

Присяжнюк О.В.,
Резіна О.В.

Основи web-розробки (HTML, CSS, JavaScript)



Кропивницький
2021

ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ВИННИЧЕНКА
Кафедра інформатики та інформаційних технологій

МЕТОДИЧНІ ВКАЗІВКИ І ЗАВДАННЯ
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ ІНФОРМАТИКИ

Частина 2. ОСНОВИ WEB-РОЗРОБКИ

Навчально-методичний посібник
для студентів напрямку підготовки 014.04
середня освіта (Математика)
спеціальності «Математика та інформатика»

Кропивницький – 2021

Методичні вказівки і завдання до лабораторних робіт з курсу інформатики. Частина 2. Основи web-розробки: Навч.-метод. посіб. / Автори-укладачі: О.В. Присяжнюк, О.В. Рєзіна – Кропивницький: ЦДПУ імені В. Винниченка, 2021. – 42 с.

Автори-укладачі: **Присяжнюк О.В.**, доцент кафедри інформатики та інформаційних технологій, канд. техн. наук, **Рєзіна О.В.**, доцент кафедри інформатики та інформаційних технологій, канд. пед. наук.

Рецензенти:

Паращук С.Д., канд. фіз.-мат. наук, доцент, завідувач кафедри інформатики та інформаційних технологій

Шлянчак С.О., канд. пед. наук доцент кафедри інформатики та інформаційних технологій

Рекомендовано рішенням Методичної ради Центральноукраїнського державного педагогічного університету імені Володимира Винниченка (протокол № 2 від 27 січня 2021 року)

Навчально-методичний посібник призначено для використання в навчальному процесі при вивченні курсу «Інформатика. Основи веб-розробки». У посібнику розглянуто питання, пов'язані з практичними аспектами технологій розробки сучасних web-додатків, зокрема можливостями клієнтської розробки на основі HTML5, CSS та JavaScript. Посібник містить вказівки до виконання лабораторних робіт, практичні та індивідуальні завдання з верстання веб сторінок за допомогою HTML, CSS та JavaScript, контрольні питання та перелік рекомендованої літератури. Методичні вказівки і завдання до лабораторних робіт з курсу інформатики можуть бути використані при організації самостійної роботи студентів.

Навчальний посібник призначено для студентів спеціальності «Математика та інформатика», а також може бути корисним для широкого кола фахівців, що займаються проблемами програмування для Інтернету.

ЗМІСТ

ВСТУП	4
Теми лабораторних робіт, що розглядаються та оцінювання знань.....	5
Шкала оцінювання: національна та ECTS.....	7
Тема 1: Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS	8
Лабораторна робота №1	8
Лабораторна робота №2	11
Лабораторна робота № 3	14
Тема 2. Блокова модель в HTML-документах. Основи проектування та верстки WEB-сторінок.....	16
Лабораторна робота №4	16
Лабораторна робота №5	18
Лабораторна робота 6	20
Тема 3. Web-програмування на боці клієнта. Основи Javascript.....	25
Лабораторна робота 7	25
Лабораторна робота 8	28
Лабораторна робота 9	33
Тема 4. Використання бібліотек JavaScript для розробки web-сторінок. Бібліотека jQuery	36
Лабораторна робота 10	36
Питання до екзамену.....	38
Список скорочень і спеціальних термінів	40
Рекомендована література.....	42

ВСТУП

Дисципліна «Інформатика. Основи web-розробки» є важливою складовою навчального процесу з підготовки кваліфікованих фахівців з інформаційних технологій. Метою викладання дисципліни є формування системи знань з основ web-технологій: складу, структури, принципів реалізації технології «клієнт-сервер», основ мов web-розмітки (HTML, CSS) та базових основ мови програмування JavaScript.

Згідно з вимогами освітньо-професійної програми в результаті виконання лабораторних робіт з дисципліни у студента формуються такі *компетентності*:

- здатність розв'язувати задачі шкільного курсу інформатики та математики різного рівня складності та формувати відповідні вміння в учнів;
- здатність добирати та використовувати сучасні інформаційно-комунікаційні технології в освітньому процесі та в позакласній роботі, аналізувати й оцінювати доцільність й ефективність їх застосування;
- здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів комп'ютерних наук, інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Отримані знання дозволяють на професійній основі проектувати і програмувати вміст HTML-документів, формувати інтерфейс майбутніх розробок, працювати із шрифтами, стилями, таблицями, списками, зображеннями, розробляти структурний макет сайту на принципах web-дизайну, а також створювати прості сценарії обробки подій та працювати з формами в JavaScript.

Навчально-методичний посібник для кожної лабораторної роботи містить тему і мету роботи, завдання та вказівки до його виконання, посилання на відповідні інтернет-ресурси з довідковими відомостями і тренувальними завданнями. В кінці кожної роботи даються контрольні питання для самоперевірки з виконуваної теми. Усі теми послідовні і тісно пов'язані між собою.

У процесі підготовки навчального-методичного посібника були використані джерела, перелік яких наведений у літературі.

Посібник призначений для студентів спеціальності 014.04 Середня освіта (Математика) з додатковою спеціальністю «Інформатика» фізико-математичного факультету.

Теми лабораторних робіт, що розглядаються та оцінювання знань

Методичні вказівки містять 10 лабораторних робіт за наступними темами:

Теми	Лабораторні роботи
1. Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS	№ 1 - № 3
2. Блокова модель в HTML-документах. Основи проектування та верстки веб-сторінок	№ 4 - № 6
3. Web-програмування на боці клієнта. Основи Javascript.	№ 7 - № 9
4. Використання бібліотек javascript для розробки web-сторінок. Бібліотека jQuery	№ 10

Форма підсумкового контролю: екзамен.

Максимальну кількість балів студент може одержати у випадку відвідування всіх лекцій, лабораторних занять, виконання всіх завдань з лабораторних робіт, що демонструється викладачу у встановлений термін проходження контролю.

На початку кожної лабораторної роботи вказано її тему, мету, що необхідно знати і вміти. Надано вказівки до матеріалу теми у вигляді відповідних посилань до онлайн підручника, разом із тренувальними завданнями. Вивчивши теоретичний матеріал теми, уважно прочитавши завдання, яке розташоване після теоретичного матеріалу, необхідно вибрати свою тематику, згідно варіанту (номер варіанту узгоджується з викладачем). Для успішного захисту лабораторної роботи необхідно виконати всі запропоновані завдання, а також відповісти на контрольні питання, що знаходяться вкінці кожної роботи.

Для виконання завдань знадобиться текстовий редактор, наприклад Notepad++ або Visual Code та будь-який Інтернет-браузер.

При виборі редактора необхідно враховувати зручність, функціональність, наскільки актуальна остання версія і чи триває підтримка.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Студенти, що вчасно виконують всі завдання та приймають активну участь у конференціях з тематикою дисципліни «Основи Web-розробки» отримують додаткові бали.

Тема 1: Основи Web-програмування. Розробка Web-сторінок за допомогою HTML та CSS

Лабораторна робота №1

Тема: Мова HTML. Використання текстових блоків. Логічне та стильове форматування тексту.

Мета: Формування вмінь використання мови HTML для розмітки Web-документів. Формування вмінь стильового оформлення тексту документа засобами технології CSS.

Питання для вивчення:

1. Мова HTML. Основні поняття: теги, елементи, атрибути.
2. Структура HTML-документа.
3. Теги визначення головних структурних особливостей HTML-документа.
4. Теги логічного та фізичного форматування тексту.
5. Специфікація HTML.
6. Декларація DOCTYPE, її компоненти.
7. Варіанти декларації DOCTYPE.
8. Поняття технології CSS.
9. Способи застосування стилів.
10. Структура правила CSS.
11. Основні параметри шрифту та тексту в CSS.

Вказівки для ознайомлення із матеріалом та завдання до виконання

1. Ознайомитися із середовищами Notepad++ та Visual Code.
2. Ознайомитися зі списком елементів HTML5 на сторінці <https://www.w3.org/TR/html-markup/elements-by-function.html>
3. HTML Headings. З'ясувати особливості використання тегів <h1> - <h6> та <hr>. Для цього перейти на сторінку http://www.w3schools.com/html/html_headings.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-4.
4. HTML Paragraphs. З'ясувати особливості використання тегів <p>,
 та <pre>. Для цього перейти на сторінку http://www.w3schools.com/html/html_paragraphs.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-4.

5. Text Formatting. З'ясувати особливості використання тегів

- - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Для цього перейти на сторінку http://www.w3schools.com/html/html_formatting.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-5.

6. Quotation. З'ясувати особливості використання тегів

- <abbr> - an abbreviation or acronym
- <address> - contact information for the author/owner of a document
- <bdo> - the text direction
- <blockquote> - a section that is quoted from another source
- <cite> - the title of a work
- <q> - a short inline quotation

Для цього перейти на сторінку http://www.w3schools.com/html/html_quotation_elements.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-4.

7. HTML Styles. З'ясувати особливості застосування стилів (використання атрибуту style). Для цього перейти на сторінку http://www.w3schools.com/html/html_styles.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-6.

8. HTML Color Names. З'ясувати особливості застосування кольорів. Для цього перейти на сторінку http://www.w3schools.com/html/html_colors.asp розглянути приклади.

9. Styling HTML with CSS. З'ясувати особливості застосування стилів. Для цього перейти на сторінку http://www.w3schools.com/html/html_css.asp розглянути приклади, наведені у розділах Inline CSS, Internal CSS, External CSS, CSS Fonts. У розділі Test Yourself with Exercises! виконати вправи 1-4.

10. CSS Fonts. З'ясувати особливості застосування стилів для шрифту. Для цього перейти на сторінку

http://www.w3schools.com/css/css_font.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-5.

11. CSS Text. З'ясувати особливості застосування стилів для тексту. Для цього перейти на сторінку http://www.w3schools.com/css/css_text.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-2.
12. Створити HTML-документ, в якому засобами структурного та логічного форматування презентувати свою курсову роботу. Документ обов'язково повинен містити:
 - тему курсової роботи;
 - зміст курсової роботи;
 - мету написання курсової роботи;
 - актуальність теми та завдання дослідження;
 - список ключових слів;
 - прізвище, ім'я та по-батькові автора роботи, номер групи.
13. Засобами технології CSS здійснити стильове оформлення шрифту і тексту (використати властивості Font та Text).

Контрольні запитання

1. Що таке мова HTML?
2. У мові HTML реалізовані традиційні алгоритмічні структури розгалуження і повторення. Так чи ні?
3. Коди, що застосовується у стандарті HTML, називаються _____.
4. Що таке елемент мови HTML?
5. Атрибути задають для елементів _____.
6. З чого складається специфікація атрибута?
7. Які текстові редактори для роботи з HTML ви знаєте?
8. Яка структура HTML-документа?
9. Яким чином здійснюється взаємодія між HTML-документом і сервером?
10. Для чого призначений тег <BODY>?
11. Для чого призначений тег <TITLE>?
12. Для чого використовують теги <H1> ... <H6>?
13. Для чого призначене логічне форматування вихідного тексту веб-документа?
14. Приклади тегів логічного форматування.
15. Для чого призначене фізичне форматування вихідного тексту веб-документа?
16. Приклади тегів фізичного форматування.

17. Що таке специфікація HTML? Для чого вона призначена?
18. Що таке декларація DOCTYPE? Які її компоненти та варіанти?
19. Що таке технології CSS? Яке її призначення?
20. Способи застосування стилів.
21. Структура правила CSS.
22. Якими способами можна додати інформацію про стилі у документ HTML?

Лабораторна робота №2

Тема: Селектори-класи. Списки в HTML-документі. Використання графіки.

Мета: Формування вмінь використання селекторів-класів у HTML-документі; створення та оформлення списків; використання графічних зображень.

Питання для вивчення:

1. Поняття селектора-класу.
2. Робота зі списками.
3. Формати графічних зображень, які використовуються в інтернеті.
4. Включення графіки до HTML-документа.
5. Використання тегів <figure> та <figcaption>.
6. Використання закруглених зображень.

Вказівки для ознайомлення із матеріалом та завдання до виконання

1. CSS .class Selector. З'ясувати особливості застосування селекторів-класів. Для цього перейти на сторінку http://www.w3schools.com/cssref/sel_class.asp розглянути приклади.
2. У документі, що створений під час виконання лабораторної роботи №1 (титольна сторінка курсової роботи), описати та використати два селектори-класи (на вибір).
3. Створити HTML-документ, в якому відобразити список термінів курсової роботи (5 пунктів). Використати теги списку означень. Написати стиль для елемента **dt**. Ім'я файлу – **def_list.htm**.
4. CSS list-style-type Property. З'ясувати особливості вибору типу маркера або нумератора в списках (застосування властивості list-style-type). Для цього перейти на сторінку

http://www.w3schools.com/cssref/pr_list-style-type.asp розглянути приклади.

5. Створити HTML-документ, в якому відобразити список літератури, використаної у курсовій роботі (5 пунктів). Список повинен відповідати правилам оформлення літератури для курсових робіт. Використати теги нумерованого списку. Ім'я файла – **liter_list.htm**. Уміти змінювати вигляд нумератора засобами CSS.
6. HTML Images. З'ясувати особливості використання графіки. Для цього перейти на сторінку http://www.w3schools.com/html/html_images.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1, 2, 4, 5, 6.
7. До титульної сторінки курсової роботи додати малюнок. Засобами CSS визначити його розмір (за необхідності) та розташування на екрані.
8. CSS Backgrounds. З'ясувати особливості додавання фону до HTML-документа. Для цього перейти на сторінку http://www.w3schools.com/css/css_background.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-5.
9. До титульної сторінки курсової роботи додати зображення як фон.
10. Створити HTML-документ, в якому презентувати автора курсової роботи. Ім'я файла – **Прізвище.htm**.

Цей документ обов'язково повинен містити:

- різноманітне стильове оформлення шрифту засобами CSS;
- маркірований список (уміти змінювати вигляд маркера засобами CSS);
- адресу автора з використанням тегу ADDRESS;
- фотокартку автора (притиснути зображення до правого краю сторінки).

11. HTML <figure> Tag. З'ясувати особливості використання тегу <figure>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_figure.asp розглянути приклади.
12. HTML <figcaption> Tag. З'ясувати особливості використання тегу <figcaption>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_figcaption.asp розглянути приклади.
13. У середовищі Notepad++ або Visual Code набрати програмний код:

```
<!DOCTYPE html>  
<html>
```

```

<body>
<figure>
  
  
  
  <figcaption>Australian Birds. From left to right,
    Kookburra, Pelican and Rainbow Lorikeet.</figcaption>
</figure>

</body>
</html>

```

Подивитися на результат, зробити висновки.

14. До однієї зі створених веб-сторінок додати галерею малюнків з підписом (використати теги <figure> та <figcaption>).
15. CSS Images. З'ясувати особливості використання закруглених зображень. Для цього перейти на сторінку http://www.w3schools.com/css/css3_images.asp розглянути приклади щодо застосування властивості border-radius.
16. Застосувати властивість border-radius до фотокартки автора (див. завд. №10).

Контрольні запитання

1. Як використовується CSS?
2. У яких випадках який спосіб застосування CSS кращий?
3. Який синтаксис визначення стилю?
4. Що таке селектор-клас? У яких випадках він використовується?
5. Як у HTML-документі створити маркірований список?
6. Як у HTML-документі створити нумерований список?
7. Як у HTML-документі створити список означень?
8. Що таке властивість list-style-type? Які її значення?
9. Як зробити малюнок маркером списку?
10. Які формати графічних файлів використовуються в інтернеті?
11. Який синтаксис включення графіки?
12. Як встановити розмір зображення, що додається до HTML-документа?
13. Як встановити вирівнювання зображення, що додається до HTML-документа?
14. Як зробити малюнок фоном Web-сторінки?
15. Як використовуються теги <figure> та <figcaption>?

16. Яке призначення властивості border-radius?

17. Чому слід завжди визначати альтернативний текст для зображень?

Лабораторна робота № 3

Тема: Створення гіпертекстових посилань.

Мета: Формування вмінь створювати гіперпосилання між HTML-документами.

Питання для вивчення:

1. Гіперпосилання, посилання з одного документа на інший. Тег <a> та його атрибути.
2. Абсолютна та відносна адресації.
3. Псевдоелементи first-line, first-letter.
4. Псевдокласи link, active, visited, hover
5. Використання комбінованих селекторів

Вказівки для ознайомлення із матеріалом та завдання до виконання:

1. CSS Pseudo-elements. З'ясувати особливості використання псевдоелементів first-line та first-letter. Для цього перейти на сторінку http://www.w3schools.com/css/css_pseudo_elements.asp розглянути типові приклади. У розділі Test Yourself with Exercises! виконати вправи 1, 2.
2. В одному з документів, створених під час виконання лабораторних робіт №1-№2, застосувати псевдоелемент first-line.
3. В одному з документів, створених під час виконання лабораторних робіт №1-№2, застосувати псевдоелемент first-letter.
4. HTML Links - Hyperlinks. З'ясувати особливості створення гіпертекстових зв'язків. Для цього перейти на сторінку http://www.w3schools.com/html/html_links.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1, 3, 4, 5.
5. CSS комбіновані селектори. З'ясувати особливості використання селектору нащадків (проміжок), дочірніх (>) та сусідніх (+) селекторів. Для цього перейти на сторінку http://www.w3schools.com/html/html_links.asp розглянути приклади.

6. Створити ієрархічну структуру HTML-документів, розроблених до власного проекту (рис.1).
7. До створених гіперпосилань застосувати псевдокласи link, active, visited, hover.

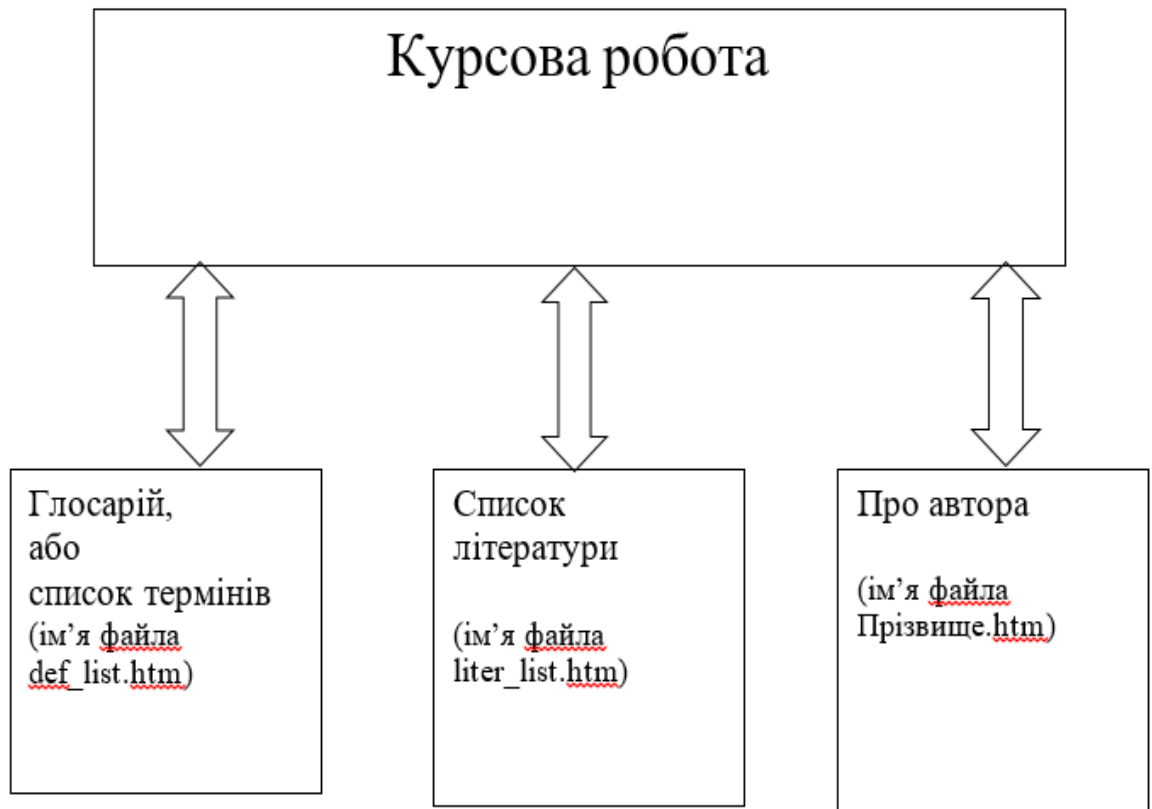


Рисунок 1. Ієрархічну структура HTML-документів

Контрольні запитання

1. Який синтаксис побудови гіпертекстових зв'язків?
2. У чому полягає відмінність між абсолютною та відносною адресацією?
3. Як відносно звернутись до файла що знаходиться в тому ж каталозі, що й вихідний файл? У батьківському каталозі? У сусідньому каталозі? У кореневому каталозі сайту?
4. Як зробити гіперпосилання на елементи поточної web-сторінки?
5. Правила запису текстових гіперпосилань.
6. Правила запису графічних гіперпосилань.
7. Яка різниця між глобальним та локальним посиланням?
8. Принципи використання псевдокласів гіперпосилань
9. Що таке псевдоелемент? Наведіть приклади.
10. Що таке псевдоклас? Наведіть приклади.
11. Що таке комбіновані селектори? Для чого вони використовуються?

12. Чи може елемент мати декілька класів? Чи можуть декілька елементів мати однаковий клас?

Тема 2. Блокова модель в HTML-документах. Основи проектування та верстки WEB-сторінок

Лабораторна робота №4

Тема: Блокова модель в HTML-документах. Створення таблиць.

Мета: Формування вмінь використання блоків та створення таблиць в HTML-документах.

Питання для вивчення:

1. Теги DIV і SPAN.
2. Блокова модель в HTML-документах. Складові блоку.
3. Таблиці в HTML-документах. Атрибути colspan та rowspan.

Вказівки для ознайомлення із матеріалом та завдання до виконання:

1. HTML Tag. З'ясувати особливості використання тегу . Для цього перейти на сторінку http://www.w3schools.com/tags/tag_span.asp розглянути приклади.
2. В одному з документів, створених під час виконання лабораторних робіт №№2, 3, 4 застосувати тегу .
3. The CSS Box Model. З'ясувати особливості блокової моделі в CSS. Для цього перейти на сторінку http://www.w3schools.com/css/css_boxmodel.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-4.
4. Створити HTML-документ, що містить 3 блоки. Вміст перших двох блоків – текст. Вміст третього блоку текст і малюнок. Описати селектори-класи для блоків з використанням усіх можливих властивостей елементів блоку. В одному з блоків використати малюнок як фон (ім'я файла – **boxes.htm**).
5. HTML <table> Tag. З'ясувати особливості створення таблиць у HTML-документах. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_table.asp розглянути приклад.

6. HTML Table Example. Розглянути приклади створення таблиць у HTML-документах. Для цього перейти на сторінку http://www.w3schools.com/html/html_tables.asp У розділі Test Yourself with Exercises! виконати вправи 1-6.
7. CSS Tables. З'ясувати особливості застосування стилів до таблиць у HTML-документах. Для цього перейти на сторінку http://www.w3schools.com/css/css_table.asp розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи 1-6.
8. Створити HTML-документ, що містить 2 таблиці:

Структура таблиці №1

Структура таблиці №2

- Змістове наповнення таблиць – довільне. Застосувати стильове оформлення до створених таблиць. (ім'я файла – tables.htm).
9. Створити таблицю до курсової роботи (ім'я файла – projecttable.htm).
 10. За допомогою гіперпосилання приєднати створену таблицю до титульної сторінки курсової роботи.

Контрольні запитання

1. Для чого призначений тег ? Наведіть приклади використання.
2. Для чого призначений тег <DIV>?
3. У чому полягає ідея блокової моделі в HTML-документах? Назвіть складові блоку.
4. Як визначається висота і ширина блоку?
5. Що визначає властивість margin?
6. Що визначає властивість padding?
7. Що визначає властивість border? Які її характеристики?
8. Як створити таблицю в HTML-документі?
9. . Опишіть базову структуру таблиці HTML?
10. Які основні атрибути елементів TABLE, TR, TD (TH)?
11. Як визначити кількість рядків/стовпчиків таблиці?
12. Як встановити товщину границі таблиці?

13. Як можна задати ширину таблиці?
14. Які існують способи вирівнювання змісту в комірці?
15. Яким чином можна об'єднати комірки таблиці в мові HTML?

Лабораторна робота №5

Тема: HTML5. Структурні елементи та макет web-сторінки.

Мета: Формування вмінь використання структурних елементів у процесі верстки web-сторінки; набування навиків розробки макету сторінки

Питання для вивчення:

1. Теги header, footer, section, article, aside, nav.
2. Застосування технології CSS до структурних елементів.
3. Позиціонування елементів. Властивість CSS position
4. Розробка структурного макету web-сторінки.

Вказівки для ознайомлення із матеріалом та завдання до виконання

1. HTML <header> Tag. З'ясувати особливості використання тегу <header>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_header.asp розглянути приклади.
2. HTML <footer> Tag. З'ясувати особливості використання тегу <footer>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_footer.asp розглянути приклади.
3. HTML <section> Tag. З'ясувати особливості використання тегу <section>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_section.asp розглянути приклади.
4. HTML <article> Tag. З'ясувати особливості використання тегу <article>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_article.asp розглянути приклади.
5. HTML <aside> Tag. З'ясувати особливості використання тегу <aside>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_aside.asp розглянути приклади.
6. HTML <nav> Tag. З'ясувати особливості використання тегу <nav>. Для цього перейти на сторінку http://www.w3schools.com/tags/tag_nav.asp розглянути приклади.
7. CSS position. З'ясувати способи позиціонування елементів сторінки. Для цього перейти на сторінку

https://www.w3schools.com/css/css_positioning.asp розглянути приклади.

8. Ознайомитись з методами створення та оформлення макету сайту. Для цього перейти на сторінку https://www.w3schools.com/html/html_layout.asp розглянути приклади. Докладніше познайомитись з варіантами оформлення макету сайту також можна на <http://htmlbook.ru/samlayout/tipovye-makety/rezinovyi-trekhkolonochnyi-maket>.

9. Виконати верстку HTML-документа з використанням структурних елементів. Вміст документа – титульна сторінка курсової роботи.

Макет сторінки повинен включати:

- «шапку», що охоплює всю ширину сторінки і містить основну назву сторінки, логотип і меню навігації;
- область для розміщення контенту, що складається мінімум з двох колонок – основного блоку та бічної панелі для розміщення, зокрема, мініатюр зображень по темі сторінки;
- футера, що містить інформацію про автора

Візуально це може виглядати наступним чином:

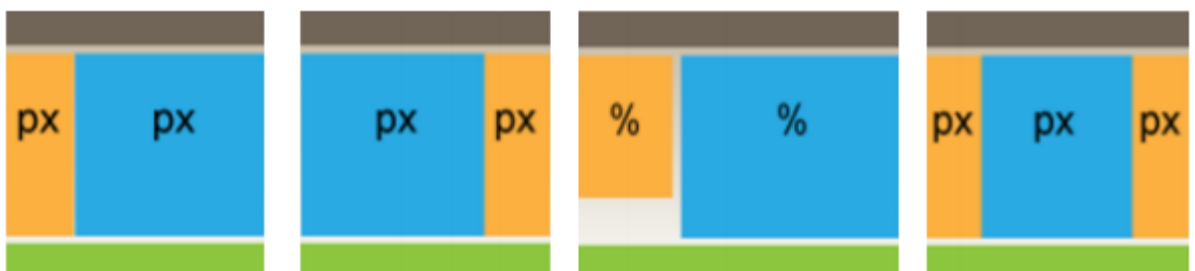


Рис.1 Ієрархічна структура HTML-документів

де ■ - «шапка» сторінки;

■ - основний контент»;

■ - «бічна панель»;

■ - «підвал».

Примітка: ознайомитись з прикладом формування логічної структури макету в надісланій викладачем папці Приклад_5.

10. До створених структурних елементів сторінки додати обгортки, застосувавши технологію CSS.

Контрольні запитання

1. Назвіть відомі Вам структурні елементи мови HTML5. Яке їх призначення?

2. Для чого використовується позиціювання елементів web-сторінки?
3. Які види позиціювання елементів web-сторінки?
4. Які властивості впливають на розмір місця, який блок займає на сторінці?
5. Що таке padding?
6. Що таке margin?
7. Як задати ширину блока в абсолютних одиницях вимірювання та в процентах?
8. Поясніть використання властивості float
9. Як впливають на розташування елемента властивості left, right, top, bottom при різних значеннях position?
10. Які є методи створення та оформлення макету web-сторінки?
11. Як застосовується технологія CSS до структурних елементів мови HTML5?
12. Що таке верстка web-сторінки?
13. Які є підходи до верстки сторінок?
14. Як здійснюється верстка web-сторінок з фіксованою шириною? З «резиною» версткою?

Лабораторна робота 6

Тема: Створення форм та використання елементів користувацького інтерфейсу в HTML-документах.

Мета: ознайомлення з формами як методом взаємодії з користувачем. Формування вмінь створення користувацьких форм на web-сторінці з використанням різних елементів форми.

Питання для вивчення:

1. Призначення форм. Створення форми тегом <form>.
2. Тег <input> для визначення поля введення даних.
3. Теги користувацького інтерфейсу <number>, <label>, <select>, <textarea>, <button>.
4. Тег <radio> для реалізації групи радіокнопок.

Вказівки для ознайомлення із матеріалом та завдання до виконання

1. Html <form> Tag. З'ясувати особливості використання тегу. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_form.asp, розглянути приклади

2. Html <input> Tag. З'ясувати особливості використання тегу <input> для введення поля даних. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_input.asp, розглянути приклади.
3. Html <number> Tag. З'ясувати особливості роботи з числовими полями та використання атрибутів (max, min, step, value). Для цього перейти на сторінку https://www.w3schools.com/tags/att_input_type_number.asp, розглянути приклади.
4. Html <label> Tag. З'ясувати особливості використання тегу <label>. З'ясувати особливості розміщення радіокнопок. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_label.asp, розглянути приклади.
5. Html <textarea> Tag. З'ясувати особливості використання тегу <textarea>. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_textarea.asp, розглянути приклади.
6. Html <select> Tag. З'ясувати особливості використання тегу <select>. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_select.asp, розглянути приклади.
7. Html <checkbox> Tag. З'ясувати особливості використання тегу <checkbox>. Для цього перейти на сторінку https://www.w3schools.com/tags/att_input_type_checkbox.asp, розглянути приклади.
8. Html <button> Tag. З'ясувати особливості використання та стилізації кнопок, створених тегом <button>. Для цього перейти на сторінку https://www.w3schools.com/tags/tag_button.asp, розглянути приклади.
9. Створити індивідуальну сторінку із запропонованою формою для введення певної інформації (за варіантом).
10. До створеної сторінки додати посилання на свою особисту сторінку, створену у попередніх роботах.
11. До сторінки та елементів форми застосувати стильове оформлення.
12. Усі матеріали виконаних завдань (файли .html, .css, .js та скриншот) оформити у вигляді відповідної папки та вислати викладачу.

Варіанти завдань

Варіант 1

Створити форму для введення інформації про людину. Повинна вводитися така інформація:

Прізвище	Просте текстове введення
Ім'я	Просте текстове введення
По батькові	Просте текстове введення
Вік	Вибір з варіантів «до 7», «від 7 до 17», «від 17 до 33», «від 33 до 65» і «65 і старше»
Стать	Повинна бути «радіокнопка»
Статус	Повинен бути «чек-бокс» (студент, працюючий, пенсіонер)

Варіант 2

Створити форму для введення адреси. Повинна вводитися така інформація:

Країна	Вибір зі списку. За замовчуванням—Україна
Поштовий індекс	Просте текстове введення, не більш шести символів
Місто	Просте текстове введення
Вулиця, номер будинку	Просте текстове введення
Вказівка на те, що адреса службова/домашня	Повинна бути «радіокнопка»

Варіант 3

Створити форму для пошуку в каталозі бібліотеки. Повинна вводитися така інформація:

Тема	Вибір зі списку. Можливо вибрати відразу кілька тем
Автор	Вводиться текстом
Назва	Вводиться текстом
Мінімальний рік видання	Вибір зі списку
Максимальний рік видання	Вибір зі списку

Варіант 4

Створити форму для реєстрації нового користувача в системі.
Повинна вводитися така інформація:

Ім'я	Текст
e-mail	Текст
URL	Текст
Пароль	Поле введення пароля
Підтвердження пароля	Поле введення пароля

Варіант 5

Створити форму для введення інформації про те, які предмети хотів би вивчати студент. Предмети мають бути перераховані по одному на рядкові й перед назвою кожного повинен стояти «*check-box*». Початково жоден предмет не повинен бути обраний. Крім того, форма має містити поле для введення імені.

Варіант 6

Створити форму для введення інформації про студента. Повинна вводитися така інформація:

Прізвище	Просте текстове введення
Ім'я	Просте текстове введення
По батькові	Просте текстове введення
Стать	Повинна бути серія « радіокнопок »
Курс	Вибір зі списку

Варіант 7

Створити форму для голосування відразу на кілька посад. Списки кандидатів на кожну посаду повинні бути представлені спадаючими меню. Напроти кожного списку написати посаду, на яку претендують кандидати. У списках кандидатів має бути позиція «Проти всіх». Ця позиція і обирається за замовчуванням.

Варіант 8

Створити форму для пошуку відносних матеріалів у бібліотеці.
Повинна вводитися така інформація:

Ключові слова	Уводяться текстом
Рік публікації	
Характер публікації	Список з можливістю вибору декількох варіантів. Наприклад, «Стаття в журналі», «Дисертація», «Монографія» та ін.
Мова	Список з можливістю вибору декількох варіантів
Доставка	Серія радіокнопок. Наприклад «Підняти в читальний зал», «Доставити на абонемент», і т.д.

Варіант 9

Створити форму для пошуку в мережі з можливістю вибору системи пошуку. Повинна вводитися така інформація:

Запит	Вводиться текстом
Система пошуку	Вибір зі списку
Кількість записів, що вертаються	Вибір зі списку. За замовчуванням, не обмежене

Варіант 10

Створити форму для замовлення товарів. Форма повинна пропонувати замовити відразу кілька товарів. Для кожного слід указувати кількість. Один раз слід вказати адресу доставки. Повинна вводитися така інформація:

Найменування товару	Вибір зі списку. Зробіть кілька однакових списків, щоб можна було замовити відразу кілька різних товарів
Кількість	Лічильник цілих чисел. Потрібно зробити окреме поле для введення кількості, напроти кожного списку товарів
Адреса доставки	Текст

Контрольні запитання

1. Що таке форма та її призначення?

2. Які основні атрибути тега `<form>`?
3. Яке призначення тега `<input>`?
4. Які атрибути тега `<input>` є обов'язковими?
5. Чим відрізняються `<input type="text" />` та `<input type="textarea" />`?
6. Що таке `<label>`, яке його відношення до елементів форми?
7. Який вигляд має `<input type="checkbox" />`?
8. Які елементи форми ви знаєте і їх призначення?
9. Властивості елементів форми
10. Які є атрибути у числових полів?
11. Який тег використовується для задання багаторядкового тексту?
12. Які атрибути можна використати для кнопок, заданих тегом `<button>`?
13. Як можна стилізувати кнопки за допомогою технологій CSS?
14. Як організувати засобами HTML вибір за допомогою радіокнопок?

Тема 3. Web-програмування на боці клієнта. Основи Javascript

Лабораторна робота 7

Тема: Розробка динамічних Web -сторінок за допомогою мови Javascript.

Мета: освоєння практики створення інтерактивних Web-документів засобами JavaScript.

Питання для вивчення:

1. Можливості та призначення мови JavaScript (JS).
2. Способи підключення скрипту JS до web-сторінки.
3. Синтаксис та особливості мови JS.
4. Можливості відображення даних в JS: `innerHTML`, `windows.alert()`, `console.log()`.
5. Об'єктна модель документа HTML DOM.
6. Способи доступу до елементів (вузлів) DOM за допомогою JS.

Вказівки для ознайомлення із матеріалом

1. Ознайомитись з можливостями мови JavaScript (JS) для організації сценаріїв на боці клієнта. Для цього перейти на сторінку

https://www.w3schools.com/js/js_intro.asp, розглянути типові приклади.

2. Ознайомитись з особливостями підключення скрипту JS до web-сторінки. Для цього перейти на сторінку https://www.w3schools.com/js/js_where.asp, розглянути типові приклади.
3. Ознайомитись із синтаксисом об'яв та операторів JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_statements.asp, розглянути типові приклади.
4. Ознайомитись із правилами синтаксису, операторами, змінними, ідентифікаторами тощо. Для цього перейти на сторінку https://www.w3schools.com/js/js_syntax.asp, розглянути типові приклади.
5. Ознайомитись із можливостями відображення даних в JS: вивід даних в елемент HTML innerHTML, вивід у вікно windows.alert(), вивід у консоль console.log(). Для цього перейти на сторінку https://www.w3schools.com/js/js_output.asp, розглянути типові приклади.

***Примітка:** для перегляду результатів програми у консолі в браузері Chrome потрібно перейти на вкладку Дополнительные инструменты/Инструменты разработчика*

6. Ознайомитись із об'єктною моделлю DOM для HTML-документів та з'ясувати її особливості. Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom.asp.
7. Метод getElementById HTML DOM. З'ясувати особливості застосування методу getElementById для доступу до HTML-елемента та властивості innerHTML для отримання/заміни вмісту елемента. Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom_methods.asp.
8. Ознайомитись із особливостями пошуку та звернення до HTML-елементів (вузлів) в DOM-моделі документа. Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom_elements.asp, розглянути типові приклади.
9. Ознайомитись із роботою з елементами HTML DOM (добавлення, вилучення, заміна вузлів). Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom_nodes.asp, розглянути типові приклади.

Завдання до виконання

10. Ознайомитись з скриптом, наведеним у надісланій викладачем папці Приклад_7. На базі цієї папки сформувати власну, доповнивши її наступними завданнями.
11. Додати у скрипт коментар про подію, що відбувається, заповнивши відповідно порожні рядки, що починаються з
// в результаті
Примітка: За зразок взяти заповнені у скрипті коментарі
12. Доповнити за зразком форму «Отримані дані: » даними, зчитавши їх з решта текстових елементів та застосувавши до елементів виведення стильове оформлення.
13. Виведіть у консоль значення зчитаних змінних (п.12) та зробіть відповідний скриншот. Додати його у папку із власним завданням.
14. Доповнити форму полями: «Дата», «E-mail», відобразивши це у виводі, змінити розташування форми Отримані дані (рис.2).
15. Біля полів «П.І.П.» та «E-mail» поставити символ «*» (заповнення обов'язкове) і у разі їх незаповнення виводити відповідне повідомлення у діалоговому вікні, використавши елемент відображення **windows.alert** (рис.3)
16. При кожному запуску оновляти форму Отримані дані, вилучивши із неї усі створені записи (використати метод вилучення вузлів DOM)
17. Оформити скрипт в окремому файлі поточної папки.

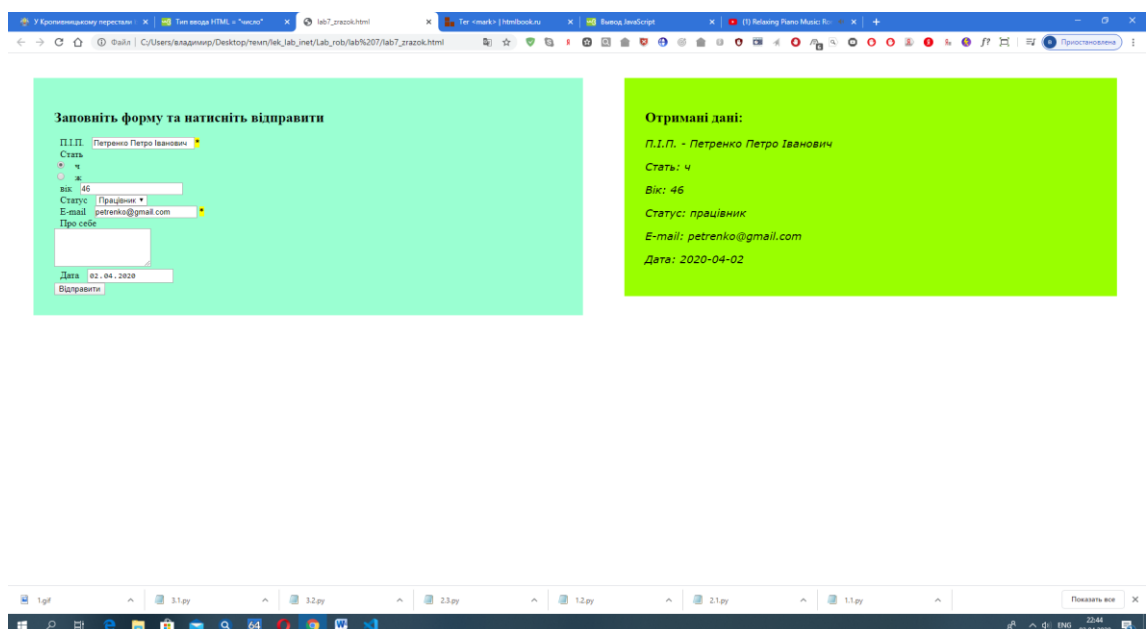


Рис.2. Рекомендований вигляд html-сторінки

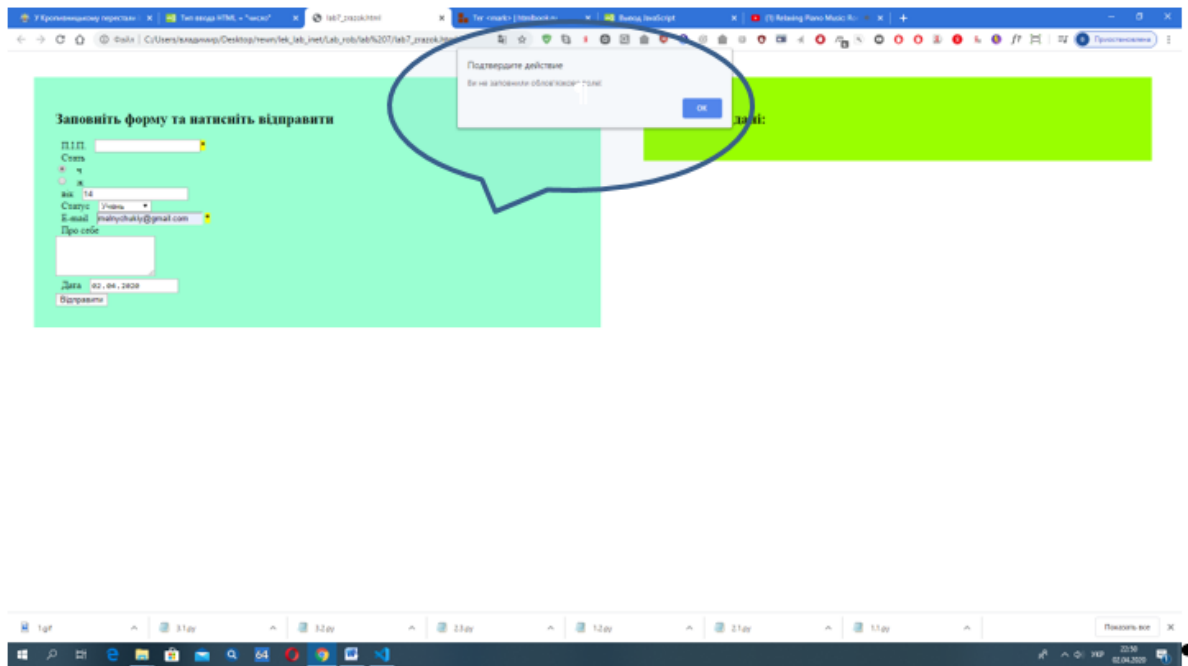


Рис.3. Виведення діалогового вікна у випадку незаповнення обов'язкового поля

Контрольні запитання

1. Яке призначення мови сценаріїв JavaScript?
2. Які є способи включення сценаріїв JavaScript у документ HTML?
3. Що таке ідентифікатор в JavaScript?
4. Як оголосити змінну в JavaScript?
5. Які типи даних передбачає JS?
6. Які є оператори в JS?
7. Порівняйте правила задання операторів у мові JavaScript та Python.
8. Які є способи відображення даних у JavaScript? Як переглянути виведення даних у консоль?
9. Як визначити власну функцію у коді JavaScript?
10. Які особливості об'єктної моделі документа DOM?
11. Які є види вузлів DOM?
12. Як знайти елемент по його ідентифікатору?
13. Які способи доступу до вузлів DOM за допомогою JavaScript?
14. Які принципи маніпуляція вузлами DOM за допомогою JS?

Лабораторна робота 8

Тема: Оператори JavaScript. Умовний оператор. Використання регулярних виразів (шаблонів) для заповнення текстових полів.

Мета: Формування вмінь розробляти сценарії JavaScript з використанням управляючих операторів (умовного оператора) та застосовувати шаблони для заповнення текстових полів.

Питання для вивчення:

1. Використання шаблонів для введення регулярних виразів в текстові поля.
2. Оператори JS.
3. Концепція типів даних JS.
4. Умовні конструкції JS. Особливості їх застосування.
5. Управляючі конструкції JS: користувацькі функції.
6. Обробники подій.

Вказівки для ознайомлення із матеріалом

1. Регулярні вирази (шаблони), атрибут pattern. З'ясувати особливості використання регулярних виразів для співставлення послідовностей символів в текстових полях теги `<input>`. З'ясувати особливості роботи з елементами типу "submit", "email", "password". Для цього перейти на сторінку https://www.w3schools.com/tags/att_input_pattern.asp, розглянути типові приклади.
2. Оператори JS: арифметичні, присвоєння, порівняння, логічні оператори. З'ясувати особливості організації та застосування операторів JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_operators.asp, розглянути типові приклади.
3. Типи даних JS. З'ясувати основні положення концепції типів даних JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_datatypes.asp, розглянути типові приклади.
4. Функції перетворення типів даних JS, функція `parseInt()`. З'ясувати особливості перетворення типів в JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_datatypes.asp.
5. Умовна конструкція JS "if .. else". З'ясувати особливості використання умовних конструкцій в JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_if_else.asp, розглянути типові приклади. У розділі Test Yourself with Exercises! виконати вправи.
6. Організація функцій у JS. Ознайомитись з особливостями організації функцій у JS. Функції, що викликаються зі скрипта, та функції, що

викликаються подією. Для цього перейти на сторінку https://www.w3schools.com/js/js_functions.asp. У розділі Test Yourself with Exercises! виконати вправи.

7. Події JS. Обробники подій. Ознайомитись з подіями JS та з'ясувати як створюються в JS обробники подій. Для цього перейти на сторінку https://www.w3schools.com/js/js_events.asp , розглянути типові приклади. У розділі Test Yourself with Exercises! виконати вправи.

Завдання до виконання

8. Ознайомитись з вмістом надісланої викладачем папки priklad_8. Проекспериментуйте із html-сторінкою. Зверніть увагу на використання регулярних виразів у полі «Ім'я», що обмежує введення будь-яких символів, окрім латиниці/кирилиці. Зверніть уваги на коментарі у файлі .js.
9. Створити форму «Форма студента» з полями для заповнення: П.І.П., E-mail, використавши відповідно тег <input> type відповідно "text" та "email" та кнопками підтвердження (тег <input>, type "submit") (рис.4). Передбачити застосування для полів введення в якості шаблонів відповідних регулярних виразів (для введення лише валідних даних).
10. Додати у форму «Форма студента» поля для заповнення: трьох навчальних предмета на вибір студента, поля для введення відповідних сумарних балів за них за шкалою від 1 до 100 (використати тег <input>, type "number"). Передбачити також автоматичну конвертацію (кожної оцінки) у систему ECTS ("A-F").
***Примітка:** передбачити створення для кожного елемента унікального id для подальшої обробки у скрипті; оформити процедуру конвертації оцінок у вигляді функції, для її багатократного використання для усіх предметів.*
11. Розробити структурну модель сторінки: шапку(картинка, назва), основний блок та футер (посилання на сторінку «Дані про автора»). Застосувати різні стилі CSS для оформлення HTML-сторінки.
12. Оформити стильову модель (.css) та скрипт (.js) в окрему файлі поточної папки.
13. Додати у скрипт додатковий сценарій за індивідуальним варіантом, доповнивши за необхідністю форму додатковими елементами DOM.
14. Усі матеріали виконаних завдань (файли .html, .css, .js та скриншот) оформити у вигляді відповідної папки та вислати викладачу.

Варіанти завдань

Варіант 1.

Додати сценарій обрахунку загальної суми балів з усіх предметів з подальшим виведенням результату у консоль `console.log()` та у вікно `windows.alert()`.

Варіант 2.

Додати сценарій обрахунку середнього балу з усіх предметів з подальшим виведенням результату у текстове поле.

Варіант 3.

Додати сценарій обрахунку мінімального балу з усіх предметів з подальшим виведенням результату у консоль `console.log()` та у текстове поле.

Варіант 4.

Додати сценарій обрахунку загальної суми балів з усіх предметів з подальшим виведенням результату у консоль `console.log()` та у текстове поле.

Варіант 5.

Додати сценарій обрахунку максимального балу з усіх предметів з подальшим виведенням результату у консоль `console.log()` та у вікно `windows.alert()`.

Варіант 6.

Додати сценарій визначення успішності студента за його оцінками в ECTS за принципом: якщо серед його показників ECTS немає {D,E,F,Fx}, то успішність отримує статус «висока», інакше «невисока». Результат успішності вивести у текстове поле.

Варіант 7.

Додати сценарій обрахунку максимального балу з усіх предметів з подальшим виведенням результату у вікно `windows.alert()` та у текстове поле.

Варіант 8.

Додати сценарій визначення можливості претендування студента на стипендію за його оцінками в балах з усіх предметів. Якщо бали з усіх

предметів перевищують 65, то студент може претендувати на стипендію, інакше – ні. Результат перевірки вивести у вікно `windows.alert()` та у консоль `console.log()`.

Варіант 9.

Додати сценарій визначення успішності студента за його оцінками в ECTS за принципом: якщо серед його показників ECTS лише {A,B,C}, то успішність «висока», якщо є хоча би один показник {F,Fx}, то успішність отримує статус «недостатня», в інших випадках - «достатня». Результат успішності вивести у вікно `windows.alert()`.

Варіант 10.

Додати сценарій обрахунку загальної суми балів з усіх предметів з подальшим виведенням результату у консоль `console.log()` та у вікно `windows.alert()`.

The image shows a web browser window displaying a form titled "Форма студента". The form has a purple header bar with the title. Below the header, there are two input fields for "П.І.П." and "E-mail", each followed by a "Увести" button. A callout box points to these fields, stating: "Поля, що містять регулярні вирази для обмеження введення символів". Another callout box points to the "Увести" buttons, stating: "Поля type='submit'". Below the input fields, there is a section titled "Успішність по обраним предметам". This section contains a table with three columns: "Назва предмету", "В балах (1..100)", and "Конвертація в ECTS(A..Fx)". The table has three rows: "Предмет 1", "Предмет 2", and "Предмет 3". Below the table, there is a "Конвертувати" button. Two callout boxes point to the table, stating: "Label, оформлені певним стилем CSS" and "Поля, що уводяться, обраховуються". At the bottom of the form, there is a footer bar with the text "Футер, додати посилання на сторінку автора". A callout box points to the "Конвертувати" button, stating: "Кнопка, що ініціює подію конвертації".

Рис. 4. Рекомендований вигляд форми

Контрольні запитання

1. Арифметичні оператори в JS та оператори присвоєння.
2. Оператори порівняння та логічні оператори в JS, їх особливості?
3. Базові типи даних JS.

4. Для чого використовується функція перетворення типу parseInt()?
5. Умовний оператор JS.
6. Застосування регулярних виразів в JS.
7. Для чого використовується при уведенні даних атрибут pattern?
8. Як перевірити, чи є символи цифровими?
9. Як перевірити, чи є символи латинськими?
10. Що представляють собою оброблювачі подій?
11. Функції у JS. Функції, що викликаються зі скрипта, та функції, що викликаються подією.
12. Яка базова конструкція функції в JavaScript?
13. Змінні в JS. Зона дії змінних.
14. Яке призначення методу alert?

Лабораторна робота 9

Тема: Циклічні конструкції JavaScript. Робота з таблицями DOM

Мета: Формування вмінь розробляти сценарії JavaScript з використанням циклічних конструкцій JavaScript та застосовувати операції обробки таблиць DOM.

Питання для вивчення:

1. Циклічні конструкції JS. Особливості операторів “for ..” та “while..”.
2. Організація масивів у JS. Методи обробки масивів.
3. Операції обробки таблиць: створення та розміщення таблиць в HTML-сторінку; управління атрибутами таблиці; зчитування та обробка елементів таблиць.
4. Пошук та маніпуляції з елементами (вузлами) DOM за допомогою JS.

Вказівки для ознайомлення із матеріалом

1. Циклічна конструкція JS “for ..”. З’ясувати синтаксис та особливості використання циклічної конструкції “for..” в JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_loop_for.asp, розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи.
2. Циклічна конструкція JS “while ..”. З’ясувати синтаксис та особливості використання циклічної конструкції “ while..” в JS. Для цього перейти на сторінку

https://www.w3schools.com/js/js_loop_while.asp, розглянути приклади.

У розділі Test Yourself with Exercises! виконати вправи.

3. Організація масивів у JS. Ознайомитись з особливостями організації масивів у JS та операціями з їх елементами. Для цього перейти на сторінку https://www.w3schools.com/js/js_arrays.asp, розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи.
4. Методи обробки масивів у JS. Для ознайомлення з методами обробки масивів та операціями з елементами масиву перейти на сторінку https://www.w3schools.com/js/js_array_methods.asp, розглянути приклади. У розділі Test Yourself with Exercises! виконати вправи.
5. Пошук HTML-елементів. З'ясувати особливості методів пошуку елементів JS DOM: getElementById(), getElementsByTagName(), getElementsByClassName(). Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom_elements.asp, розглянути приклади.
6. З'ясувати особливості роботи з вузлами. З'ясувати особливості методів створення (createElement(), createTextNode()), додавання (appendChild(), insertBefore()) та вилучення (remove()) елементів DOM засобами JS. Для цього перейти на сторінку https://www.w3schools.com/js/js_html_dom_nodes.asp, розглянути приклади.

Завдання до виконання

7. Ознайомитись з вмістом папки prikklad_9. Проекспериментуйте із html-сторінкою. Файлу script_9 поміняйте розширення на .js . Зверніть уваги на коментарі у файлі .js. Ознайомитись із макетом сторінки (.css).
8. У файлі script_9.js заповніть коментарі (поля, що починаються // ??).
9. Оформити футер HTML-сторінки.
10. У динамічно створюваній таблиці «Успішність» замінити поле «Сума» на «Середнє арифметичне» та відповідні розрахунки суми на розрахунки середнього арифметичного для кожного студента.
11. Передбачити кнопку «Очистити», яка видаляє створену таблицю, та приховує кнопку «Обрахувати».
12. Додати кнопку «Сформувати список студентів», яка формує список студентів та виводить його у праву колонку HTML-сторінки (рис. 5) .
13. Протестувати сторінку та зробити скриншот сторінки із проведеними розрахунками та сформованим списком студентів.
14. Усі матеріали виконаних завдань (файли .html, .css, .js та скриншот) оформити у вигляді відповідної папки та вислати викладачу.



Таблиця "Успішність"

Для формування таблиці задайте відповідно її розмірність

Заголовки рядків - це Прізвища, заголовки стовпців - предмети, значення комірок - бали

Виберіть кількість студентів для відображення

Виберіть кількість предметів для відображення

Прізвище/ Предмет	Алгебра	Фізика	Матаналіз	Сума
Ручко	5	5	4	14
Іваненко	4	3	4	11
Петренко	5	3	4	12
Кошуба	2	4	3	9
Сидір	3	3	3	9

Дані про автора сторінки, посилання

Сформований список студентів

Прізвища студентів

1. Ручко - 14
2. Іваненко - 11
3. Петренко - 12
4. Кошуба - 9
5. Сидір - 9

Рис. 5. Рекомендований вигляд форми

Контрольні запитання

1. Циклічні конструкції JS. Особливості оператора “for ..”.
2. Циклічні конструкції JS. Особливості оператора “while..”
3. Організація масивів у JS.
4. Порівняйте особливості створення та доступу до масиву у JS та python.
5. Що представляють собою оброблювачі подій?
6. Функції у JS. Функції, що викликаються зі скрипта, та функції, що викликаються подією.
7. Як передаються параметри у функції у JS?
8. Назвіть основні операції у мові сценаріїв JS.
9. Які є методи пошуку вузлів DOM за допомогою JavaScript?
10. Які є методи створення нових вузлів в JS DOM?
11. Які є методи створення додавання/вилучення вузлів в JS DOM?
12. Які принципи маніпуляція вузлами DOM за допомогою JS?
13. Що зберігає масив childNodes?
14. Що зберігає властивість parentNode?

Тема 4. Використання бібліотек JavaScript для розробки web-сторінок. Бібліотека jQuery

Лабораторна робота 10

Тема: Розробка динамічних web-сторінок за допомогою JavaScript-бібліотеки jQuery.

Мета: Формування вмінь розробляти динамічні web-сторінки за допомогою Javascript-бібліотеки jQuery. .

Питання для вивчення:

1. Можливості та призначення бібліотеки jQuery.
2. Способи підключення скрипту бібліотеки jQuery до web-сторінки.
3. Синтаксис та селектори jQuery.
4. Ефекти jQuery: приховати/показати, переміщення, анімація.
5. Методи jQuery для управління елементами і атрибутами HTML.

Вказівки для ознайомлення із матеріалом

1. Ознайомитись анімація з можливостями бібліотеки jQuery для спрощення організації сценаріїв на боці клієнта. Для цього перейти на сторінку https://www.w3schools.com/jquery/jquery_intro.asp.
2. Ознайомитись зі способами та особливостями підключення бібліотеки jQuery до web-сторінки. Для цього перейти на сторінку https://www.w3schools.com/jquery/jquery_get_started.asp
3. Синтаксис jQuery Ознайомитись із синтаксисом та особливостями застосування методів jQuery. Для цього перейти на сторінку https://www.w3schools.com/jquery/jquery_syntax.asp.
4. Селектори jQuery. З'ясувати види селекторів jQuery та особливості їх застосування. Для цього перейти на сторінку https://www.w3schools.com/jquery/jquery_selectors.asp, розглянути типові приклади.
5. Ефекти jQuery. З'ясувати основні види ефектів jQuery: приховати/показати (hide()/show()), переміщення (fade(), slide()), анімація (animate()).
https://www.w3schools.com/jquery/jquery_hide_show.asp,
https://www.w3schools.com/jquery/jquery_fade.asp,
https://www.w3schools.com/jquery/jquery_slide.asp,

https://www.w3schools.com/jquery/jquery_animate.asp, розглянути типові приклади.

6. Методи jQuery управління елементами DOM. З'ясувати можливості та особливості jQuery маніпулювання елементами DOM. Методи text(), html(), val(). Для цього перейти на сторінку https://www.w3schools.com/jquery/jquery_dom_get.asp, розглянути типові приклади.

Завдання для виконання

7. Відкрити власну web-сторінку із індивідуальною формою, розроблену засобами HTML та CSS, в ході виконання лабораторної роботи № 6. В один із зазначених способів підключити до сторінки бібліотеку jQuery.
8. Застосувати до елементів сторінки ефекти jQuery: приховування елементів форми, переміщення та анімацію. Доповнити за необхідності форму кнопками та додатковими елементами.

Контрольні запитання

1. Які переваги популярних JavaScript-бібліотек?
2. Охарактеризуйте концепції бібліотеки jQuery.
3. Які способи підключення бібліотеки jQuery до web-сторінки?
4. Синтаксис та особливості методів jQuery.
5. Селектори бібліотеки jQuery.
6. Чому методи jQuery розміщуються в події готовності документа?
7. Як здійснюється доступ до вузлів DOM засобами бібліотеки jQuery?
8. Які принципи маніпуляції вузлами DOM за допомогою бібліотеки jQuery?
9. Які можливості надає бібліотека jQuery?
10. Охарактеризуйте методи бібліотеки text(), html(), val()

Питання до екзамену

1. Комп'ютерні мережі: призначення сервера та браузера
2. Комп'ютерні мережі: доменна організація імен
3. Комп'ютерні мережі: призначення протоколу HTTP. Особливості роботи HTTP.
4. Класифікація інтернет-технологій
5. Веб-сайти: класифікація, етапи розробки.
6. Інтернет-технології створення сайту CMS: загальні поняття
7. Мова HTML, основні поняття: теги, елементи, атрибути.
8. Структура HTML-документа. Організація вмісту HTML-документа
9. Поняття фізичної та логічної розмітки тексту та відповідні теги
10. Застосування каскадної таблиці стилів CSS
11. Синтаксис CSS, правила синтаксису.
12. Використання селекторів та ідентифікаторів для стилізації елементів Теги структурної розмітки для організації списків (нумеровані, маркіровані, означень).
13. Засоби CSS для оформлення списків
14. Підключення графіки до HTML-документа. Атрибути зображення.
15. Структурування текстів у Web-дизайні. Розмітка абзаців, визначення заголовків та підзаголовків, розділів та підрозділів тексту
16. Технологія використання графічних зображень у Web-документах. Типи та формати зображень для Web-документів. Особливості використання зображень.
17. Призначення та навігаційні можливості гіпертекстових посилань. Види гіпертекстових посилань на сайті.
18. Технологія створення гіперпосилань для об'єднання сторінок в єдиний інформаційний продукт
19. Стили CSS для створення псевдокласів `hover`, `link`
20. Блокова модель HTML-документа, теги `DIV`, `SPAN`.
21. Блокова модель HTML-документа, відступи, рамки, позиціонування.
22. Таблиці в HTML-документах.
23. Структурні елементи веб-документа: теги `HEADER`, `NAV`, `SECTION`
24. Структурні елементи веб-документа: теги `SECTION`, `ASIDE`, `FOOTER`

25. Призначення, можливості Java Script
26. BOM-модель браузера
27. DOM-модель документа
28. Основні події веб-застосувань
29. Поле Input, типи, властивості, атрибути
30. Синтаксис JS: основні правила
31. Конструкції розгалуження в Java Script
32. Циклічні конструкції Java Script
33. Регулярні вирази в Java Script
34. Робота з таблицями в DOM
35. Методу доступу в Java Script до DOM-об'єктів

Список скорочень і спеціальних термінів

Контент (web-сторінки, сайту) - це інформація, розташована на сторінках сайту. Інформація може бути як текстова, так і графічна, мультимедійна і т.д. Іншими словами контентом називають все те, що можна знайти за допомогою пошукових систем. У будь-якому випадку найпоширенішим контентом є тексти (статті).

SEO розшифровується як Search Engine Optimization - це оптимізація web-сторінки для подальшого просування сайту в рейтингу пошукових систем

CMS - це англійська аббревіатура, яка розшифровується як Content Management System. Дослівний переклад - Система керування контентом. CMS часто називають словами «движок», «адмінка» або просто: «на чому побудований сайт».

Сайт - це набір різноманітних інтернет-сторінок, які доступні за певними адресами в інтернеті (URL). CMS дозволяє управляти контентом сайту, змінювати, додавати і видаляти сторінки сайту, а також коригувати їх вміст.

Для створення web-сторінок та сайтів використовують: HTML, CSS, JavaScript, FLASH, PHP (Perl), реляційні БД, наприклад, MySQL. Клієнтське середовище (браузер) – візуальна частина роботи програми- тут відображаються HTML сторінки у вікні і обслуговуються історії сеансів HTML-сторінок, що відображаються в браузері протягом сесії. Об'єкти середовища за допомогою клієнтської мови – JavaScript повинні мати можливість маніпулювати сторінками, вікнами, історією. При роботі з сервером необхідні мови типу PHP (Perl), рел. БД, наприклад, MySQL.

HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») - стандартна мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). За допомогою HTML необхідно розмітити текст, описати за допомогою тегів його структуру. За допомогою CSS-можна розширити можливості по оформленню Web-сторінок.

CSS - (Cascading Style Sheets) - Каскадні таблиці стилів. Це звід стильових описів, тих чи інших HTML тегів (далі елементів HTML), який може бути застосований як до окремого тегу - елементу, так і одночасно до всіх ідентичних елементів на всіх сторінках сайту.

JS (JavaScript) – мова програмування, що вбудовується в HTML-сторінки з метою зручності спілкування з користувачем, обробки помилок та створення динамічності сторінки в цілому. JS є об'єктно-орієнтованою та

інтерпретованою мовою, а також системно-незалежною, оскільки працює на будь-яких платформах. Зазвичай JS код називають скриптами.

Dom (Document Object Model) - об'єктна модель документа, яка дозволяє динамічно змінювати веб-сторінку, застосовуючи мову написання сценарію. DOM має для кожного елемента або об'єкта, що визначається за допомогою атрибуту ID (ідентифікатора об'єкта), функцію JavaScript, що дозволяють керувати властивостями атрибутів об'єкта, що задаються через CSS.

Правила хорошого тону Web - дизайну вимагають щоб уявлення Web - сторінки було відокремлене від її структури та сценарію обробки. Тобто, визначення стилів виноситься в окремі CSS файли, а скрипти – відповідно у JS файли.

Рекомендована література

1. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну
Видавництво: Вид група ВНУ, 2009, 336с.
2. Фримен Э., Робсон Э. Изучаем программирование на HTML5. –
СПб.: Питер, 2013. – 640 с.
3. Фримен Э., Фримен Э. Изучаем HTML, XHTML и CSS. – СПб.:
Питер, 2012. – 656 с.
4. Моррисон М. Изучаем JavaScript. – СПб.: Питер, 2013. – 592 с
5. Інтернет-уроки w3schools [Електронний ресурс]
<https://www.w3schools.com/>
6. Глобальні інформаційні мережі. Бібліотека jQuery [Електронний
ресурс] <https://www.victoria.lviv.ua/library/students/km/work4.html>