-right:3px; margin-top:5px; margin-bottom:2px;

		правого краю, а третє - від нижнього краю. margin:2px 3px 2px 3px; по черзі встановлюється відступ від верхнього, правого, нижнього і лівого краю.	
5	padding	padding:2px; внутрішній відступ зі всіх сторін об'єкта Більша кількість параметрів працює як в margin	padding-left:2px; padding-right:3px; padding-top:5px; padding-bottom:2px;
6	float	float:left; об'єкт притискається до лівокраю попереднього об'єкта і обтікає його праворуч	float:right;
7	clear	clear:both; відмінняє обтікання зі всіх сторін	

Індивідуальна Задача 5.2

1. На базі html файлу другої лабораторної роботи створити файл 5.2.html.

Підключити файл зі стилями 5.2.css

2. Для всіх блоків додати бордюр рожевого кольору, товщиною 3 px, сплошний

3. Налаштувати зовнішній відступ у всіх блоків 5 px

4. Зробити фон всіх блоків голубим

5. Зробити фон зовнішнього блоку жовтим з використанням id селектору

6 Налаштувати фон блоків 3, 4, 5 за допомогою селектору класа в сірий колір.

7. Для 3, 4, 5 блоків зробити внутрішній відступ 5px (в селекторі класу)

8. Налаштувати ширину 3, 4, 5 блоків у співвідношенні 25%, 60%, 13%

9. Вирівняти 3, 4, 5 блоки в один ряд (використовуючи float:, clear:.....)(не очікуйте ідеально вирівняного сайту за шириною)

Індивідуальні завдання.

Розробити макет сайту згідно з моделлю, що наведена в індивідуальному завданні. Макет має розтягуватись до ширини екрану дівайса, на котрому переглядатиметься. Числові еквіваленти кольорів отримати за допомогою редактору Paint, або любого іншого графічного редактора.

Варіант 1



Варіант 2



Варіант 3



Варіант 4



Варіант 5



Лабораторна робота №6.

CSS стилі. Стили тексту і шрифту.

Мета. Формування предметних компетентностей по налаштуванню тексту гіпертекстового документа

Хід роботи:

1. Ознайомитись з стилями шрифту (Таблиця 6.1):
 1. font-family
 2. font-style
 3. font-variant
 4. font-weight
 5. font-size
 6. font
2. Ознайомитись з стилями тексту (Таблиця 6.2):
 1. word-spacing
 2. letter-spacing
 3. text-decoration
 4. vertical-align
 5. text-transform
 6. text-align
 7. text-indent
 8. line-height
3. Виконайте Приклад 6.1.
4. Виконайте варіант індивідуального завдання.
5. Розмістіть результуючу сторінку lab6.html на хостингу.

Теоретичний матеріал.

Кожний сайт потребує налаштувань шрифтів і тексту. Проводячи паралель з текстовим процесором розуміємо важливість цих налаштувань. Основний стиль шрифту - це font (Таблиця 6.1).

Основними при налаштуванні тексту слід вважати стиль text-align, цей стиль найчастіше застосовується (Таблиця 6.2).

Таблиця 6.1. Стили налаштування шрифтів

1	font-family	Використовується щоб оголосити конкретний шрифт для використання або сімейство шрифтів або і те й інше. Оскільки неможливо передбачити, встановлений той чи інший шрифт на комп'ютері відвідувача вашого сайту, рекомендується прописувати всі можливі варіанти однотипних шрифтів. В такому випадку браузер буде перевіряти їх наявність, послідовно перебираючи запропоновані варіанти. font-family: "Times New Roman", Georgia, Serif; font-family: serif; font-family: sans-serif; font-family: monospace; font-family: cursive; font-family: fantasy; font-family: system-ui; font-family: inherit; font-family: initial;	family-name - Назва (ім'я) сімейства шрифтів, наприклад, Times, Courier, Arial. Рекомендується вказувати разом з базовим сімейством. generic-family Базове сімейство. CSS визначає п'ять базових сімейств шрифтів: Шрифти із зарубками - Serif (Times New Roman, Times, Garamond, Georgia) Рубані шрифти - Sans-serif (Helvetica, Geneva, Arial, Verdana, Trebuchet, Univers) Моноширинних шрифти - Monospace (Courier, Courier New, Andale Mono) Рукописні шрифти - Cursive (Comic Sans, Gabriola, Monotype Corsiva, Author, Zapf Chancery) Алегоричні шрифти (Western, Woodblock, Klingon) initial Встановлює значення властивості в значення за замовчуванням. inherit Успадковує значення властивості від батьківського елемента.
---	-------------	--	---

2	font-style	Стиль шрифту italic (курсив), oblique (нахилений стиль), и normal (стиль за замовчуванням).	font-style: normal; font-style: italic; font-style: oblique; font-style: inherit; font-style: initial;
3	font-variant	small-caps и normal.	font-variant: small-caps;
4	font-weight	Встановлює "вагу" шрифту, роблячи його більш-менш "жирним"	font-weight: normal; font-weight: bold; font-weight: lighter; font-weight: bolder; font-weight: 100; font-weight: 200; font-weight: 300;
5	font-size	Встановлює кегль шрифту. Можна оголосити абсолютний розмір, відносний, або у відсотках від розміру за замовчуванням.	font-size: 14px; font-size: 0.8em; font-size: 80%;
6	font	Короткий запис всіх властивостей шрифту. Порядок значень важливий:font {font-style font-variant font-weight font-size/line-height font-family;}. Довільне зі значень може бути пропущене.	P {font: bold 12pt/14pt Helvetica,sans-serif;} font: 12pt/14pt sans-serif; font: 80% sans-serif; font: x-large/110% "new century schoolbook", serif; font: bold italic large Palatino, serif; font: normal small-caps 120%/120% fantasy; font: condensed oblique 12pt "Helvetica Neue", serif;

Таблиця 6.2. Стили налаштування тексту

1	word-spacing	Встановлює кількість порожнього простору між словами. P {word-spacing: 0.5em;}
2	letter-spacing	Встановлює кількість порожнього простору між буквами. P {letter-spacing: 0.5em;}
3	text-decoration	Встановлює ефекти, такі як підкреслення, перекреслення і миготіння. Можливі комбінації цих ефектів. U {text-decoration: underline;} .old {text-decoration: line-through;}
4	vertical-align	Встановлює вертикальне вирівнювання базової лінії елемента по відношенню до висоти в батьківському елементі. Від'ємні значення дозволені. vertical-align: super; vertical-align: 50%;
5	text-transform	Змінює регістр набору тексту. H1 {text-transform: uppercase;} .title {text-transform: capitalize;}
6	text-align	Встановлює горизонтальне вирівнювання тексту. P {text-align: justify;} H4 {text-align: center;}

7	text-indent	Встановлює відступ першого рядка елемента (новий рядок). Від'ємні значення дозволені. P {text-indent: 5em;} H2 {text-indent: -25px;}
8	line-height	Встановлює відстань між базовими лініями в елементі, повну висоту рядка. P {line-height: 18pt;} H2 {line-height: 200%;}

Приклад 6.1

Створити абзац з текстом і встановити наступні налаштування (Таблиця 6.3)

Таблиця 6.3. Вимоги до налаштування тексту в абзаці

1	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Comic Sans MS 8px курсив 1 по ширині 5,25 px за бажанням
---	--	--

Розв'язок (таблиця 6.3.1)

Таблиця 6.3.1 Розв'язок прикладу 6.1

lab6.1.html	6.1.css
<pre><!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/6.1.css"> </head> <body> <div id="content"> Блок контент id="content" ВІД ЗИМИ І ДО ЗИМИ <p id="pp1">Січень січе, Лютий лютує, Березень</pre>	<pre>#pp1 {font:italic 8px/1em "Comic Sans MS"; text-align: justify; text-indent: 5.25px; }</pre>

плаче, Квітень квітує. Травень під ноги Стелить травицю. Червень складає Сіно в копиці. Липень медовий Бджілок чарує. Серпень чудовий Булки дарує. Вересень трусить Груші в садочку. Жовтень гаптує Клену сорочку. Йде листопад, Застеля килими. Грудень надходить — Початок зими. Ф. Петров	
---	--

</div>

</body>

</html>

Блок контент id="content" ВІДЗИМИ І ДОЗИМИ

*Січень січе, Лютий лютує, Березень плаче, Квітень квітує. Травень під ноги
Стелить травицю. Червень складає Сіно в копиці. Липень медовий Бджілок
чарує. Серпень чудовий Булки дарує. Вересень трусить Груші в садочку.
Жовтень гаптує Клену сорочку. Йде листопад, Застеля килими. Грудень
надходить — Початок зими. Ф. Петров*

Індивідуальні задачі.

На сторінці створити 3 абзаці. Для кожного з абзаців виконати налаштування згідно вимог, що задано в таблиці (Таблиця 6.4).

Варіант 1.

1 абзац - 4 налаштування

2 абзац - 3 налаштування

3 абзац - 2 налаштування

Варіант 2.

1 абзац - 3 налаштування

2 абзац - 2 налаштування

3 абзац - 1 налаштування

Варіант 3.

1 абзац - 2 налаштування

2 абзац - 1 налаштування

3 абзац - 4 налаштування

Варіант 4.

1 абзац - 4 налаштування

2 абзац - 3 налаштування

3 абзац - 2 налаштування

Варіант 5.

1 абзац - 4 налаштування

2 абзац - 1 налаштування

3 абзац - 2 налаштування

Варіант 6.

1 абзац - 5 налаштування

2 абзац - 6 налаштування

3 абзац - 8 налаштування

Варіант 7.

1 абзац - 6 налаштування

2 абзац - 7 налаштування

3 абзац - 8 налаштування

Варіант 8.

1 абзац - 5 налаштування

2 абзац - 8 налаштування

3 абзац - 6 налаштування

Варіант 9.

1 абзац - 8 налаштування

2 абзац - 7 налаштування

3 абзац - 6 налаштування

Варіант 10.

1 абзац - 6 налаштування

2 абзац - 5 налаштування

3 абзац - 7 налаштування

Таблиця 6.4. Варіанти налаштувань

1	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Times New Roman 14px курсив 1,5 по лівому краю 5,25 px за бажанням
2	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Arial 16px жирний 1,5 по правому краю краю -5,25 px за бажанням
3	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Comic Sans MS 10px підкреслений 1 по центру не має за бажанням
4	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Verdana 16px нахилений 2 по ширині 5,25 px за бажанням
5	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Times New Roman 18px звичайний 1 по правому краю 7 px за бажанням
6	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал	Arial 10 px курсив 2

	Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	по правому краю краю 7,25 рх за бажанням
7	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Comic Sans MS 14 рх перекреслений 1,5 по центру -7,25 рх за бажанням
8	Шрифт Розмір шрифту Стиль шрифту Міжрядковий інтервал Вирівнювання тексту в абзаці Відступ червоного рядка Налаштування відступу між словами	Verdana 16рх нахилений 1 по ширині 6,25 рх за бажанням

Лабораторна робота №7.

CSS стилі. Стилi налаштування кольорової гами гіпертекстового документа.

Мета. Формування предметних компетентностей по налаштуванню кольорової гами гіпертекстового документа

Хід роботи:

1. Ознайомитись з стилями color та background (Таблиця 7.1):
 1. color
 2. background-color
 3. background-image
 4. background-repeat
 5. background-attachment
 6. background-position
 7. background
2. Виконайте Приклад 7.1.
3. Виконайте варіант індивідуального завдання.
4. Розмістіть результуючу сторінку lab7.html на хостингу.
5. Налаштуйте остаточний дизайн вашого підсумкового сайту в межах творчого завдання.

Теоретичний матеріал.

Таблиця 7.1. Стилi налаштування кольору тексту і фону

1	color	Встановлює колір елемента. Для тексту це колір тексту, для інших елементів, таких як <hr> - колір лінії. p {color: teal;}
2	background-color	Встановлює колір фону. h1 {background-color: white;}
3	background-image	Встановлює картинку в якості фону. Повторяється фонові картинка зліва - направо, зверху - вниз. div {background-image: url(bg41.gif);}
4	background-repeat	Встановлює стиль повторення фону. no-repeat

		Встановлює одне фонове зображення в елементі без його повторень, положення якого визначається властивістю background-position (за замовчуванням в лівому верхньому куті). Аналогічно no-repeat по-repeat. repeat Фонове зображення повторюється по горизонталі і вертикалі. Аналогічно repeat repeat. repeat-x Фоновий малюнок повторюється тільки по горизонталі. repeat-y Фоновий малюнок повторюється тільки по вертикалі. inherit Успадковує значення батька. space Зображення повторюється стільки раз, щоб повністю заповнити область; якщо це не вдається, між картинками додається порожній простір. round Зображення повторюється так, щоб в області помістилося ціле число малюнків; якщо це не вдається зробити, то фонові малюнки масштабуються. div {background-repeat: no-repeat;}
5	background-attachment	Визначає чи буде фон скроліроваться разом з текстом. fixed Робить фонове зображення елемента нерухомим. scroll Дозволяє переміщатися фону разом з вмістом. inherit Успадковує значення батьківського об'єкта. local Фон фіксується з урахуванням поведінки елемента. Якщо елемент має прокрутку, то фон буде прокручуватися разом з вмістом, але фон виходить за рамки елемента залишається на місці div {background-attachment: fixed;}
6	background-position	Встановлює початкову позицію фоновому кольору або малюнка. Якщо це колір, то він заповнює всі починаючи з цієї позиції. Якщо малюнок, то перша картинка поміщається як зазначено тут, а потім

		починає повторюватися відповідно до вищеописаних властивостей. top left = left top = 0% 0% (в лівому верхньому кутку) top = top center = center top = 50% 0% (по центру вгорі) right top = top right = 100% 0% (у правому верхньому куті) left = left center = center left = 0% 50% (по лівому краю і по центру) center = center center = 50% 50% (по центру) right = right center = center right = 100% 50% (по правому краю і по центру) bottom left = left bottom = 0% 100% (в лівому нижньому кутку) bottom = bottom center = center bottom = 50% 100% (по центру внизу) bottom right = right bottom = 100% 100% (в правому нижньому кутку) div {background-position: top center;}
7	background	короткий запис всіх властивостей фону. Порядок не важливий. BODY {background: white url(bg41.gif) fixed center;}

Приклад 7.1. Налаштувати відображення фонової картинки вгорі - по центру блоку і зафіксувати без повторень, фон зробити синім

Розв'язок (таблиця 7.2)

Таблиця 7.2 Розв'язок прикладу 7.1

lab7.1.html	7.1.css
<pre><!doctype html> <html> <head> <title>Гіпертекстовий документ</title> <meta charset="utf-8"> <link rel="stylesheet" href="style/7.1.css"> </head> <body> <div id="content"> Блок контент id="content" ВІД ЗИМИ І ДО ЗИМИ Січень січе, Лютий лютує, Березень плаче,</pre>	<pre>#content {background: blue url("7.1.png") fixed center top no- repeat;}</pre>

Квітень квітує. Травень під ноги Стелить
травицю. Червень складає Сіно в копиці. Липень
медовий Бджілок чарує. Серпень чудовий Булки
дарує. Вересень трусить Груші в садочку.
Жовтень гаптує Клену сорочку. Йде листопад,
Застеля килими. Грудень надходить — Початок
зими. Ф. Петров

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

Блок контент id="content" ВІД ЗІМНИ І ДО ЗІМНИ Січень січе, Лютий лютує, Березень плаче, Квітень квітує. Травень під ноги
Стелить травицю. Червень складає Сіно в копиці. Липень медовий Бджілок чарує. Серпень чудовий Булки дарує. Вересень
трусить Груші в садочку. Жовтень гаптує Клену сорочку. Йде листопад, Застеля килими. Грудень надходить — Початок зими. Ф.
Петров

Індивідуальні завдання.

Створити на гіпертекстовій сторінці блок з довільним текстом. Для цього блоку налаштувати фон згідно з варіантом вашого завдання. Для виконання завдання підберіть фонову картинку невеликого розміру (20x20px). Скористайтесь коротким записом всіх властивостей фону (таблиця 7.3).

Таблиця 7.3. Індивідуальні завдання

Варіант 1	Встановити колір фону: рожевий Встановлює картинку в якості фона. Встановити стиль повторення фону: no-repeat Визначає чи буде фон скроліроваться разом з текстом: fixed Встановлює початкову позицію фонового кольору або малюнка. top left
Варіант 2	Встановити колір фону: синій Встановлює картинку в якості фона. Встановити стиль повторення фону: repeat Визначає чи буде фон скроліроваться разом з текстом: scroll Встановлює початкову позицію фонового кольору або малюнка. top
Варіант 3	Встановити колір фону: червоний Встановлює картинку в якості фона. Встановити стиль повторення фону: repeat-y Визначає чи буде фон скроліроваться разом з текстом:

	fixed Встановлює початкову позицію фонового кольору або малюнка. right top
Варіант 4	Встановити колір фону: зелений Встановлює картинку в якості фона. Встановити стиль повторення фону: repeat-x Визначає чи буде фон скролірватися разом з текстом: scroll Встановлює початкову позицію фонового кольору або малюнка. left
Варіант 5	Встановити колір фону: жовтий Встановлює картинку в якості фона. Встановити стиль повторення фону: space Визначає чи буде фон скролірватися разом з текстом: fixed Встановлює початкову позицію фонового кольору або малюнка. center
Варіант 6	Встановити колір фону: бірюзовий Встановлює картинку в якості фона. Встановити стиль повторення фону: round Визначає чи буде фон скролірватися разом з текстом: scroll Встановлює початкову позицію фонового кольору або малюнка. right
Варіант 7	Встановити колір фону: коричневий Встановлює картинку в якості фона. Встановити стиль повторення фону: repeat-y Визначає чи буде фон скролірватися разом з текстом: fixed Встановлює початкову позицію фонового кольору або малюнка. bottom left
Варіант 8	Встановити колір фону: чорний Встановлює картинку в якості фона. Встановити стиль повторення фону: repeat-x Визначає чи буде фон скролірватися разом з текстом: scroll

	Встановлює початкову позицію фонового кольору або малюнка. bottom
Варіант 9	Встановити колір фону: оранжевий Встановлює картинку в якості фона. Встановити стиль повторення фону:space Визначає чи буде фон скроліроваться разом з текстом: fixed Встановлює початкову позицію фонового кольору або малюнка. bottom right
Варіант 10	Встановити колір фону: рожевий Встановлює картинку в якості фона. Встановити стиль повторення фону:round Визначає чи буде фон скроліроваться разом з текстом: scroll Встановлює початкову позицію фонового кольору або малюнка. top left

Творче завдання.

Підготувати індивідуальний дизайн вашого сайту.

Лабораторна робота №8.

Мова HTML. Форми.

Мета. Формування предметних компетентностей у створенні форми

Хід роботи:

1. Ознайомитись з теоретичним матеріалом "Як форма збирає дані".
2. Ознайомитись з правилами створення полів форми на основі тегу input (9 штук).
 - 2.1. type="text"
 - 2.2. type="password"
 - 2.3. type="radio"
 - 2.4. type="checkbox"
 - 2.5. type="file"
 - 2.6. type="hidden"
 - 2.7. type="button"
 - 2.8. type="reset"
 - 2.9. type="submit"
3. Ознайомитись з правилами створення поля textarea (багаторядкове поле)
4. Ознайомитись з правилами створення розкривного списку варіантів, select list
5. Створити форму на основі прикладів таблиці. (Таблиця 8.1)
6. Розв'язати один варіант **"Індивідуального завдання"**.
7. Результати викласти на хостингу мережі Інтернет.

Теоретичний матеріал

Форма - це об'єкт браузера, що містить об'єкти-поля форми. Кожне поле форми збирає дані і передає ці дані на оброблення.

1. **Як форма збирає дані.** Тег form містить обов'язковий атрибут **action** (Таблиця 8.1 строка А), значенням якого є адреса файла-обробника форми. Після натискання на кнопку submit (Таблиця 8.1 строка 10) дані форми у вигляді пар ім'я=значення передаються в програму-обробник.
 - а. Пусте значення атрибуту action зазвичай сприймається, як передача даних в той самий файл, де міститься сама форма.

- b. Атрибут `method="get"`, в цьому випадку дані (`name=value`) передаються через приєднання до URL адреси файла-обробника форми (рис. 8.1).
- c. За умови передачі даних методом `post`, дані в файл-обробник надходять в стандартному вхідному потоці (цей потік шифрується при використанні `https` протоколу).

2. **Однострокове поле (`type="text"`)** введення текстової інформації (Таблиця 8.1 строка 3). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту `name` унікально ідентифікує поле. Значення атрибуту `value` необов'язкове. Значення, що вводиться в поле користувачем передається як значення змінної, що визначено в атрибуті `name`. Атрибут **size** виставляє видиму довжину поля в символах, **maxlength** налаштовує максимально допустиму довжину тексту, що вводиться в символах.

3. **Поле для введення паролю (`type="password"`)** (Таблиця 8.1 строка 4). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту `name` унікально ідентифікує поле. Значення атрибуту `value` необов'язкове, зазвичай не встановлюється. Значення, що вводиться в поле користувачем відображається у вигляді зірочок, передається у текстовому форматі як значення відповідної змінної `name`. Атрибут **size** виставляє видиму довжину поля в символах, **maxlength** налаштовує максимально допустиму довжину пароля в символах.

4. **Радіокнопка або перемикач (`type="radio"`)** (Таблиця 8.1 строка 5). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту `name` унікально ідентифікує одне поле або групу радіокнопок. В групі радіокнопок вибрати можна лише одну. Значення атрибуту `value` обов'язкове, саме воно передається як значення змінної. В групі радіокнопок всі кнопки різняться своїми значеннями атрибуту **value**. В програму-обробник приходить значення `value` з обраної (чекнутої) радіокнопки. Для встановлення обраної за замовчуванням радіокнопки в групі радіокнопок треба визначити атрибут **checked="checked"**.

5. **Чекбокс або прапорець (`type="checkbox"`)** (Таблиця 8.1 строка 6). Це

поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує одне поле. Можна створювати групу прапорців, але в групі ім'я кожного прапорця унікальне. Значення атрибуту **value** обов'язкове, саме воно передається як значення імені поля. В програму-обробник приходить значення **value** з обраного (чекнутого) прапорця. Для встановлення обраного за замовчуванням прапорця треба визначити атрибут **checked="checked"**.

6. **Поле для передачі файлової інформації (type="file")** (Таблиця 8.1 строка 7). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує одне поле. Файли пересилаються формою виключно методом **post**, має бути налаштовано атрибут **enctype**, що визначає спосіб кодування даних форми (**<form ... method="post" enctype="multipart/form-data">**).

7. **Приховане поле (type="hidden")** (Таблиця 8.1 строка 8). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує одне поле. Обов'язкове значення атрибуту **value**. Змінні і значення в обов'язковому порядку потрапляють з форми на оброблення.

8. **Кнопка (type="button")** (Таблиця 8.1 строка 9). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує одне поле. Значення атрибуту **value** визначає підпис на кнопці і значення, що передається на оброблення. Кнопка за замовчуванням не виконує жодної дії в формі.

9. **Кнопка "очистити" (type="reset")** (Таблиця 8.1 строка 10). Ця кнопка виставляє всі поля форми в початковий стан (очищає зміни користувача). Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує одне поле. Значення атрибуту **value** визначає підпис на кнопці і значення, що передається на оброблення.

10. **Кнопка "відправити" (type="submit")** (Таблиця 8.1 строка 11). Ця кнопка відправляє дані форми на оброблення в файл, що визначено в атрибуті

action тегу **form**. Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує поле. Значення атрибуту **value** визначає підпис на кнопці і значення, що передається на оброблення.

11. **Багаторядкове поле для введення текстової інформації** (<textarea ...>...</textarea>). (Таблиця 8.1 строка 1). Це поле надає можливість ввести великий, багаторядковий текст. Тег **textarea** має бінарну структуру (відкривається і закривається). Значення за замовчуванням визначається між тегамі. Атрибут **rows**, **cols** визначають відповідно кількість строк, кількість символів, що відображаються в полі. Це поле може бути однозначно ідентифіковано через **id**. В межах форми значення атрибуту **name** унікально ідентифікує поле.

12. **Розкривний список варіантів** (<select ...></select>). (Таблиця 8.1 строка 2). Це поле надає можливість вибрати 1 варіант з багатьох. За умови додавання атрибуту **multiple="multiple"** до тегу **select** створюється можливість багатозначного вибору (одночасного вибору декількох елементів списку). Атрибут **size** розкриває список на вказану кількість рядків (Таблиця 8.1 строка 2, другий приклад). Для зазначення вибраного за замовчуванням елементу використовується атрибут **selected="selected"**.

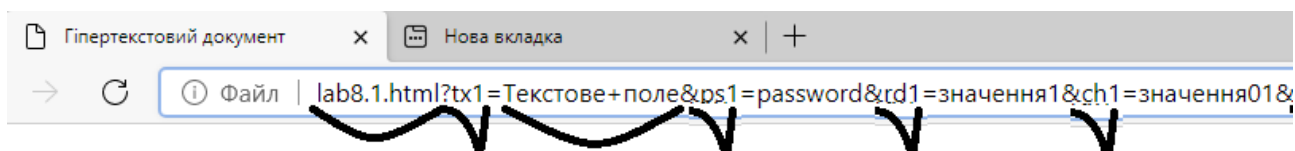
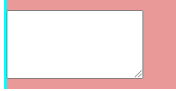
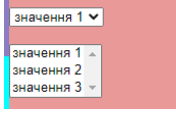
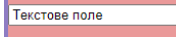
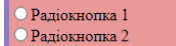
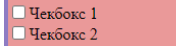
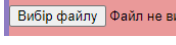
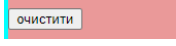


Рис.8.1 Приєднання пар **name=value** до URL адреси при передачі даних методом **get**

Таблиця 8.1 Форма

A		<form	method="get" (post)	action="">	enctype="multipart/form-data" (The enctype attribute can be used only if method="post")
1		<textarea name="area1" cols="20" rows="5"></textarea>			
2		<select name="sl1"> <option value="v1" selected="selected">значення 1 </option> <option value="v2">значення 2 </option> <option value="v3">значення 3 </option> </select>			<select name="sl2" size="3" multiple="multiple"> ... </select>
3		<input	type="text"	name="tx1"	value="Текстове поле">
4		<input	type="password"	name="ps1"	value="password1">
5		<input <input	type="radio" type="radio"	name="rd1" name="rd1"	value="значення1">Радіокнопка 1 value="значення2">Радіокнопка 2
6		<input <input	type="checkbox" type="checkbox"	name="ch1" name="ch2"	value="значення01">Чекбокс 1 value="значення02">Чекбокс 2
7		<input	type="file"	name="fl1">	(method="post")
8		<input	type="hidden"	name="hd1"	value="приховане значення">
9		<input	type="button"	name="bt1"	value="кнопка">
10		<input	type="reset"	name="rs1"	value="очистити">
11		<input	type="submit"	name="sb1"	value="відправити">
		</form>			

Індивідуальні завдання.

Виконати завдання за вашим варіантом

До виконаного завдання створити звітну сторінку наступної структури:

1. Картинка з вашим варіантом завдання
2. Форма, що структурована на основі таблиці
3. В графічному редакторі розписати імена змінних і їх значення з адресного рядка браузера після відправлення форми. Приєднати рисунок до звітної сторінки з формою

Завдання 1

Створити форму для додавання даних про нарахування заробітної плати:

ПІБ	Однорядкове текстове поле. Кількість відкритих символів 30.
Оклад	Однорядкове текстове поле. Обмежити довжину рядка, що можна ввести 6 символами.
Чи основне місце роботи?	Перемикач (група з 2 радіокнопок)
Додаткові нарахування	Чотири прапорці (4 чекбокса)

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 2

Створити форму для додавання персональних даних про абітурієнтів:

Країна	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів).
Індекс	Однорядкове текстове поле. Обмежити довжину рядка, що можна ввести 6 символами.
Вид документу про освіту	Група радіокнопок (перемикачів). Придумати 3 варіанти значень перемикачів
Автобіографія	Багаторядкове поле. Текст за замовчуванням "Введіть стислу автобіографію"

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 3

Створити форму для введення інформації про бажаний для клієнта варіант міжміського обміну. Повинні вводитись наступні дані:

ПІБ	Однорядкове текстове поле. Кількість відкритих символів 30.
Кількість кімнат у клієнта	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів).
Бажані міста	П'ять прапорців (5 чекбоксів).

України	
Бажана кількість кімнат	Група з 5 перемикачів (з 5 радіокнопок)
Додаткові умови	Багаторядкове поле. Текст за замовчуванням "Введіть ваш пріоритетний варіант".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 4

Створити форму для реєстрації користувача:

Логін	Однорядкове текстове поле. Кількість відкритих символів 15. Максимальна можлива довжина логіна 10 символів.
Пароль	Поле для введення паролю
Країна	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням "Україна"
Стать клієнта	Група з 2 перемикачів (з 2 радіокнопок).
Напрямки інтересів	П'ять прапорців (5 чекбоксів).

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 5

Створити форму для входу користувача в систему. Має вводитись наступна інформація:

Логін	Однорядкове текстове поле. Кількість відкритих символів 15. Максимальна можлива довжина логіна 10 символів.
Пароль	Поле для введення паролю
Зберегти логін і пароль для наступного входу?	Група з 2 перемикачів (з 2 радіокнопок).

Виберіть бажану мову	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням "українська мова"
----------------------	---

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 7

Створити форму для формування запиту розширеного пошуку. Мають вводитись наступні дані:

Ключове слово або фраза	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина фрази 200 символів.
Діапазон дат створення результату	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням "Поточний рік"
Мова результату	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням ("українська")
Формат результатів	П'ять прапорців (5 чекбоксів). Можливі формати: відео файли, картинки, pdf файли, всі результати, аудіо файли

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 8

Створити форму для пошуку матеріалів у бібліотеці. Має вводитись наступна інформація:

Ключове слово або фраза	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина фрази 200 символів.
Рік публікації	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів).
Характер публікації	П'ять прапорців (5 чекбоксів). Можливі варіанти: "Стаття", "Дисертація", "Монографія" ...
Мова результату	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням ("українська")

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 9

Створити форму для замовлення комп'ютера.

Процесор	Вибір з розкритого списку варіантів (вказати 5 можливих варіантів).
Пам'ять	Однорядкове текстове поле. Кількість відкритих символів 20. Максимальна можлива довжина фрази 20 символів.
Мережевий адаптер	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням "Поточний рік"
Додаткові вимоги	Багаторядкове поле. Текст за замовчуванням "Введіть бажані параметри комп'ютера".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 10

Створити форму для введення адреси. Має вводиться наступна інформація:

Країна	Вибір з розкритого списку варіантів (вказати 5 можливих варіантів). За замовчуванням "Україна"
Поштовий індекс	Однорядкове текстове поле. Кількість відкритих символів 6. Максимальна можлива довжина фрази 6 символів.
Місто	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина фрази 40 символів.
Вказівка на те, чи домашня, чи робоча адреса	Група з 2 перемикачів (з 2 радіокнопок). Значення за замовчуванням відсутнє

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 11

Створити форму для оформлення замовлення на придбання комп'ютера. Має вводиться наступна інформація:

Процесор	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням
Пам'ять	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина фрази 20 символів.
Необхідність камери	Група з 2 перемикачів (з 2 радіокнопок). Значення за замовчуванням "Так"
Додаткові вимоги	Багаторядкове поле. Текст за замовчуванням "Введіть бажані параметри комп'ютера".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 12

Створити форму для завантаження файлової інформації на сервер. Має вводитись наступна інформація:

Виберіть файл на локальному комп'ютері	Поле для передачі файлової інформації
Ім'я користувача	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина імені 40 символів.
Тип файла	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів). За замовчуванням "pdf"
Опис вмісту файла	Багаторядкове поле. Текст за замовчуванням "Введіть бажані параметри комп'ютера".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом post в поточний файл. В формі має бути правильно налаштовано атрибут enctype. Для виконання 3 пункта індивідуального завдання перешліть всі дані з форми крім файлової методом get

Завдання 13

Створити форму для замовлення товарів. Мають вводитись наступні дані:

Назва товару	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів).
--------------	--

Кількість	Однорядкове текстове поле. Кількість відкритих символів 4. Максимальна можлива довжина 4 символа.
Адреса доставки	Багаторядкове поле. Текст за замовчуванням "Введіть повну адресу".
Виберіть країну	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів). За замовчуванням "Україна"

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 14

Створити форму для замовлення ремонту комп'ютера. Має вводитись наступна інформація:

Ім'я клієнта	Однорядкове текстове поле. Кількість відкритих символів 20. Максимальна можлива довжина 20 символів.
Країна	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням Україна
Процесор	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів). За замовчуванням "Україна"
Характер несправності	Багаторядкове поле. Текст за замовчуванням "Введіть повну адресу".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Завдання 15

Створити форму для оформлення замовлення на ремонт автомобіля. Повинна вводитись наступна інформація:

ПІБ замовника	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина 60 символів.
Місто	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів). За замовчуванням "Одеса"

Модель	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням не встановлюється
Характер несправності	Багаторядкове поле. Текст за замовчуванням "Введіть характер несправності".

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Задание 16

Створити форму для замовлення автомобіля. Має вводитись наступна інформація:

ПІБ замовника	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина 60 символів.
Країна	Вибір з розкривного списку варіантів (вказати 5 можливих варіантів). За замовчуванням "Україна"
Модель	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням не встановлюється
Колір	П'ять прапорців (5 чекбоксів).
Додаткові умови вибору	Багаторядкове поле. Текст за замовчуванням "Введіть ваші базові побажання комплектації"

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Задание 17

Створити форму для пошуку в каталозі бібліотеки. Має вводитись наступна інформація:

Тема книги	Вибір зі списку варіантів (вказати 5 можливих варіантів, що відкриті). Вказати варіант за замовчуванням
Автор	Однорядкове текстове поле. Кількість відкритих символів 40. Максимальна можлива довжина 60 символів.

Мова видання	П'ять прапорців (5 чекбоксів).
Діапазон років видання	Група з 5 перемикачів (з 5 радіокнопок). Значення за замовчуванням не встановлюється

В формі має працювати кнопка "Очистити форму".

Форма має направляти на оброблення дані методом get в поточний файл.

Лабораторна робота №9.

Мова JavaScript. Підключення до гіпертекстового документа. Оброблення подій об'єктів браузера.

Мета. Формування предметних компетентностей у опрацюванні подій об'єктів браузера.

Хід роботи:

1. Ознайомитись з переліком базових подій, на котрі реагують об'єкти браузера. (Таблиця 9.1)
2. Створити гіпертекстовий документ на основі **Прикладу 9.1.**
3. Виконати індивідуальне завдання. (Таблиця 9.3)
4. Результати викласти на хостингу мережі Інтернет.

Теоретичний матеріал:

Перелік базових подій, на котрі реагують об'єкти браузеру доволі великий.

Щоб зрозуміти сутність, слід ознайомитись з синтаксисом виклику події, зрозуміти для яких об'єктів ця подія може бути застосована і при яких умовах ця подія відбувається (Таблиця 9.1). Можна ознайомитись з розширеною версією переліку подій на W3school [1].

Таблиця 9.1. Перелік базових подій, на котрі реагують об'єкти гіпертекстового документа

onabort	Користувач зупиняє завантаження документа, зображення (Image, Document)
onblur	Користувач прибирає фокус із об'єкта (Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, Document)
onchange	Користувач змінює зміст елемента форми (FileUpload, Select, Text)

onclick	Користувач натискає кнопку миші (Button, Checkbox, document, Link, Radio, Reset, Submit , і т.д. – майже всі об'єкти
ondblclick	Користувач двічі натискає на кнопку миші (Button, Checkbox, document, FileUpload, Hidden, Link, Password, Radio, Reset, Select, Submit, Image)
ondragdrop	Користувач перетягує об'єкт у вікно
onerror	Виникла помилка під час завантаження (Image, window)
onfocus	Користувач перемістив фокус на об'єкт (Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, window)
onkeydown	Користувач натиснув клавішу (document, Image, Link, Text, Textarea)
onkeypress	Користувач натиснув клавішу і відпустив. (document, Image, Link)
onload	Закінчено завантаження документа (Frame, Image, Layer)
onmousedown	Користувач натиснув кнопку миші. (Button, document, Link)
onmouseout	Користувач перемістив курсор за межі об'єкта (Area, Layer, Link)
onmouseover	Користувач помістив курсор миші над об'єктом (Area, Layer, Link)

onreset	Користувач натиснув кнопку Reset у формі (Form)
onresize	Користувач змінив розмір вікна (Frame, window)
onselect	Користувач виділив текст (Text, Textarea)
onsubmit	Користувач відправив дані з форми (Form)
unload	Користувач залишає сторінку, наприклад, завантажуючи інший документ (Frame, window)

Приклад 9.1 Запустити метод alert('Назва події, що відбулась') при настанні події (1 колонка "подія") для визначених в центральній, в другій, колонці об'єктів. Розв'язок наведено в 3 колонці (Таблиця 9.2). При програмній реалізації цього прикладу зверніть увагу на правильне оформлення форми з відповідними полями.

Таблиця 9.2. Умова і розв'язок задачі з прикладу 9.1

подія	об'єкт	розв'язання
onload	document	<body onload ="alert('документ загрузено')">
onunload	document	<body onload="alert('документ загрузено')" onunload ="alert('розпочато виграження документа')">
onclick	Textarea	<textarea name="area1" cols="20" rows="5" onclick ="alert('клікнули мишкою в полі')"></textarea>
ondblclick	Textarea	<textarea name="area1" cols="20" rows="5" onclick="alert('клікнули мишкою в полі')" ondblclick ="alert('клікнули двічі мишкою в полі')"></textarea>
onmouseover	Image	onmouseover ="alert('Мишка наїхала на рисунок')">
onmouseout	Image	onmouseover="alert('Мишка наїхала на рисунок')"</code> <code>onmouseout="alert('Мишка зіхала з рисунку')"></code>
onfocus	Text	<code><input type="text" name="tx1" onfocus="alert('Поле в фокусі')"></code>
onblur	Text	<code><input type="text" name="tx1" onfocus="alert('Поле в фокусі')"</code> <code>onblur="alert('З поля знято фокус')"></code>
onchange	Select	<code><select name="sl1" onchange="alert('Змінено вибраний пункт')"></code> <code><option value="v1" selected="selected">значення 1</code> <code></option></code> <code><option value="v2">значення 2 </option></code> <code><option value="v3">значення 3 </option></code> <code></select></code>

Індивідуальні завдання:

Запустити метод `alert('Назва події, що відбулась')` при настанні 3 різних подій для 2 об'єктів. Виконати роботу згідно з вашим варіантом (Таблиця 9.3). Додати результат до сторінки з результатом рішення Прикладу 9.1.

Таблиця 9.3. Умови індивідуальних задач

Варіант	подія	об'єкт
Варіант 1	<code>onload</code> <code>onclick</code> <code>onmouseover</code>	Document Textarea
Варіант 2	<code>onunload</code> <code>onfocus</code> <code>onblur</code>	Document Textarea
Варіант 3	<code>onclick</code> <code>onmouseover</code> <code>onmouseout</code>	Textarea Image
Варіант 4	<code>ondblclick</code> <code>onfocus</code> <code>onblur</code>	Textarea Text
Варіант 5	<code>onmouseover</code> <code>onmouseout</code> <code>onchange</code>	Image Select
Варіант 6	<code>onmouseout</code>	Image

	onclick onload	document
Вариант 7	onfocus ondblclick onmouseover	Text Image
Вариант 8	onblur onchange onmouseover	Text Select
Вариант 9	onchange ondblclick onfocus	Select Textarea
Вариант 10	ondblclick onfocus onblur	Textarea Image

Лабораторна робота №10.

Мова JavaScript. Функції. Керування стилями. Підключення JavaScript програм
Мета. Формування предметних компетентностей у динамічному налаштуванні властивостей стилів для об'єктів браузера. Функції. Підключення JavaScript програм.

Хід роботи:

1. Ознайомитись з синтаксисом побудови функцій в мові JavaScript
2. Ознайомитись з питаннями "Визначення і виклик функції. Підключення JavaScript програми до гіпертекстової сторінки". Розглянути **Задачу 10.1.**
3. Ознайомитись з "Правила звернення до об'єктів браузера з JavaScript програми"
4. Ознайомитись з "Динамічне налаштування CSS стилів об'єктів"
5. Виконати **Задача 10.2**
6. Виконати індивідуальну задачу згідно з завданням лабораторної роботи №9
7. Викласти результати на хостинг.

Теоретичний матеріал:

При обробленні подій об'єктів браузера можна виконати будь-яку послідовність JavaScript команд. Зазвичай така послідовність команд оформлюється у вигляді функції.

Синтаксис функції JavaScript

Функція JavaScript визначається за допомогою **function** ключового слова, за яким слідує ім'я і дужки ().

Імена функцій можуть містити літери, цифри, підкреслення і знаки долара (ті ж правила, що і для змінних).

Код, який має виконати функцію, поміщається в фігурні дужки: {}. Код в фігурних дужках є тілом функції. Все, що передує тілу є заголовком функції.

function f_name1()

{

```
// код для виконання
```

```
}
```

Після імені функції в дужках можуть включатись імена параметрів, розділені комами:

```
(Parameter1, parameter2, ...)
```

```
function f_name2(parameter1, parameter2, parameter3)
```

```
{
```

```
    // код для виконання
```

```
}
```

Більш детально з функціями можна ознайомитись за підручником на W3school [1]

Визначення і виклик функції. Підключення JavaScript програми до гіпертекстової сторінки.

Для розуміння процедури визначення і виклику функції розглянемо задачу:

Задача 10.1. На подію наїзду миші на рисунок запустити виконання функції $f()$, а при з'їзді миші з рисунку запустити функцію $g()$. Визначити функції $f()$ і $g()$ в JavaScript програмі.

Розв'язок (таблиця 10.1)

Таблиця 10.1. Розв'язок задачі 10.1

lab10.html I варіант розв'язку (підключення файлу з JavaScript програмою (lab10.js) в частині head html файлу)	lab10.js I варіант розв'язку (визначення функцій в JavaScript програмі, файл lab10.js)	lab10.html II варіант розв'язку (визначення JavaScript програми в файлі html)
<!doctype html> <html> <head> <title>Гіпертекстовий документ до Лаб 10</title>	function f() {...} function g() {...}	<!doctype html> <html> <head> <title>Гіпертекстовий документ до Лаб 10</title> <meta charset="utf-8">

<pre> <meta charset="utf-8"> <link rel="stylesheet" href="style/style.css"> <script src="lab10.js"></scri pt> </head> <body> </body> </html> </pre>	<pre> <link rel="stylesheet" href="style/style.css"> <script> function f() {...} function g() {...} </script> </head> <body> </body> </html> </pre>
--	--

Правила звернення до об'єктів браузера з JavaScript програми (таблиця 10.2)

Таблиця 10.2 Звернення до об'єктів гіпертекстової сторінки з JavaScript програми

JavaScript object	HTML object
document.getElementById('pic1')	
document.getElementsByName('area1')	<textarea name="area1" cols="20" rows="5"></textarea>
document.images[0]	(перший рисунок в гіпертекстовому документі)
document.forms[0].field_name document.form_name.field_name	До форми і до полів форми можна звертатися за переліченими структурами.

В якості базового підходу оберемо однозначну ідентифікацію об'єктів з використанням id і звернення до об'єктів через конструкцію

document.getElementById

Функція \$ дозволяє спростити звернення до об'єктів браузера, що однозначно ідентифікуються через id (Наприклад: <div id="MyDiv">...</div>).

document.getElementById('MyDiv') будемо заміняти на \$('MyDiv')

Функція \$ обов'язкову має бути присутня в файлі зі скриптами.

```
function $()
{
    var elements = new Array();
    for (var i = 0; i < arguments.length; i++)
    {
        var element = arguments[i];
        if (typeof element == 'string')
            element = document.getElementById(element);
        if (arguments.length == 1)
            return element;
        elements.push(element);
    }
    return elements;
}
```

Динамічне налаштування CSS стилів об'єктів

Програма на мові JavaScript надає можливість динамічного налаштування CSS стилів.

document.getElementById('MyDiv').style.backgroundColor="blue";

При використанні функції \$ отримаємо спрощений код:

\$('#MyDiv').style.backgroundColor="blue";

Для керування властивостями з JavaScript програми ми маємо знати синтаксичну відповідність між стилем і властивістю об'єкта в JavaScript (Таблиця 10.1)

Таблиця 10.1. Відповідність назв стилів і JavaScript властивостей

CSS свойство	JavaScript аналог
1. background	Background
2. background-attachment	backgroundAttachment
3. background-color	backgroundColor
4. background-image	backgroundImage

5. background-position	backgroundPosition
6. background-repeat	backgroundRepeat
7. border	Border
8. border-bottom	borderBottom
9. border-bottom-color	borderBottomColor
10.border-bottom-style	borderBottomStyle
11.border-bottom-width	borderBottomWidth
12.border-color	borderColor
13.border-left	borderLeft
14.border-left-color	borderLeftColor
15.border-left-style	borderLeftStyle
16.border-left-width	borderLeftWidth
17.border-right	borderRight
18.border-right-color	borderRightColor
19.border-right-style	borderRightStyle
20.border-right-width	borderRightWidth
21.border-style	borderStyle
22.border-top	borderTop
23.border-top-color	borderTopColor
24.border-top-style	borderTopStyle
25.border-top-width	borderTopWidth
26.border-width	borderWidth
27.clear	Clear
28.clip	Clip
29.color	Color
30.cursor	Cursor
31.display	Display
32.filter	Filter
33.font	Font

34.font-family	fontFamily
35.font-size	fontSize
36.font-variant	fontVariant
37.font-weight	fontWeight
38.height	Height
39.left	Left
40.letter-spacing	letterSpacing
41.line-height	lineHeight
42.list-style	listStyle
43.list-style-image	listStyleImage
44.list-style-position	listStylePosition
45.list-style-type	listStyleType
46.margin	Margin
47.margin-bottom	marginBottom
48.margin-left	marginLeft
49.margin-right	marginRight
50.margin-top	marginTop
51.overflow	Overflow
52.padding	Padding
53.padding-bottom	paddingBottom
54.padding-left	paddingLeft
55.padding-right	paddingRight
56.padding-top	paddingTop
57.page-break-after	pageBreakAfter
58.page-break-before	pageBreakBefore
59.position	Position
60.float	styleFloat
61.text-align	textAlign
62.text-decoration	textDecoration

63.text-decoration: blink	textDecorationBlink
64.text-decoration: line-through	textDecorationLineThrough
65.text-decoration: none	textDecorationNone
66.text-decoration: overline	textDecorationOverline
67.text-decoration: underline	textDecorationUnderline
68.text-indent	textIndent
69.text-transform	textTransform
70.top	Top
71.vertical-align	verticalAlign
72.visibility	Visibility
73.width	Width
74.z-index	zIndex

Задача 10.2 На подію наїзду миші на рисунок запустити виконання функції, що створює рамку навколо рисунка і змінює ширину рисунку, а при з'їзді миші з рисунку запустити функцію, що змінює колір бордюру довкола рисунку на червоний і ширину рисунку до 50 px. Визначити функції в JavaScript програмі, що підключається з окремого файлу.

Розв'язок – (таблиця 10.2)

Таблиця 10.2. Розв'язок задачі 10.2

lab10.html I варіант розв'язку (підключення файла з JavaScript програмою (lab10.js) в частині head html файлу)	lab10.js I варіант розв'язку (визначення функцій в JavaScript програмі, файл lab10.js)
<pre><!doctype html> <html> <head> <title>Гіпертекстовий документ до Лаб 10</title> <meta charset="utf-8"> <link rel="stylesheet"</pre>	<pre>function \$() { var elements = new Array(); for (var i = 0; i < arguments.length; i++) { var element = arguments[i]; if (typeof element == 'string') element =</pre>

<pre> href="style/style.css"> <script src="lab10.js"></script> </head> <body> </body> </html> </pre>	<pre> document.getElementById(element); if (arguments.length == 1) return element; elements.push(element); } return elements; } function f() { alert('Наступила подія onmouseover. Після закриття цього вікна ви побачите жовтий бардюр і розширення зображення'); \$("pic1").style.border="solid 2px yellow"; \$("pic1").style.width="500px"; } function g() { alert('Наступила подія onmouseout. Після закриття цього вікна ви побачите зміну коліру бардюру на червоний і зменшення ширини зображення до 50 px'); \$("pic1").style.borderColor="red" ; \$("pic1").style.width="50px"; } </pre>
--	--

Індивідуальні завдання

Виконати завдання згідно з вашим варіантом попередньої лабораторної роботи. Доопрацювати вашу задачу з попередньої лабораторної роботи додаванням динамічної зміни перелічених властивостей (для яких об'єктів буде динамічно змінено властивості стилів обрати самостійно). При запуску alert необхідно написати подію, що відбулась і зміни, що ми очікуємо після закриття цього вікна alert. Виконати роботу згідно з вашим варіантом (Таблиця 10.2). Додати результат до сторінки з результатом рішення Задачі 10.2.

Таблиця 10.2. Умови індивідуальних задач

Варіант	подія	об'єкт	властивості, що треба динамічно налаштувати (Таблиця 10.1)
Варіант 1	onload onclick onmouseover	document Textarea	1, 11, 21, 31, 41, 51
Варіант 2	onunload onfocus onblur	document Textarea	2, 12, 22, 32, 42, 52
Варіант 3	onclick onmouseover onmouseout	Textarea Image	3, 13, 23, 33, 43, 53
Варіант 4	ondblclick onfocus onblur	Textarea Text	4, 14, 24, 34, 44, 54
Варіант 5	onmouseover onmouseout onchange	Image Select	5, 15, 25, 35, 45, 55
Варіант 6	onmouseout onclick onload	Image document	6, 16, 26, 36, 46, 56
Варіант 7	onfocus ondblclick onmouseover	Text Image	7, 17, 27, 37, 47, 57
Варіант 8	onblur onchange onmouseover	Text Select	8, 18, 28, 38, 48, 58
Варіант 9	onchange ondblclick onfocus	Select Textarea	9, 19, 29, 39, 49, 59
Варіант 10	ondblclick onfocus onblur	Textarea Image	10, 20, 30, 40, 50, 60

Список використаних джерел.

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Издательство «Питер», 2000, 672с.
2. W3schools. JavaScript Функції. URL:
https://schoolsw3.com/js/js_functions.php (дата звернення 25.05.2021)
3. W3schools. HTML 4 Entity Names. URL:
https://www.w3schools.com/charsets/ref_html_entities_4.asp (дата звернення 05.06.2021)
4. HTML5 book. CSS-шрифты. URL: <https://html5book.ru/css-shrifty> (дата звернення 28.05.21)
5. W3school. HTML DOM Events. URL:
https://www.w3schools.com/jsref/dom_obj_event.asp (звернення 25.05.21)
6. Лапінський В. В. Проблема вибору першої мови програмування - сьогоднішнє бачення. *Комп'ютер у школі та сім'ї*. №1, 2014, С. 14-17.
URL:http://lib.iitta.gov.ua/2168/1/перша_мова.pdf (звернення: 27.12.2020)
7. Щербаков Є.В., Щербакова М.Є. Аналіз класів мови програмування JavaScript. *Вісник Східноукраїнського національного університету імені Володимира Даля*. № 8 (238) 2017, С.90-93.
URL:<http://dspace.snu.edu.ua:8080/jspui/bitstream/123456789/1865/1/090-093.pdf> (дата звернення: 27.12.2020)
8. Городецька О. С. Методичні вказівки до самостійної роботи студентів з вивчення дисципліни «Інтернет технології та web-дизайн» для студентів спеціальності 172 Телекомунікації та радіотехніка. Вінниця. 2018, С. 25.
URL:
https://ecopy.posibnyky.vntu.edu.ua/txt/2018/Gorodezka_mv_srs_p038.pdf
(дата звернення: 11.09.2020).
9. Рухляда В.С., Карпенко М.Ю., Уфимцева В.Б. Методичні вказівки для виконання лабораторної роботи «Розробка персонального WEB-сайту» та самостійної підготовки студентів з дисципліни «Комп'ютерні мережі та телекомунікації». Харків: ХНАМГ, 2008, С. 24. URL:

http://eprints.kname.edu.ua/4997/1/Web-дизайн_22.04.08.pdf (дата звернення: 11.09.2020).

10. В. М. Базурін. Особливості навчання веб-програмування мовою javascript студентів-математиків. *Вісник Житомирського державного університету*. Випуск 1 (73) 2014, С. 79-83. URL: <http://eprints.zu.edu.ua/11082/1/15.pdf> (дата звернення: 11.09.2019).
11. Рамський Ю.С., Іваськів І.С., Ніколаєнко О.Ю. Вивчення Web-програмування у школі: навчальний посібник. Тернопіль, 2009, С. 200.
12. Дегтярьова Н.В. Методичні особливості навчання студентів розмітки блоків при вивченні таблиць каскадних стилів. *Фізико-математична освіта*. №1(11), 2017, С. 32-36. URL: <https://cyberleninka.ru/article/n/metodichni-osoblivosti-navchannya-studentiv-rozmitki-bloktiv-pri-vivchenni-tablits-kaskadnih-stiliv/viewer> (дата звернення: 11.09.2020).
13. Огурцов В. В., Гриньов Д. В., Щербаков О. В. Основи веб та веб-дизайн, програмування на боці клієнта. Лабораторний практикум, ХНЕУ ім. С. Кузнеця. 2015, С. 208. URL: [http://www.repository.hneu.edu.ua/bitstream/123456789/10442/1/2015 Огурцов В. В., Гринев Д. В., Щербаков О. В..pdf](http://www.repository.hneu.edu.ua/bitstream/123456789/10442/1/2015%20Огурцов%20В.%20Гринев%20Д.%20Щербаков%20О.%20В..pdf) (дата звернення: 11.09.2020).
14. Шувалова О.І. Web-програмування сторони серверу: методичні рекомендації. Одеса: Університет Ушинського, 2019, С. 95. URL: [http://dspace.pdpu.edu.ua/bitstream/123456789/2287/4/Shuvalova%2c%20Olg a%20%20d0%86gorivna.pdf](http://dspace.pdpu.edu.ua/bitstream/123456789/2287/4/Shuvalova%2c%20Olg%20a%20%20d0%86gorivna.pdf) (дата звернення: 05.06.2021).
15. Шувалова О.І. Web-програмування. Побудова Web-орієнтованої інформаційної системи: методичні рекомендації. Одеса: Університет Ушинського, 2019, С. 55. URL: http://dspace.pdpu.edu.ua/bitstream/123456789/4261/3/Web_2.pdf (дата звернення: 11.09.2020).

Додаток А. Хостинг мережі Інтернет на прикладі хостингу zzz.com.ua

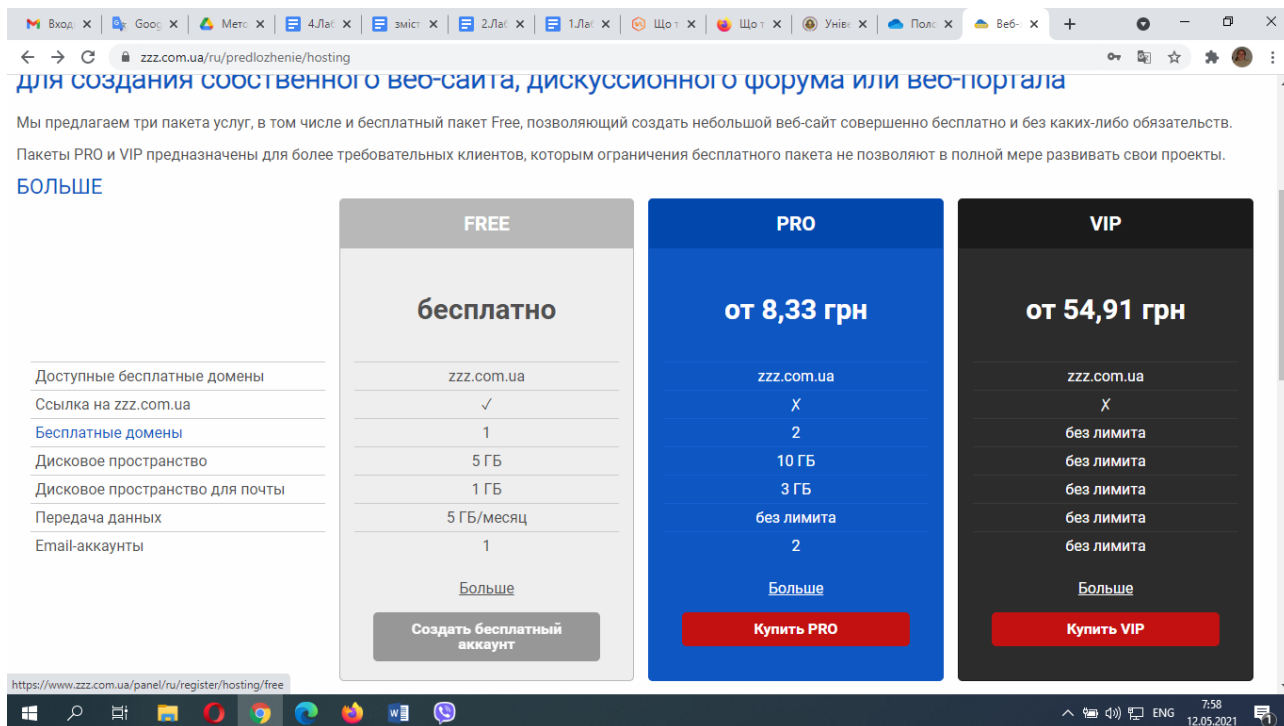


Рис. А.1. Реєстрація безкоштовного хостингу zzz.com.ua

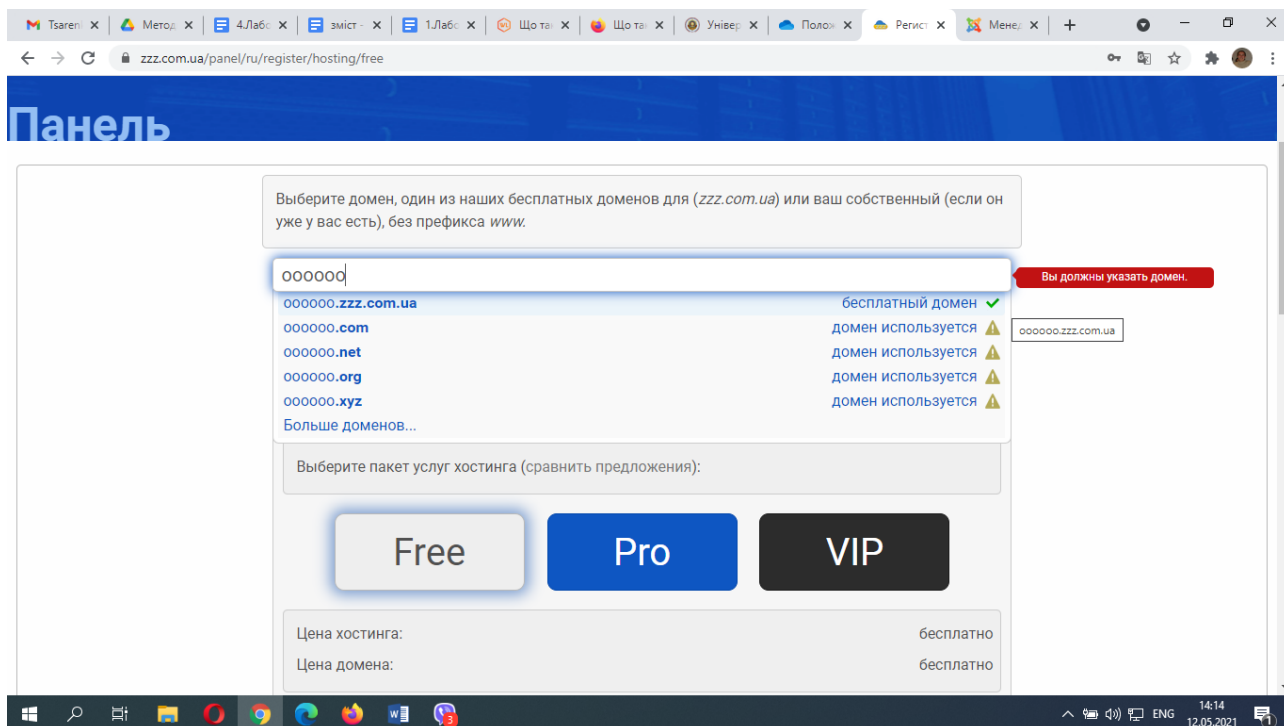


Рис. А.2. Реєстрація безкоштовного домену

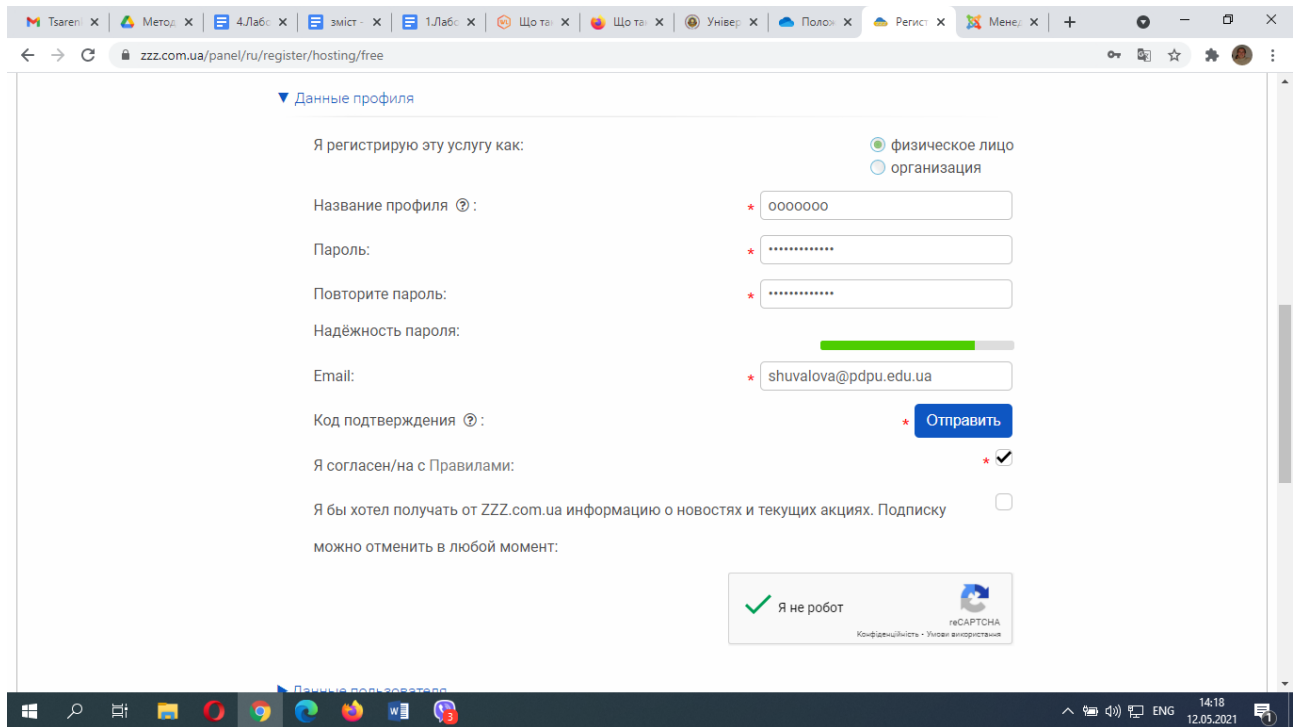


Рис. А.3. Заповнення форми реєстрації. Отримання коду підтвердження.
Введення коду, що закінчує реєстрацію

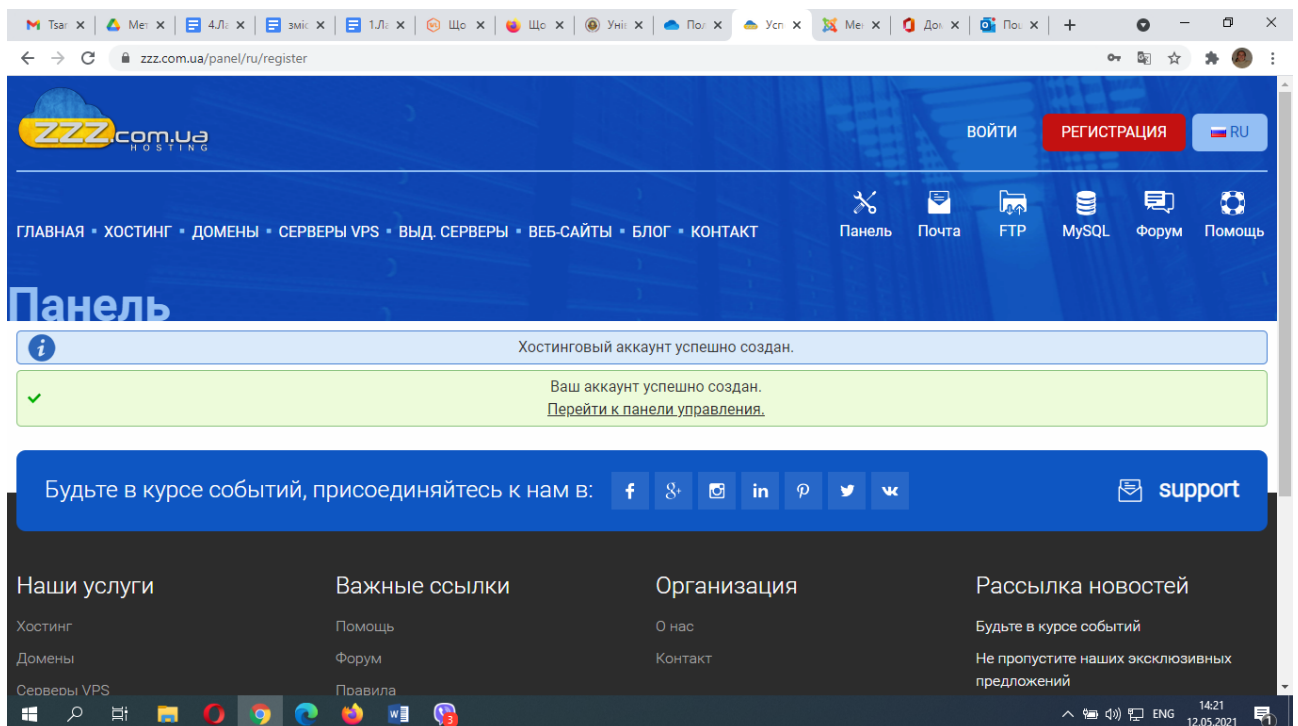


Рис. А.4. Завершення реєстрації

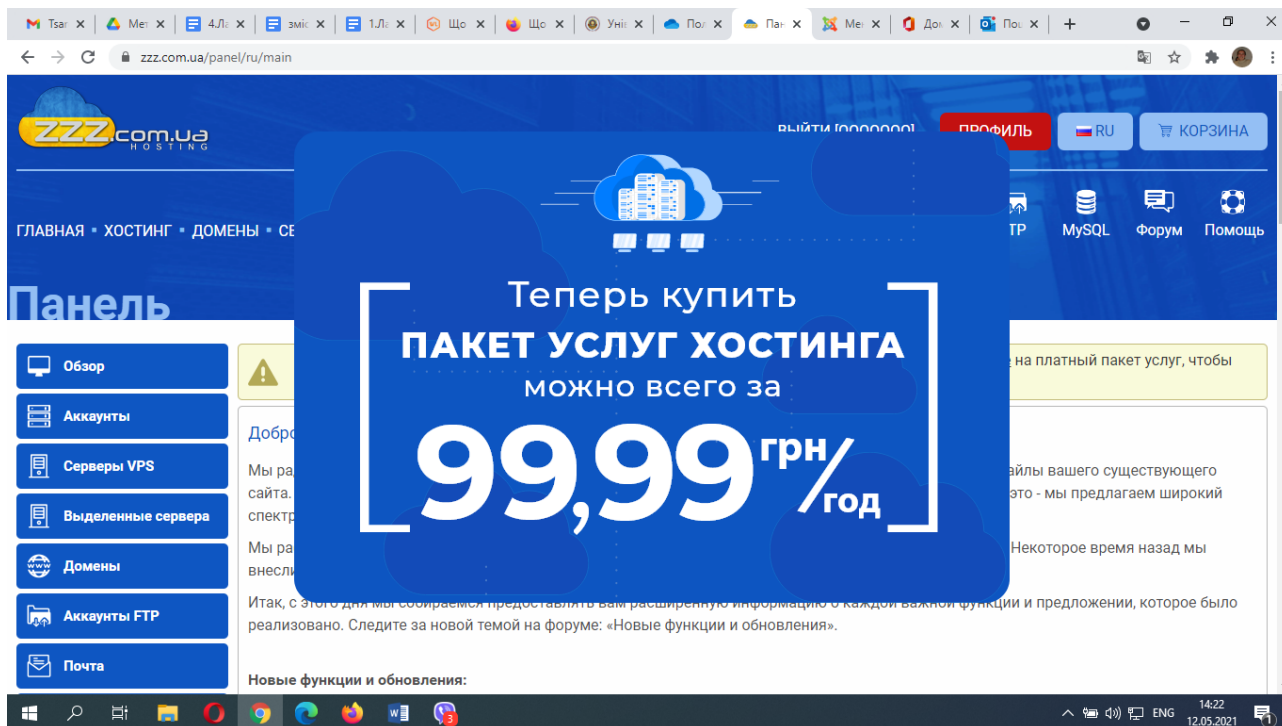


Рис. А.5. Вартість пакету послуг

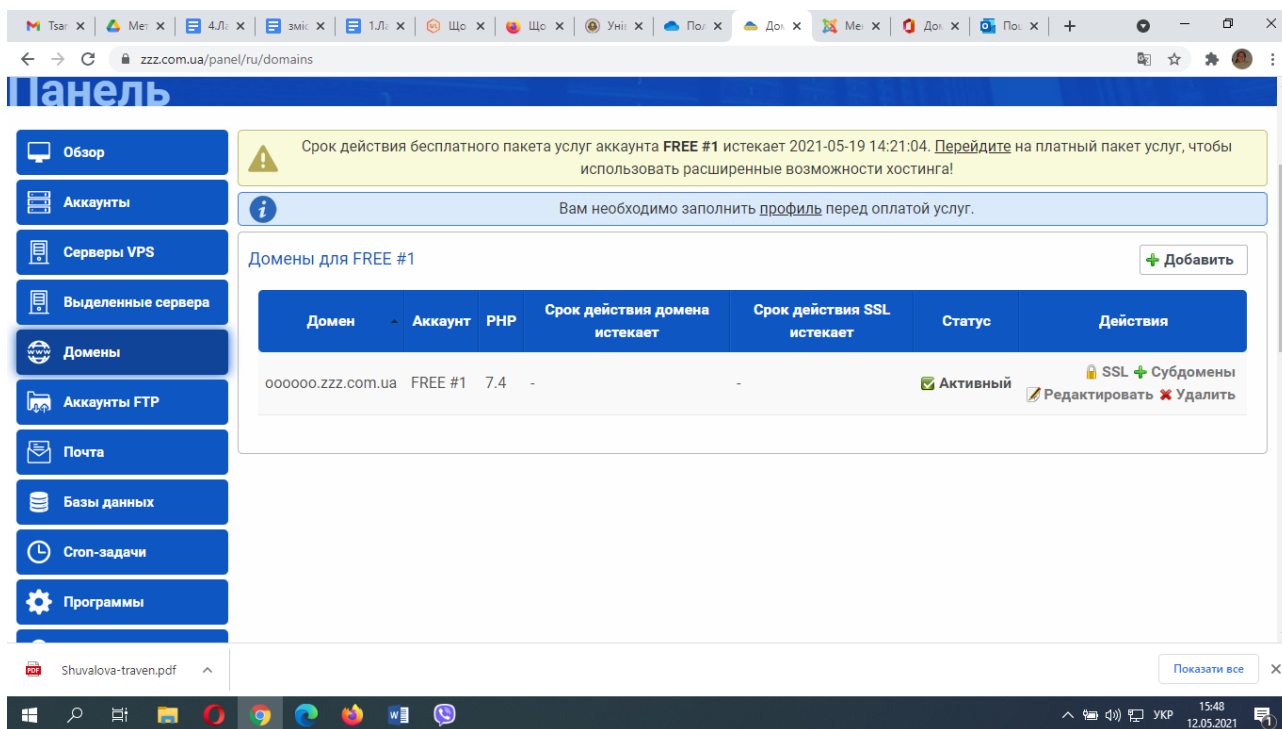


Рис. А.6. Панель управління хостингом. Заведення доменів і субдоменів

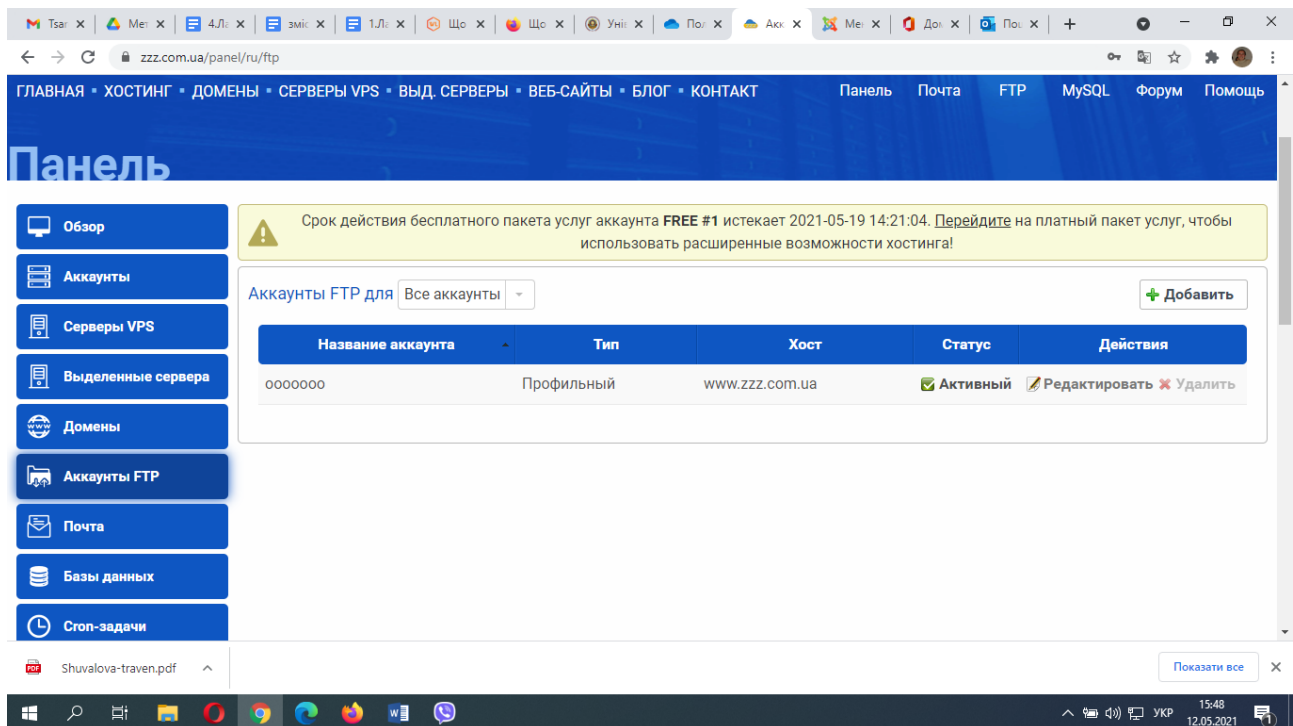


Рис. А.7. Заведення акаунтів FTP. За замовчанням після реєстрації вже є акаунт FTP з логіном і паролем входу на хостинг

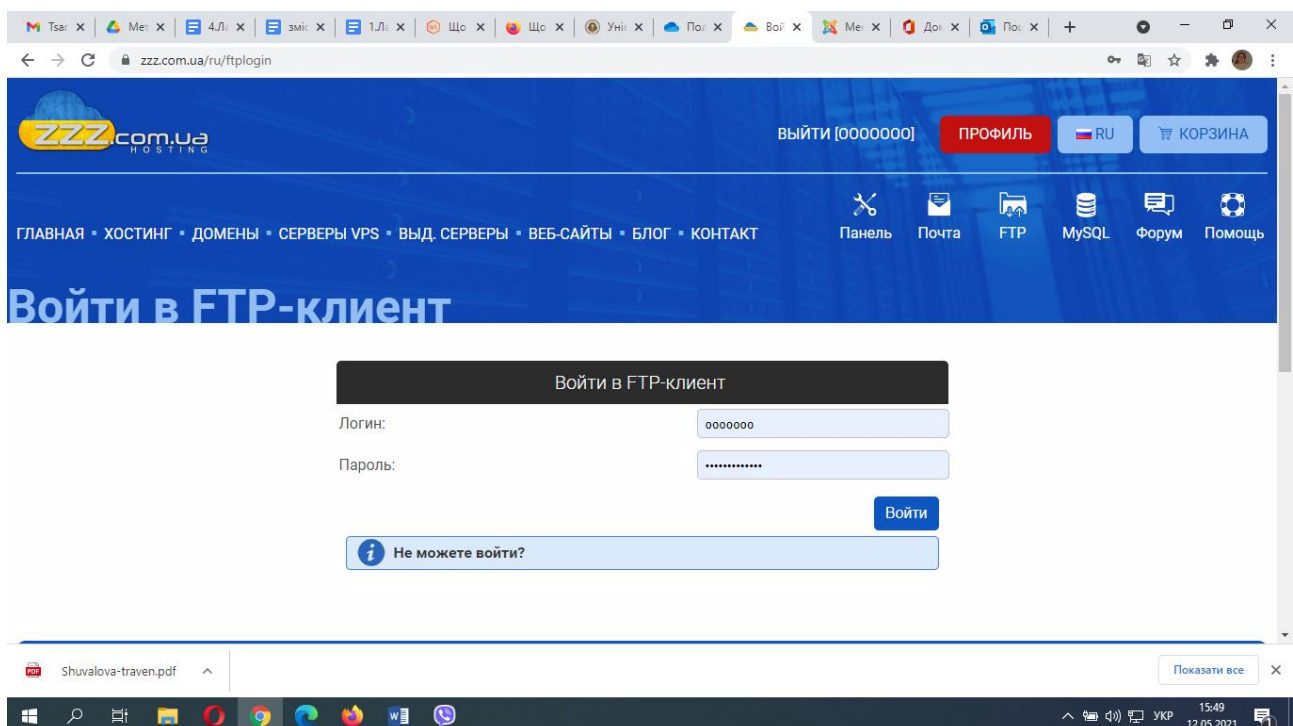


Рис. А.8. Перехід до гіпертекстового клієнта служби FTP (служби пересилки файлової інформації)

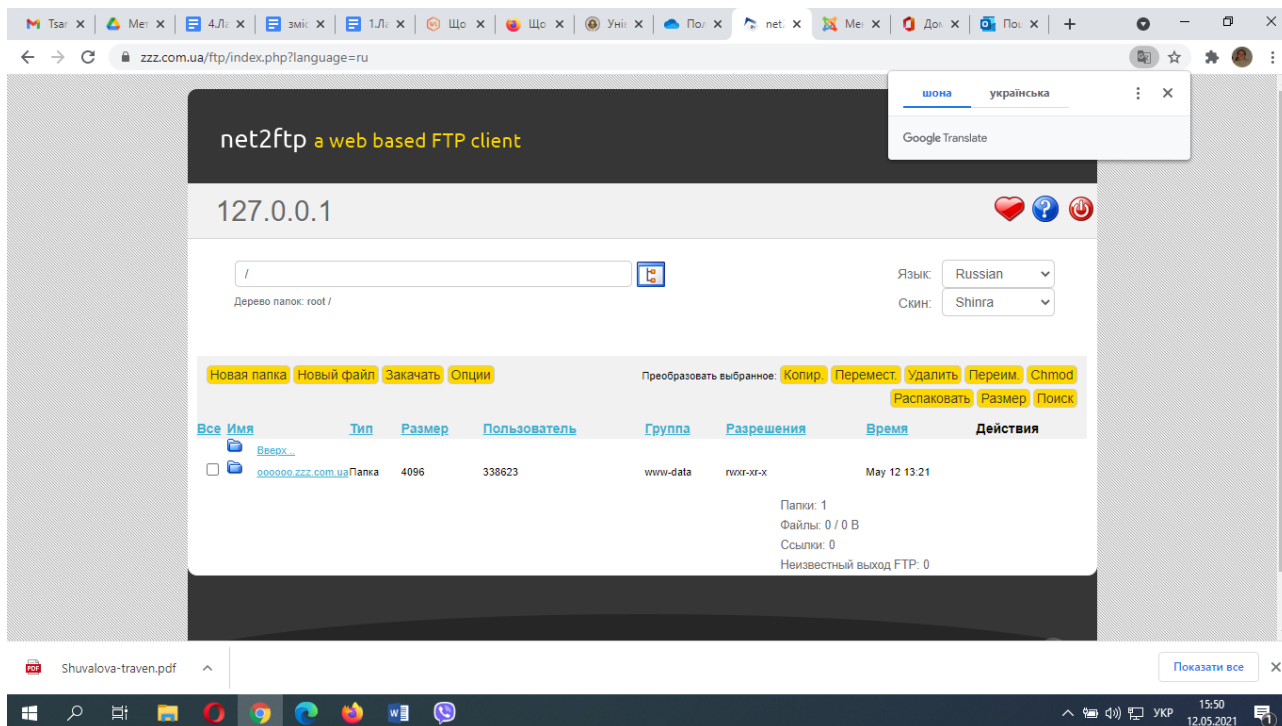


Рис. А.9. Для створення сайту, що відкривається з URL адреси входимо в папку з доменним ім'ям

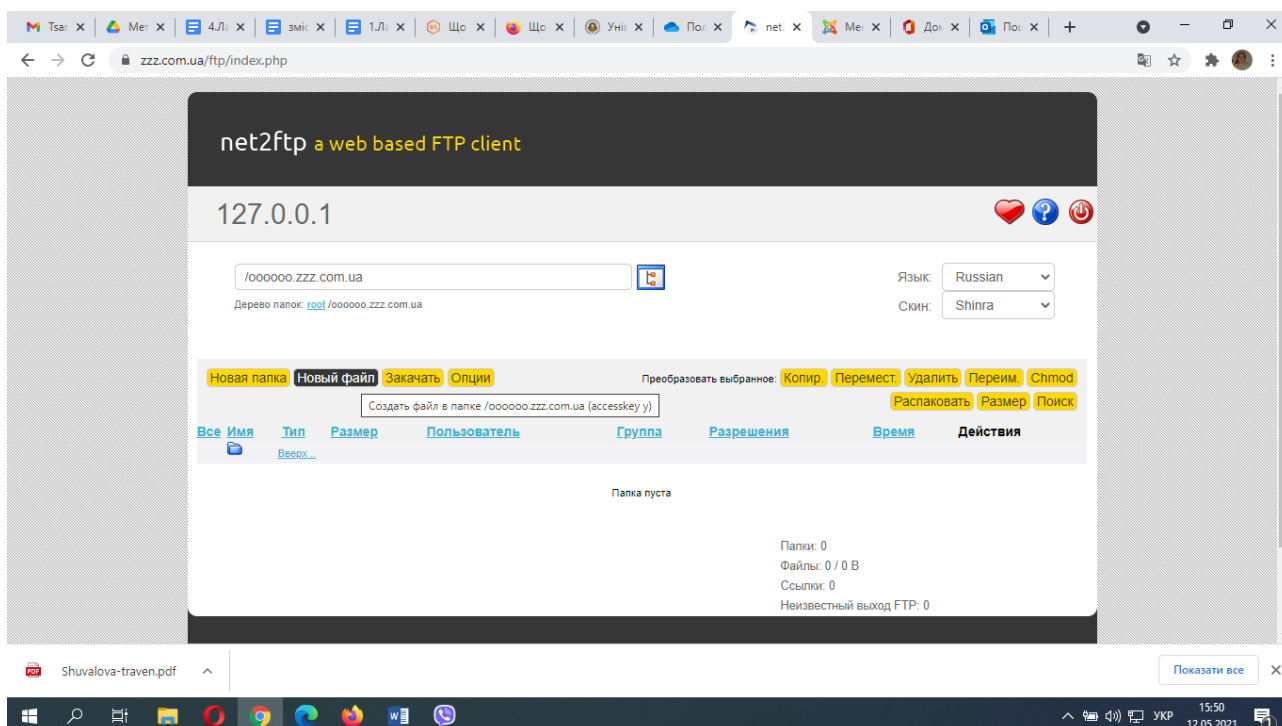


Рис. А.10. Переходимо в режим створення нового файла

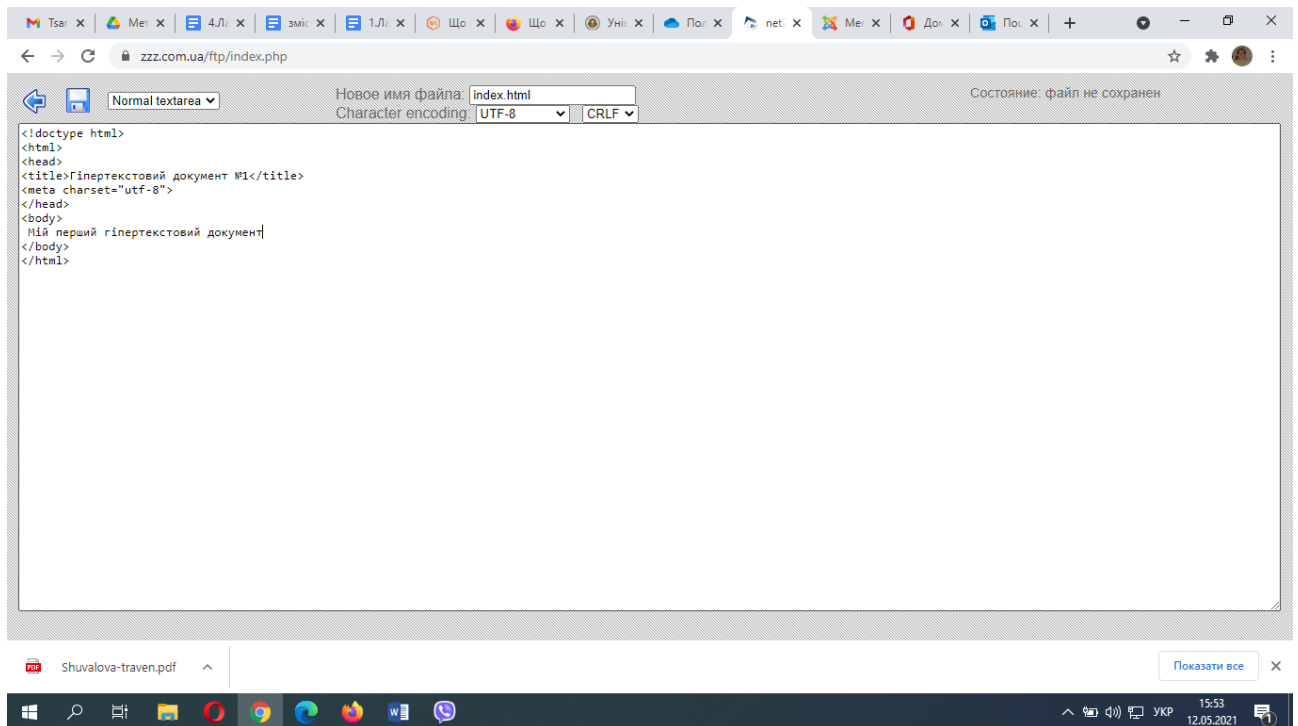


Рис. А.11. Створюємо файл index.html базової структури, кодування UTF-8

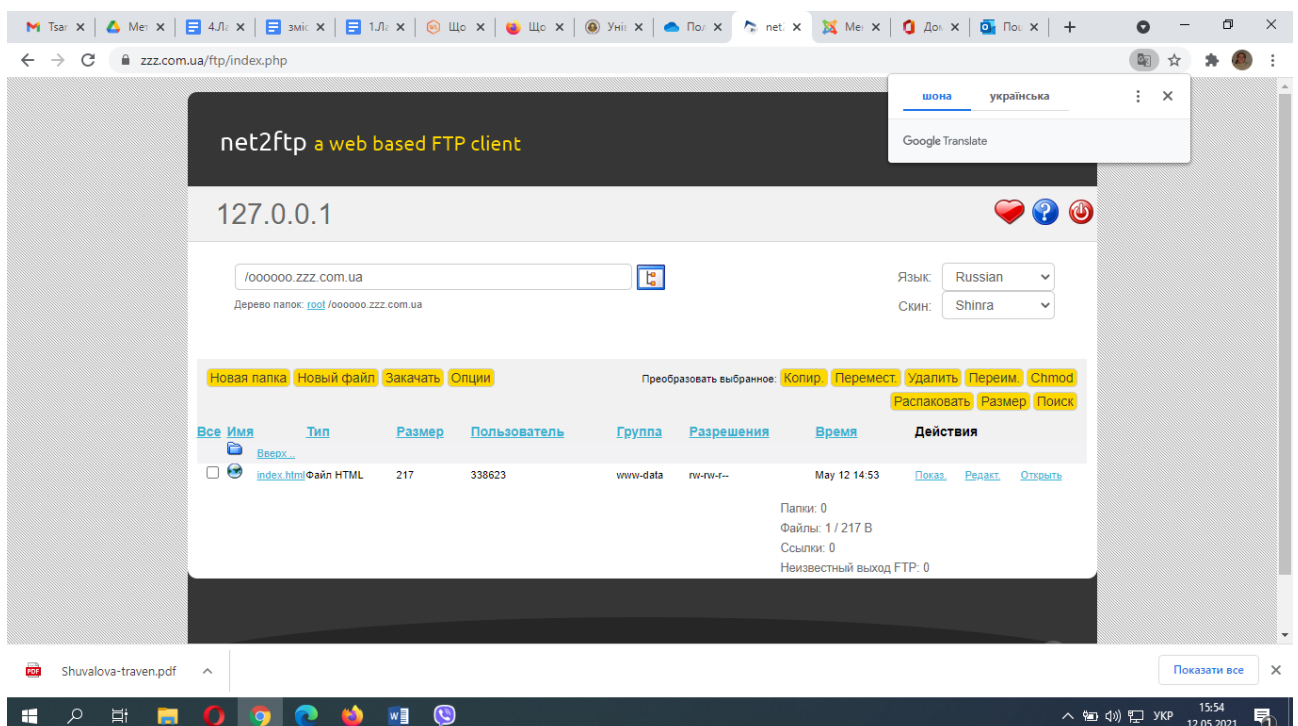


Рис. А.12. Файл index.html можна створити на локальному комп'ютері в простому текстовому редакторі і завантажити в папку (кнопка “закачать”)

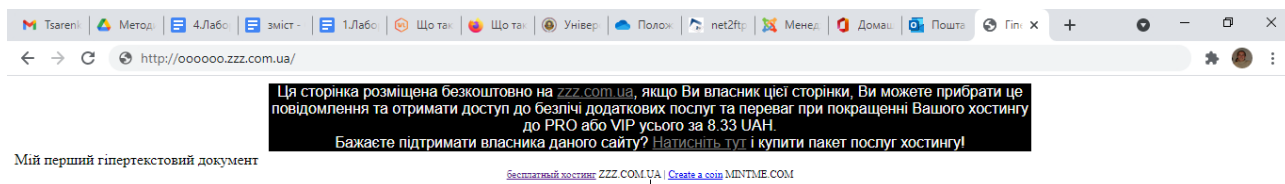


Рис. А.13. Перегляд результату з URL адреси

Додаток Б. Робота з GitHub як з хостингом.

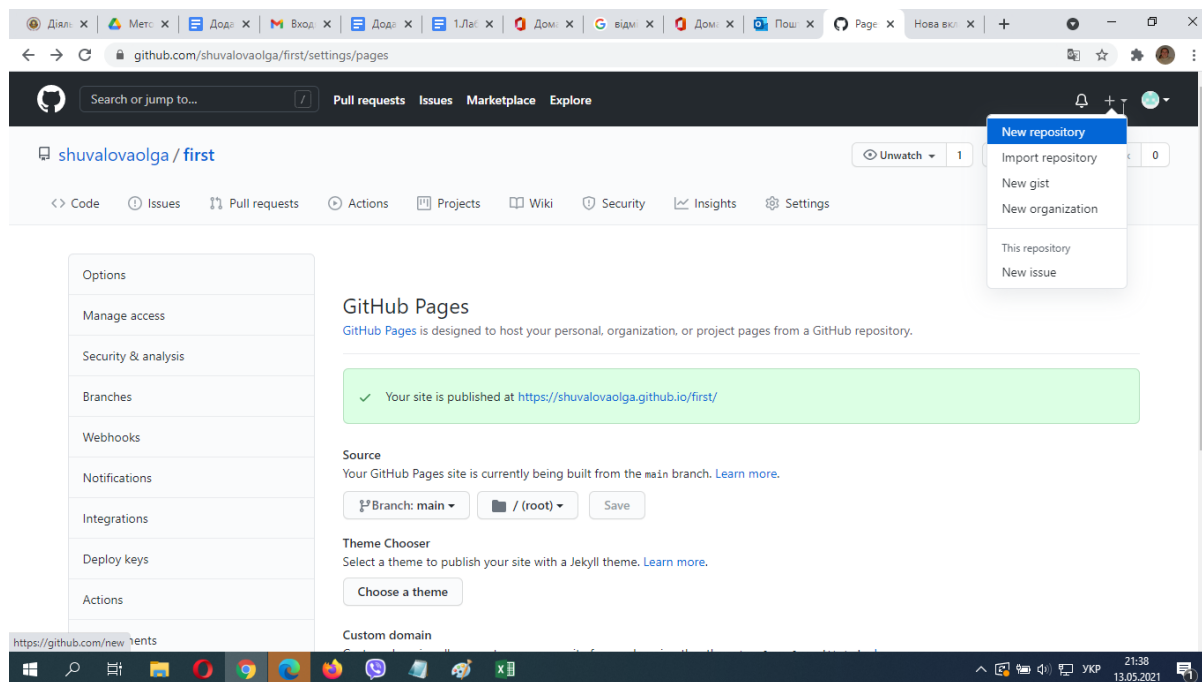


Рис. Б.1. Інтерфейс розділу “репозитарій” після входу в GitHub

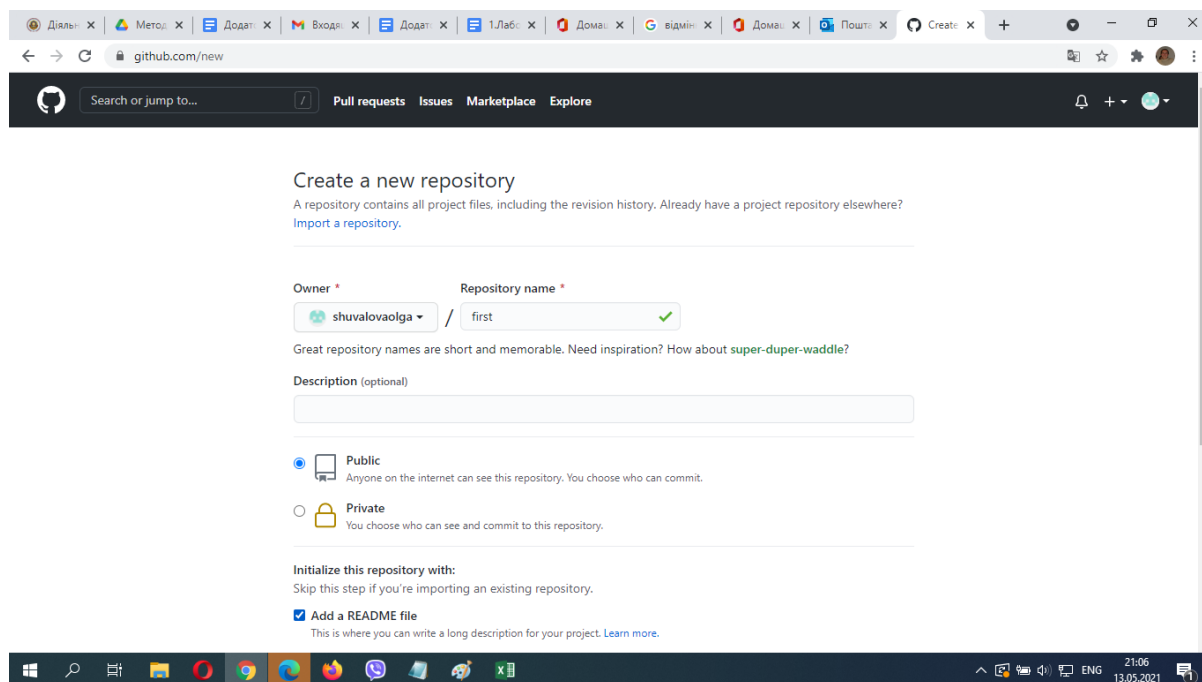


Рис. Б.2. Створення нового репозитарія. Ставимо галочку “Add a README file”

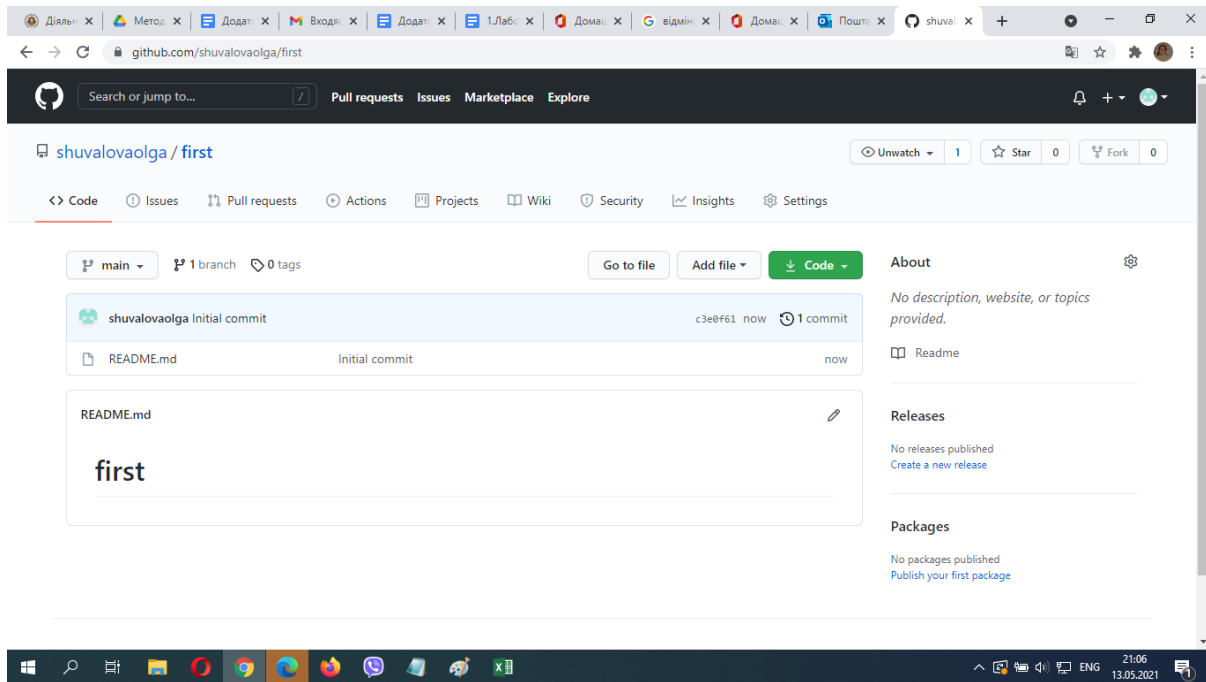


Рис. Б.3. Вигляд нового репозитарію

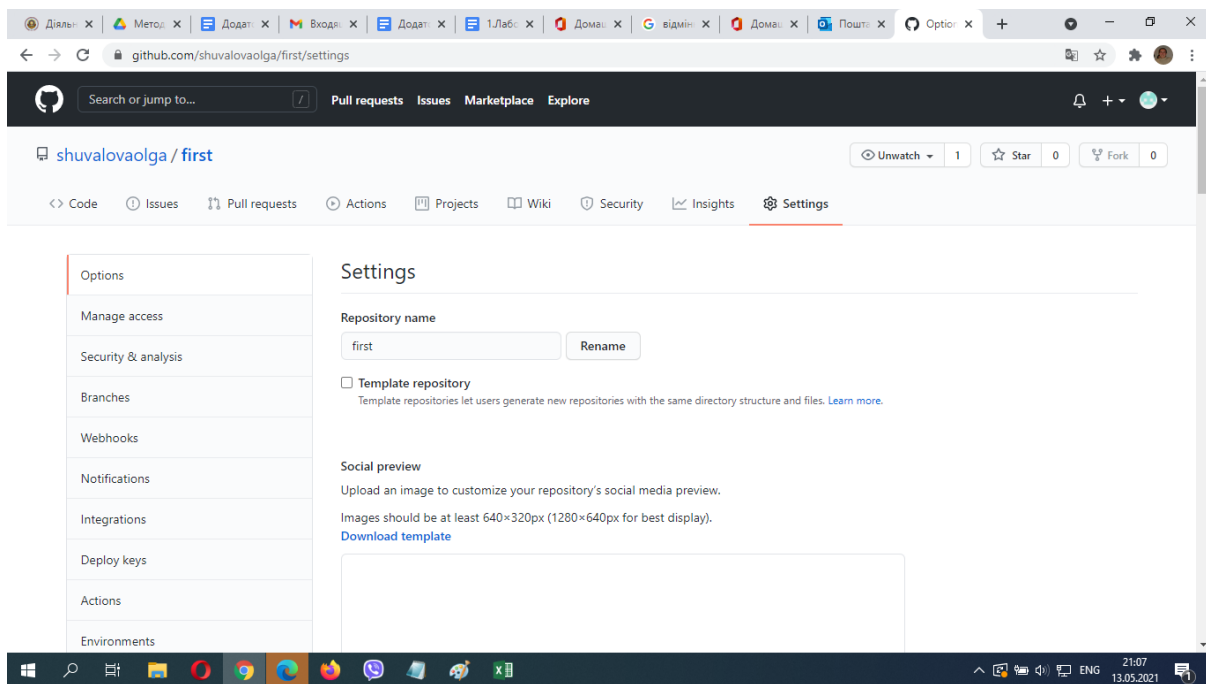


Рис. Б.4 Переходимо в налаштування Settings. Налаштовуємо URL адресу до репозитарію

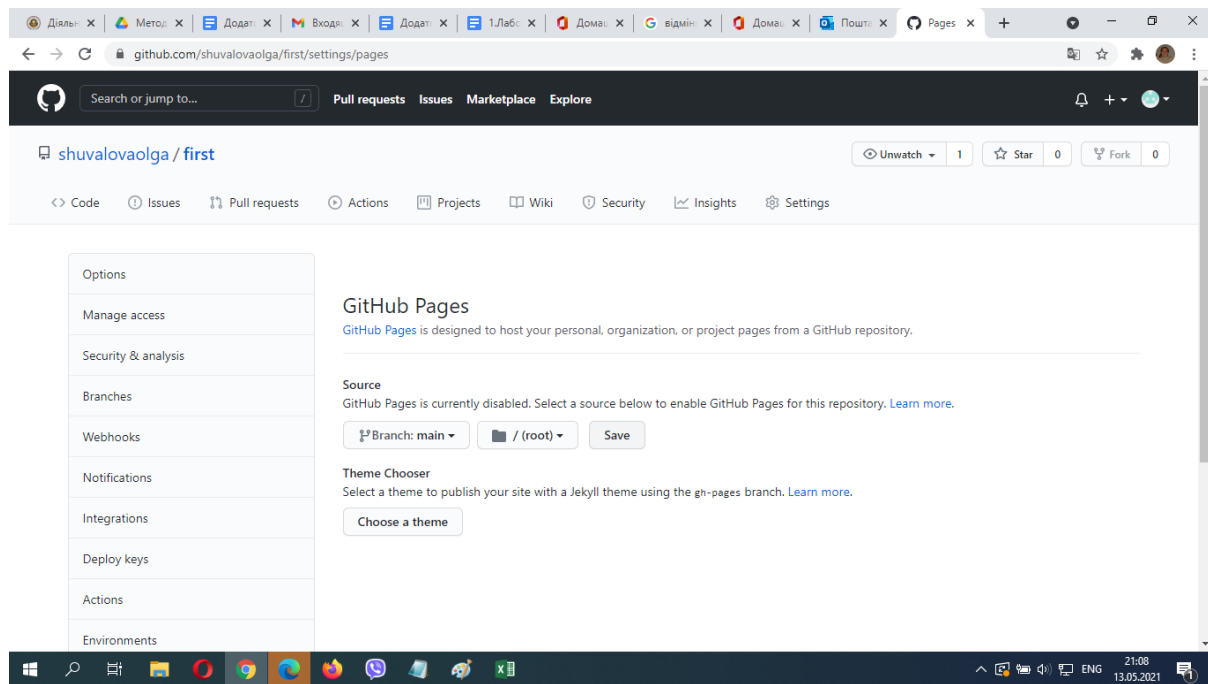


Рис. Б.5 Переходимо в розділ GitHub Pages і встановлюємо Branch: main

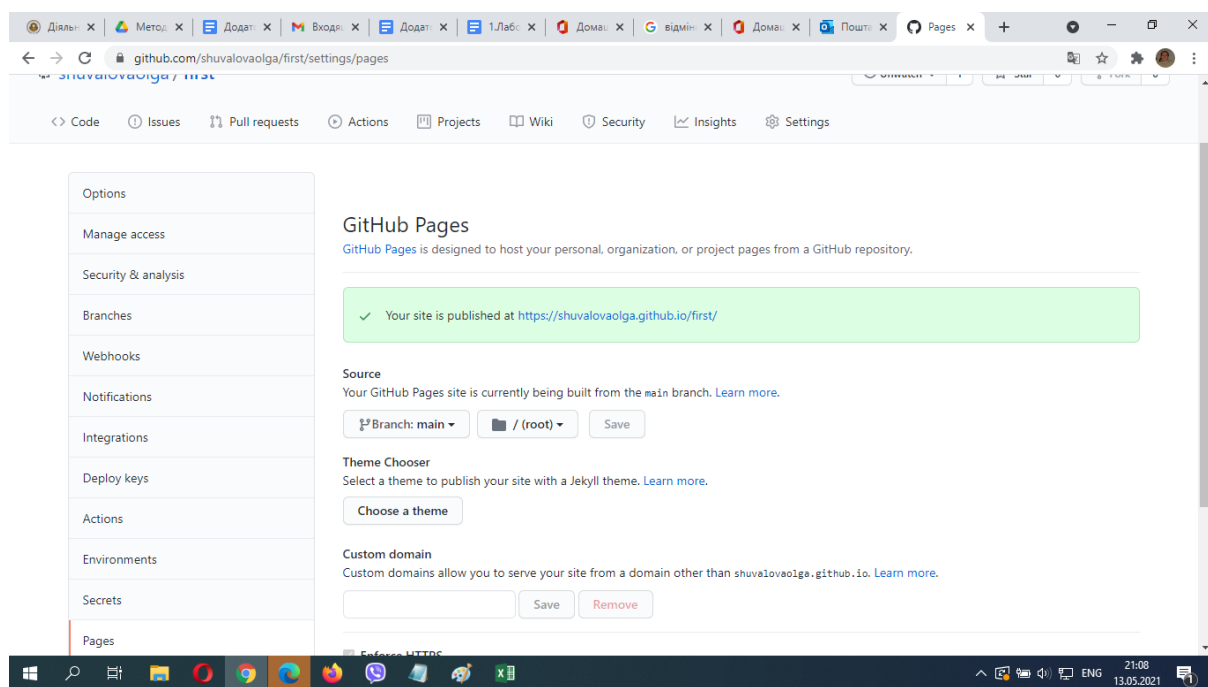


Рис. Б.6. Отримуємо URL адресу до репозитарію

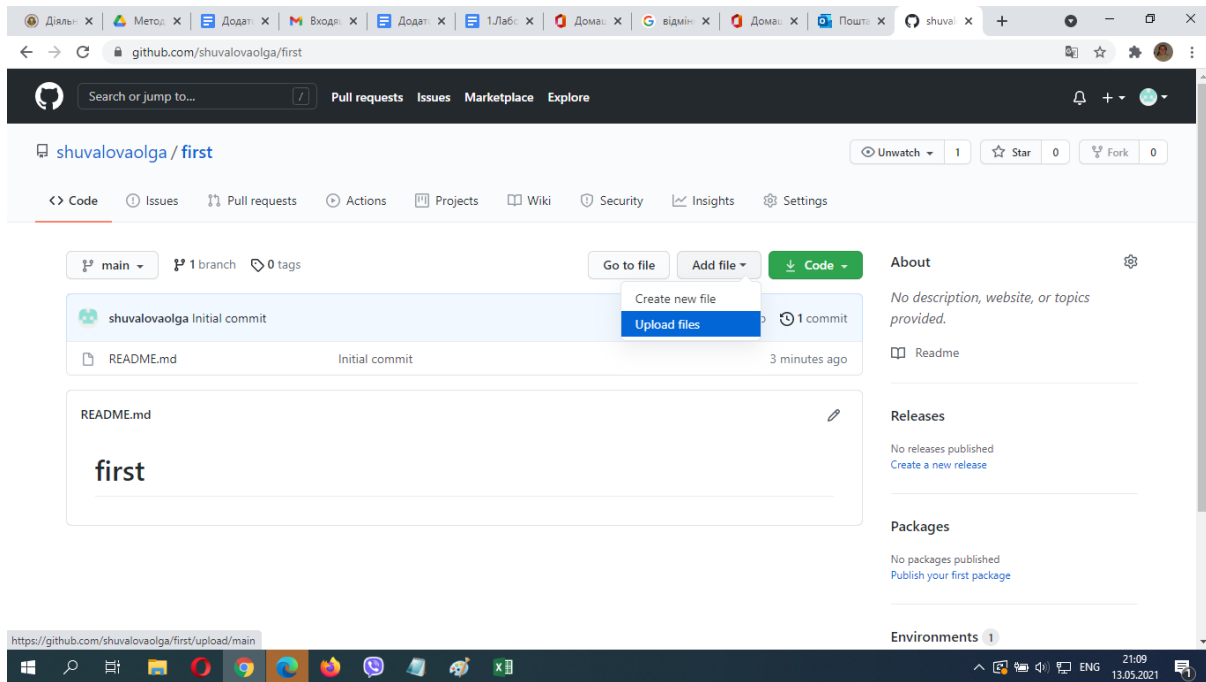


Рис. Б.7. Загружаємо з локального комп'ютера файл index.html

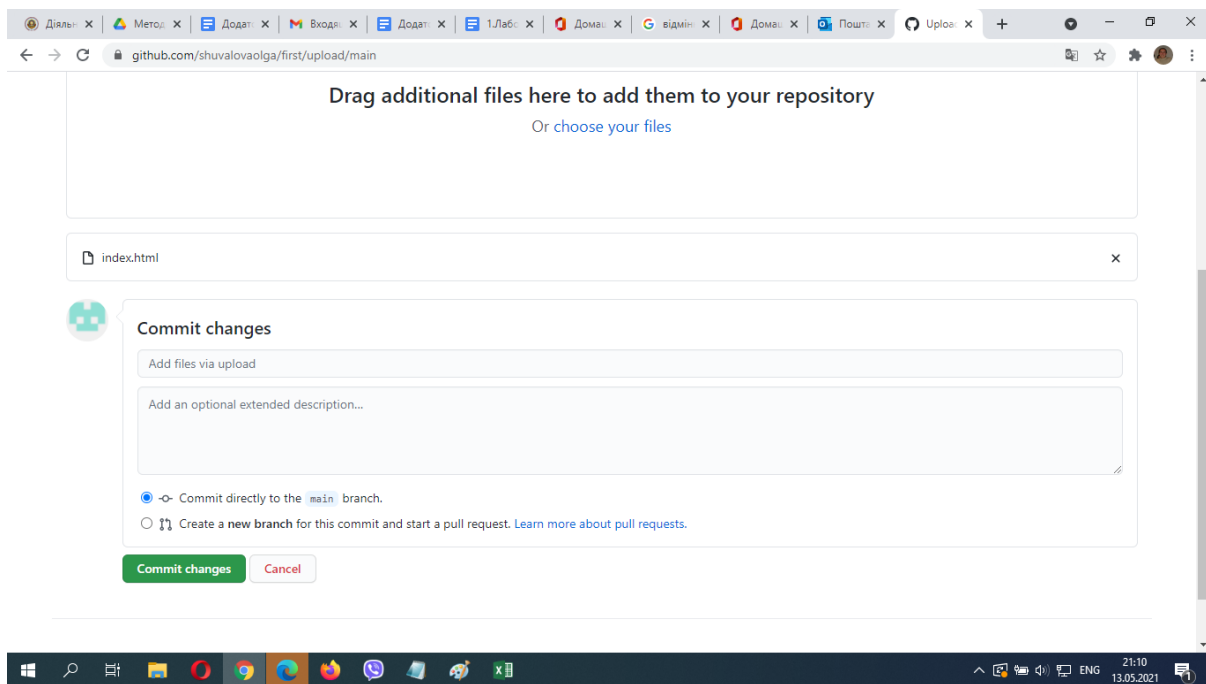


Рис. Б.8. Підтверджуємо завантаження через натискання “Commit changes”

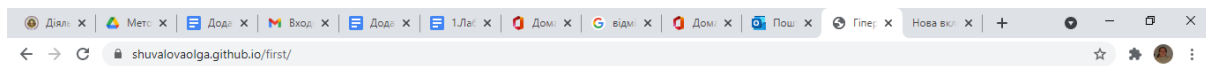


Рис. Б.9. За URL адресою переглядаємо index.html файл

Додаток В. Таблиці кодів символів гіпертекстового документа

Таблиця В.1. Кодування символів в амперсанд послідовностях

Char	Entity	Dec	Hex	Description
&	&	&	&	ampersand
<	<	<	<	less than
>	>	>	>	greater than
	 	 	 	no-break space = non-breaking space
¡	¡	¡	¡	inverted exclamation mark
¢	¢	¢	¢	cent sign
£	£	£	£	pound sign
¤	¤	¤	¤	currency sign
¥	¥	¥	¥	yen sign = yuan sign
	¦	¦	¦	broken bar = broken vertical bar
§	§	§	§	section sign
¨	¨	¨	¨	diaeresis = spacing diaeresis
©	©	©	©	copyright sign
^a	ª	ª	ª	feminine ordinal indicator
«	«	«	«	left-pointing double angle quotation mark = left pointing guillemet