

Міністерство освіти і науки України

Національний університет водного господарства та  
природокористування

Кафедра обчислювальної техніки

**04-04-245M**

## **Методичні вказівки**

до лабораторних робіт з навчальної дисципліни  
**«Web-програмування»**  
для здобувачів вищої освіти першого (бакалаврського)  
рівня за освітньо-професійною програмою «Комп'ютерна  
інженерія» спеціальності 123 «Комп'ютерна інженерія»  
денної і заочної форм навчання

Рекомендовано науково-  
методичною радою з якості  
ННІАКОТ  
Протокол № 6 від 26.04.2022 р.

Рівне – 2022

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Web-програмування» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» денної і заочної форм навчання [Електронне видання] / Бойчура М. В., Сидор А. І. – Рівне : НУВГП, 2022. – 38 с.

Укладачі: Бойчура М. В., к.т.н., старший викладач кафедри  
обчислювальної техніки;  
Сидор А. І., к.т.н., доцент кафедри  
обчислювальної техніки.

Відповідальний за випуск: Круліковський Б. Б., к.т.н.,  
доцент, завідувач кафедри обчислювальної техніки.

Керівник (гарант) освітньої програми  
«Комп'ютерна інженерія»  
спеціальності  
123 «Комп'ютерна інженерія»

Сидор А. І.

© М. В. Бойчура,  
А. І. Сидор, 2022  
© НУВГП, 2022

## ЗМІСТ

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Вступ .....                                                                             | 4  |
| Лабораторна робота №1. Розмітка Web-сторінки засобами HTML. Теги.....                   | 6  |
| Лабораторна робота №2. Розмітка Web-сторінки засобами HTML. Атрибути .....              | 10 |
| Лабораторна робота №3. Структурування Web-сайту .....                                   | 15 |
| Лабораторна робота №4. Вступ до CSS .....                                               | 18 |
| Лабораторна робота №5. Найпростіші застосування JavaScript при розробці Web-сайтів..... | 22 |
| Лабораторна робота №6. Мова JavaScript для забезпечення динамічності сайту .....        | 27 |
| Лабораторні роботи № 7-8. Фреймворк Vue .....                                           | 30 |
| Лабораторні роботи № 9-10. Фреймворк Express .....                                      | 34 |
| Рекомендована література.....                                                           | 37 |

## Вступ

Методичні вказівки дисципліни «Web-програмування» є невід'ємною частиною нормативно-методичного забезпечення навчального процесу зі сфери знань 12 «Інформаційні технології». Курс розроблено відповідно до стандарту підготовки бакалавра з предметної області 123 «Комп'ютерна інженерія». Знання та вміння з дисципліни «Web-програмування» допомагають набути наступні фахові навички: вміння використовувати сучасні методи та мови програмування для розробки алгоритмів та Web-застосунків для мереж; володіння базовими елементами, що допомагають успішно писати та захищати бакалаврську роботу.

Дисципліна «Web-програмування» вивчається у 3-му семестрі і є ключовою до оволодіння знаннями та навичками, які готують до професійної діяльності з планування, проектування та розробки Web-додатків. Курс передбачає комплексну підготовку фахівця з Web-програмування у всіх аспектах набуття професійних навичок, визначених у курсі бакалавра комп'ютерної інженерії.

Метою вивчення дисципліни є надання теоретичних знань і практичних навичок створення Web-додатків, а також вивчення сучасного програмного забезпечення для підтримки та організації процесу створення продукту.

Основною ціллю викладання даної дисципліни є розвиток професійних компетенцій студентів для розуміння всього циклу розробки програмного забезпечення: проектування, планування, розробка, тестування, доставка та розгортання Web-додатків.

В результаті виконання усіх лабораторних робіт студенти повинні:

Знати:

- фази розробки програмного забезпечення;
- мову гіпертекстової розмітки HTML;
- каскадну таблицю стилів CSS;
- мову динамічного відображення тексту JavaScript;
- фреймворки Vue та Express;
- сучасні засоби розробки;
- переваги використання фреймворку для полегшення розробки програмного забезпечення;
- знати структуру Web-проекту;

Вміти:

- правильно організувати процес розробки програмного забезпечення;
- створювати власні проекти та розбивати їх на завдання;
- вдало комбінувати мови HTML і JavaScript, стилі CSS та фреймворки Vue і Express;
- програмувати відповідно до принципів Code Convention.

## **Лабораторна робота №1**

### **Розмітка Web-сторінки засобами HTML. Теги**

#### *Мета*

Ознайомитись із синтаксисом, зокрема тегами, мови HTML.

#### *Завдання*

1. Створити Web-сторінку, яка буде містити 100 різноманітних тегів (не враховуючи закриваючі теги) мови HTML.

#### *Варіанти до виконання поставленого завдання*

Варіант 1. Потрібно створити сторінку із автобіографією. Там описати свої хобі (із фотографіями), результати поточної успішності (у вигляді таблиці) і т.д.

Варіант 2. Розробити сторінку інтернет-магазину. Тут описати інформацію про наявні продукти (із фотографіями), способи доставки, вартість товарів (у вигляді таблиці) і т.д.

Варіант 3. Розробити сторінку кафедри. Тут описати інформацію про викладачів (із фотографіями), предмети, які викладають викладачі кафедри (у вигляді таблиці) і т.д.

Варіант 4. Розробити сторінку аптеки. Тут описати перелік ліків (із фотографіями), їх кількості та ціни (у вигляді таблиці) і т.д.

Варіант 5. Розробити сторінку футбольного комітету. Тут описати керівний склад комітету (із фотографіями), список команд з додатковою інформацією (у вигляді таблиці) і т.д.

Варіант 6. Розробити сторінку громадської організації. Тут описати основних учасників організації (із фотографіями), напрямки діяльності з вказанням відповідальних учасників (у вигляді таблиці) і т.д.

Варіант 7. Розробити сторінку студентського гуртка. Тут описати основних учасників гуртка (із фотографіями), розклад заходів у поточному році (у вигляді таблиці) і т.д.

Варіант 8. Розробити сторінку косметичної компанії. Тут описати наявність розповсюджуваної продукції (із фотографіями), опис характеристик та ціну (у вигляді таблиці) і т.д.

Варіант 9. Розробити сторінку політичної організації. Тут описати керівний склад організації (із фотографіями), напрямки діяльності та відповідальних виконавців (у вигляді таблиці) і т.д.

Варіант 10. Розробити сторінку з заголовком «Олімп». Тут описати грецьких або римських богів (із фотографіями), сфери відповідальності кожного з них (у вигляді таблиці) і т.д.

Варіант 11. Розробити сторінку інформаційного відділу підприємства. Тут описати персональний склад відділу (із фотографіями), перелік послуг з вказанням їх вартості (у вигляді таблиці) і т.д.

Варіант 12. Розробити сторінку Вашої студентської групи. Тут описати перелік студентів, які навчаються у даній академічній групі (із фотографіями), персональні дані (номер залікової крижки, дата народження, адреса проживання – у вигляді таблиці) і т.д.

Варіант 13. Розробити сторінку банку. Тут описати базу даних клієнтів банку (із фотографіями), наявні послуги з вказанням цінової політики (у вигляді таблиці) і т.д.

Варіант 14. Розробити Web-сторінку кінотеатру. Тут описати перелік фільмів у прокаті (із фотографіями), прайс-лист (у вигляді таблиці) і т.д.

Варіант 15. Розробити сторінку асоціації кінолюбителів. Навести перелік членів асоціації (із фотографіями), перелічити переможців різних кінопремій

за останні 5 років (у вигляді таблиць), розмістити на сторінці анонси найближчих відомих фільмів.

Варіант 16. Розробити сторінку «Сайт наукових новин». Тут описати поточні новини (із фотографіями), склад команди журналістів з інформацією про них (у вигляді таблиці) і т.д.

Варіант 17. Розробити сторінку агентства нерухомості. Розмістити на ній всі наявні пропозиції щодо продажу та купівлі житла (із фотографіями), навести перелік ріелторів, їх номери телефонів та досвід роботи.

Варіант 18. Розробити сторінку автомобільного салону. Навести перелік автомобілів з їх фотографіями та цінами (у вигляді таблиці), розмістити інформацію із подяками від задоволених покупців тощо.

Варіант 19. Розробити сторінку асоціації книголюбів. Тут навести перелік її осередків у країні, склад кожного з осередків (із фотографіями), перелічити рекомендовані асоціацією книги для прочитання (у вигляді таблиці) тощо.

Варіант 20. Розробити сторінку студентської ради. Тут навести склад ради і перелік асоційованих членів із фотографіями (у вигляді таблиці), джерела фінансування, проекти ухвалених рішень, статут тощо.

### *Вимоги та рекомендації до програмної реалізації*

Інформація на Web-сторінці має виглядати зв'язно. Для отримання максимальної оцінки студентам необхідно реалізувати хоча б 100 тегів. Для оцінки добре – не менше 80 тегів. Для задовільної оцінки мінімальна кількість тегів – 60.

Рекомендовано уникати теги `<form>`, `<button>`, `<input>`, оскільки вони потребують додаткових знань, а саме: мови JavaScript, яка буде вивчатись в подальших лабораторних роботах (лаб. № 5+).



### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. Вступне відео по основам HTML:  
<https://www.youtube.com/watch?v=E2oVj1fRIgY>.

Довідкова інформація по наявних тегах HTML:

2. <http://htmlbook.ru/html>.

3. <https://www.w3schools.com/tags/default.asp>.

### *Контрольні питання*

1. Для чого призначена мова HTML?
2. Чи є HTML мовою програмування?
3. Що таке Web-сайт? Які бувають види Web-сайтів?
4. Яка різниця між Web-сторінкою, Web-сайтом і Web-порталом?
5. Що таке тег?
6. Яка черговість виконання вкладень тегів?
7. Яким чином реагує браузер на відсутність закриваючого тегу деякого контейнера?
8. Яку загальну структуру має документ HTML5?
9. Для чого використовують тег HEAD?
10. Для чого використовують тег META?
11. Якими тегами обмежують заголовок документа?
12. Якими тегами обмежують тіло документа?
13. Якими тегами обмежують весь код Web-сторінки?
14. Як на Web-сторінку вставити зображення?
15. Як створити гіперпосилання на інший документ?
16. Які теги використовуються для форматування тексту?
17. Які теги використовуються для формування таблиці?
18. Як в HTML задаються коментарі?
19. Які основні види редакторів коду HTML Ви знаєте?
20. Що таке верифікація?

## Лабораторна робота №2

### Розмітка Web-сторінки засобами HTML. Атрибути

#### Мета

Ознайомитись із синтаксисом, зокрема атрибутами тегів, мови HTML.

#### Завдання

1. Потрібно доповнити Web-сторінку, розроблену на лабораторній роботі №1, атрибутами мови HTML.

#### Вимоги та рекомендації до програмної реалізації

Інформація на Web-сторінці має виглядати зв'язно. Для отримання максимальної оцінки студентам необхідно застосувати хоча б 100 атрибутів (допускаються повторення). Для оцінки добре – не менше 80 атрибутів. Для задовільної оцінки мінімальна кількість атрибутів – 60.

#### Допоміжна таблиця атрибутів HTML

| Ім'я атрибуту | Опис                                                                                                                                        |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| accesskey     | Визначає клавіші, що використовуватимуться для активації або додавання фокусу щодо того чи іншого елемента.                                 |
| alt           | Задає альтернативний текст, якщо зображення не може бути відображено; працює з тегами <area>, <img>, <input>.                               |
| autofocus     | Призначений для автоматичного фокусування на елементі після завантаження сторінки; працює з тегами <button>, <input>, <select>, <textarea>. |
| autoplay      | Призначений для вказання, що аудіо чи відео потрібно відтворити одразу після його готовності; працює з тегами <audio>, <video>.             |
| bgcolor       | Задає колір фону елемента.                                                                                                                  |
| border        | Задає ширину рамки навколо елемента                                                                                                         |
| charset       | Визначає кодування сторінки або сценарію; використовується разом з тегами <meta>, <script>.                                                 |
| checked       | Вказує, чи слід відмітити елемент <input> під час завантаження сторінки.                                                                    |
| cite          | Містить URI, який вказує на джерело цитати чи внесених змін; працює разом з <blockquote>, <del>, <ins>, <q>.                                |

|                 |                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------|
| class           | Задає ім'я чи імена класів деякого елемента.                                                                                   |
| color           | Встановлює колір тексту.                                                                                                       |
| cols            | Визначає ширину видимої частини текстової області <textarea>.                                                                  |
| colspan         | Визначає кількість стовпців, яку займає комірка; працює з тегами <td>, <th>.                                                   |
| contenteditable | Вказує, чи дозволяється редагувати вміст елемента.                                                                             |
| contextmenu     | Задає контекстне меню для елемента.                                                                                            |
| controls        | Вказує, чи потрібно відображати користувачеві кнопки керування стосовно тегів <audio>, <video>.                                |
| coords          | Набір значень, що задає координати області активної ділянки <area>.                                                            |
| data            | Задає URL-адресу ресурсу для об'єкта <object>.                                                                                 |
| datetime        | Вказує дату та час; використовується разом з <del>, <ins>, <time>.                                                             |
| default         | Вказує, що доріжка повинна бути доступна, якщо налаштування користувача не говорять про зворотне; працює разом з <track>.      |
| dir             | Визначає напрямок тексту.                                                                                                      |
| disabled        | Вказує, чи користувач може взаємодіяти з елементами <button>, <fieldset>, <input>, <optgroup>, <option>, <select>, <textarea>. |
| download        | Вказує на те, що посилання використовується для завантаження; використовується з тегами <a>, <area>.                           |
| height          | Задає висоту елемента для <canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>.                                      |
| hidden          | Його функція пов'язана із тим, чи показувати відповідний компонент у вікні браузера.                                           |
| high            | Вказує нижню межу верхнього діапазону для тега <meter>.                                                                        |
| href            | URL пов'язуваного ресурсу; застосовний до тегів <a>, <area>, <base>, <link>.                                                   |
| hreflang        | Вказує мову пов'язаного ресурсу для тегів <a>, <area>, <link>.                                                                 |
| icon            | Присвоює іконку для <menuitem>, яка представлятиме певну команду в контекстному меню.                                          |
| id              | Унікальний ідентифікатор деякого елемента. Значення атрибута має бути унікальним.                                              |
| ismap           | Показує, що зображення є частиною серверної карти зображень; застосовний до теги <img>.                                        |
| keytype         | Визначає тип використовуваного ключа; застосовний до тега <keygen>.                                                            |
| kind            | Вказує вид тексту доріжки для тега <track>.                                                                                    |
| label           | Вказує заголовок текстової доріжки для тегів <track>, <option>, <optgroup>.                                                    |
| lang            | Визначає мову для елемента.                                                                                                    |

|             |                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------|
| list        | Пов'язаний з тегом <datalist>, який містить наперед заданий набір параметрів для тега <input>.                                       |
| loop        | Вказує, чи слід медіа елементам починати програвання спочатку після їх завершення; застосовне до тегів <audio>, <video>.             |
| low         | Вказує межу нижнього діапазону для тега <meter>.                                                                                     |
| manifest    | Задає URL-адресу кешованого маніфесту документа <html>.                                                                              |
| max         | Вказує максимальне значення для тегів <input>, <meter>, <progress>.                                                                  |
| maxlength   | Визначає максимальну кількість символів, допустимих в елементі стосовно тегів <input>, <textarea>.                                   |
| media       | Задає підказку, для якого пристрою пов'язаний ресурс був спроектований; застосовний до тегів <a>, <area>, <link>, <source>, <style>. |
| min         | Вказує мінімальне допустиме значення для тегів <input>, <meter>.                                                                     |
| multiple    | Вказує, що користувач може ввести кілька значень стосовно тегів <input>, <select>.                                                   |
| muted       | Призначений для відключення аудіовиходу для тегів <video>, <audio> в момент завантаження сторінки.                                   |
| open        | Вказує, чи відображати фрагмент, розміщений в <details>...</details> під час завантаження сторінки.                                  |
| optimum     | Вказує оптимальне числове значення для <meter>.                                                                                      |
| pattern     | Визначає регулярний вираз, по якому перевірятимуться вхідні дані елемента <input>.                                                   |
| placeholder | Надає підказку користувачеві щодо того, що можна ввести в поля тегів <input>, <textarea>.                                            |
| poster      | URL-адреса зображення, яке бачить користувач від час завантаження відео або до натискання кнопки запуску відео.                      |
| preload     | Вказує чи завантажувати медіа повністю, чи його частину або не завантажувати взагалі; застосовний до тегів <audio>, <video>.         |
| pubdate     | Вказує, що вказане – є датою та часом публікації <article>.                                                                          |
| readonly    | Вказує, чи можна редагувати елементи <input>, <textarea>.                                                                            |
| rel         | Задає зв'язок між цільовим файлом і даним; застосовний до тегів <a>, <area>, <link>.                                                 |
| reversed    | Впорядковує список <ol> за спаданням.                                                                                                |
| rows        | Визначає кількість рядків у <textarea>.                                                                                              |
| rowspan     | Визначає кількість рядків, які охоплює комірка таблиці; застосовний до тегів <td>, <th>.                                             |
| selected    | Встановлює деякий варіант <option> у якості вже вибраного.                                                                           |
| shape       | Задає форму області <area>.                                                                                                          |
| size        | Визначає розмір видимої частини тегів <input>, <select>.                                                                             |
| sizes       | Задає розміри зображень для тегів <img>, <link>, <source>.                                                                           |

|            |                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| span       | Задає кількість стовпців стосовно тегів <col>, <colgroup>.                                                               |
| spellcheck | Вказує, чи дозволена перевірка правопису для цього елемента.                                                             |
| src        | Задає URL вбудованого вмісту для тегів <audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video>. |
| srcdoc     | Дозволяє лише за допомогою атрибута встановити вміст фрейма <iframe>.                                                    |
| srclang    | Дозволяє задавати мову субтитрів для тега <track>.                                                                       |
| srcset     | Задає набір зображень, які будуть або не будуть відображатись залежно від ситуації. Працює з тегами <img>, <source>.     |
| start      | Визначає початковий номер впорядкованого списку <ol>.                                                                    |
| step       | Задає допустимий інтервал числових значень для <input>.                                                                  |
| tabindex   | Перепишує порядок табуляції.                                                                                             |
| target     | Вказує де відкривати пов'язаний документ; працює з тегами <a>, <area>, <base>.                                           |
| title      | Текст, який відображається у підказці при наведенні на нього мишки.                                                      |
| translate  | Вказує чи варто перекладати вміст елемента.                                                                              |
| type       | Визначає тип елемента для тегів <button>, <embed>, <input>, <link>, <menu>, <object>, <script>, <source>, <style>.       |
| usemap     | Створює зв'язок між зображенням <img> (чи об'єктом <object>) та картою <map>                                             |
| value      | Задає значення для тегів <button>, <input>, <li>, <option>, <meter>, <progress>, <param>.                                |
| width      | Задає ширину елемента <canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>                                     |

### *Контрольні питання*

1. Для чого використовуються атрибути?
2. Які атрибути має тег body?
3. Які використовуються атрибути до тега <a>?
4. Які використовуються атрибути при додаванні зображень в документ?
5. Який атрибут призначений для вирівнювання абзаців?
6. Які атрибути призначені для форматування тексту?
7. Як встановити горизонтальний і вертикальний розміри малюнка?

8. Якщо по якимось причинам Web-сторінці не вдалось завантажити рисунок, то яким чином доцільно пояснювати користувачу, що тут мало бути зображено?

9. Які одиниці вимірювання застосовуються при заданні ширини та висоти об'єктів на сторінці?

10. Які види масштабування рисунків Ви знаєте?

11. Як здійснити перехід на нову Web-адресу, але не покидаючи дану вкладку (при використанні тегу <a>)?

12. Які способи масштабування фоновому рисунку Web-сторінки Ви знаєте?

13. Щодо яких тегів можливо і доцільно застосовувати атрибуту width та height?

14. Яким чином можна побудувати конструкцію у вигляді тунелю при застосуванні таблиць?

15. Яким чином можна побудувати таблицю без використання тегу table?

16. Щодо яких тегів передбачене застосування атрибуту border?

17. Чим відрізняються теги від атрибутів?

18. Чи є такі теги, які можна замінити атрибутами?

19. Яким чином варто поєднувати теги на атрибуту?

20. Яким чином реагує браузер на задання одного і того ж атрибуту деякого конкретного тегу – двічі?

## Лабораторна робота №3 Структурування Web-сайту

### *Мета*

Ознайомитись із підходами до структурування Web-сторінки, Web-сайту, а також файлів і папок.

### *Завдання*

1. Потрібно структурувати Web-сторінку, розроблену на попередніх двох лабораторних роботах, по аналогії до HTML-коду, наведеного за посиланням: [https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction to HTML/Document and website structure](https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure).

2. Використовуючи тег <a> та роботу з фреймами створити Web-сайт на основі вже розробленої Web-сторінки (тобто розподілити контент Web-сайту між кількома вкладками).

### *Вимоги та рекомендації до програмної реалізації*

Інформація на Web-сайті має виглядати зв'язно.

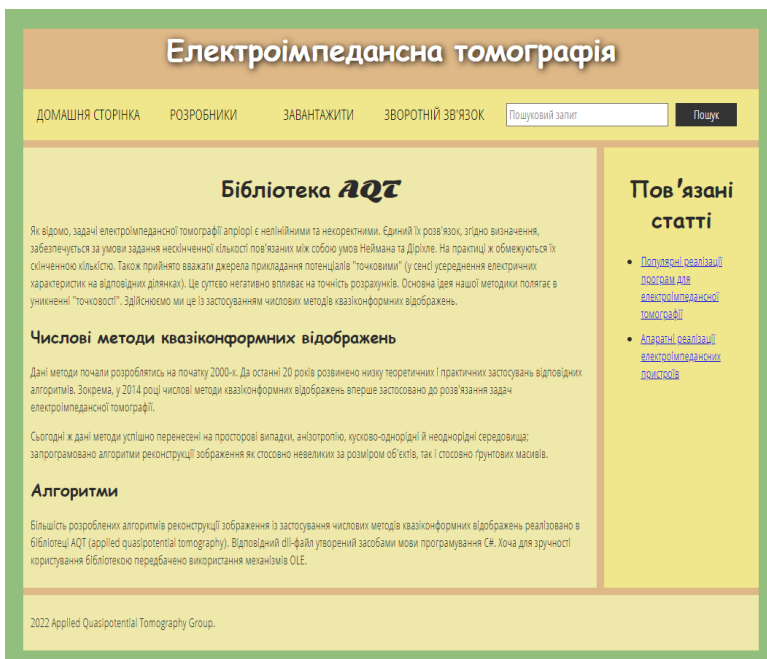
Для отримання максимальної оцінки студентам необхідно доцільно застосувати всі наведені теги <header>, <nav>, <main>, <iframe>, <article>, <section>, <div>, <aside>, <footer>. При цьому, повинно працювати хоча б 4 пункти навігації (<nav>) та має бути вичерпне наповнення Web-сайту.

Отримання оцінки добре можливе за умови доцільного застосування тегів <header>, <nav>, <main>, <aside>, <footer>. При цьому, повинно працювати хоча б 3 пункти навігації (<nav>) та має бути задовільне наповнення Web-сайту.

Для отримання задовільної оцінки потрібно реалізувати хоча б теги <header>, <nav>, <footer>. При

цьому, повинно працювати хоча б 2 пункти навігації (<nav>) та має бути мінімальне наповнення Web-сайту.

Функціонування кнопки «Пошук» можна не реалізовувати. В загальному Web-сторінка повинна виглядати наступним чином:



### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріал за наступним посиланням:

1. <https://www.youtube.com/watch?v=Jkgd2CSfUDU>.

### *Контрольні питання*

1. Які основні компоненти Web-сайту Ви знаєте?
2. Які Вам відомі структурні елементи Web-сайту?
3. Для чого використовують гіперпосилання?



4. Для чого використовують фрейми?
5. Як вкласти один фрейм в інший?
6. Які обмеження мають місце при використанні фреймів?
7. Із яких популярних Web-ресурсів не вдається «підтягнути» відео у вікно фрейма?
8. Як фрейми можуть спростити процес розробки Web-сайту із, наприклад, сотнями Web-сторінок однакової структури?
9. Яким чином створюється Web-сайт із можливістю вибору мови відображення?
10. Чим відрізняються теги `<header>` та `<main>`?
11. Чим відрізняються теги `<header>` та `<footer>`?
12. Яка специфіка взаємного розміщення тегів `<header>`, `<nav>`, `<main>`, `<aside>`, `<footer>`?
13. Яким чином доцільно користуватись тегами `<iframe>`, `<article>`, `<section>`, `<div>` при структуруванні контенту Web-сайту?
14. Які відмінності тега `<div>` від тегів `<header>`, `<nav>`, `<main>`, `<aside>`, `<footer>`?
15. Які із перелічених тегів `<header>`, `<nav>`, `<main>`, `<iframe>`, `<article>`, `<section>`, `<div>`, `<aside>`, `<footer>` можна вкладати один в одний?
16. Чим відрізняються у даному випадку теги `<div name="nav">` та `<nav>`?
17. Чи можливо розмістити тег `footer` візуально у вищій позиції, ніж тег `header`?
18. Як використовується контейнер `<style>...</style>` при структуруванні Web-сторінок?
19. Яким чином можна застосовувати тег `table` при структуруванні Web-сторінки?
20. Як помістити у фрейм Web-сторінку з каталогу, який є вищим у ієрархії за поточний?

## Лабораторна робота №4

### Вступ до CSS

#### *Мета*

Набути навички по стилізації Web-сайтів із використанням каскадної таблиці стилів.

#### *Завдання*

1. Переробити Web-сайт, розроблений на попередній лабораторній роботі, із застосуванням каскадної таблиці стилів (файл style.css).

#### *Вимоги та рекомендації до програмної реалізації*

Усі стилі повинні застосовуватись доречно. У файлах виду \*.html не має безпосередньо виконуватись жодне форматування. Окремо варто звернути увагу на важливий тег display.

Для отримання максимальної оцінки розмір файлу style.css повинен бути більшим, ніж 2 Кб, з реалізованими:

1) загальними селекторами:

- універсальний  
\* {  
}
- по назві тега (по типу елемента)  
тег {  
}
- по класу  
.ім'я\_класу {  
}
- по ідентифікатору  
#ім'я\_ідентифікатора {  
}
- по атрибуту

```
тег[атрибут] {  
}
```

2) селекторами-комбінаторами:

- кома  
тег1, тег2 {  
}
- нащадків  
тег1 тег2 {  
}
- дочірній  
тег1 > тег2 {  
}
- всіх сусідніх елементів  
тег1 ~ тег2 {  
}
- наступного сусіднього елемента  
тег1 + тег2 {  
}

3) селекторами по псевдокласу і псевдоелементу:

- по псевдокласу  
тег:псевдоклас {  
}
- по псевдоелементу  
тег::псевдоелемент {  
}

Варто зазначити, що існує низка підвидів селекторів по атрибуту, по псевдокласу і по псевдоелементу. Серед найбільш важливих варто виділити наступні: :first-child, :last-child, :nth-child(число), :nth-of-type(число), :hover, :focus, :active, :checked, :visited, :link, ::first-line, ::first-letter, ::selection, ::before, ::after, [атрибут="значення"] (після атрибуту часто використовують один із знаків ~, |, ^, \$, \*).

Також комбінація різних селекторів дозволяє отримувати вражаючі візуальні ефекти.

Для отримання оцінки добре необхідно застосувати хоча б половину селекторів кожного виду. Очікуваний розмір файлу style.css – від 1 Кб.

Для отримання оцінки задовільно необхідно застосувати хоча б 2 загальні селектори та 2 селектори-комбінатори. Очікуваний розмір файлу style.css – від 400 Мб.

### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. «Рецепти» CSS <http://htmlbook.ru/faq>.
2. Селектори [https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Selectors](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Selectors).

Відео-уроки:

3. <https://www.youtube.com/watch?v=AmdaLDk-0cU>.
4. <https://www.youtube.com/watch?v=I5Z3FXpPDYk>.

### *Контрольні питання*

1. Для чого використовують CSS?
2. Які версії CSS Ви знаєте?
3. Опишіть основну концепцію CSS?
4. Яка різниця між HTML та CSS?
5. Що є пріоритетнішим для браузера: CSS чи атрибути?
6. Чому прийнято створювати саме окремий файл із стилями?
7. Які Ви знаєте властивості стилю?
8. Якими способами можна змінити стиль елемента?
9. Які властивості CSS призначені для форматування тексту?

10. В яких одиницях можуть задаватись властивості в CSS?

11. Які є загальні правила визначення й присвоєння стилевих властивостей, передбачених в CSS?

12. Як в CSS задаються коментарі?

13. Які Ви знаєте типи селекторів?

14. Чим відрізняються селектори по класу та селектори по ідентифікатору?

15. Що означає запис `#languages li`?

16. Як працює комбінатор «,» у даному випадку `p, input, li`?

17. Чи можна вкладати один вид селектора в інший?

18. Яким чином зручно задавати однакові властивості одразу для кількох елементів?

19. Чи можна лише засобами CSS створити повноцінну Web-сторінку із кнопками, текстовими полями, гіперпосиланнями тощо?

20. Яка різниця між абсолютним та відносним позиціонуваннями?

## **Лабораторна робота №5**

### **Найпростіші застосування JavaScript при розробці Web-сайтів**

#### *Мета*

Ознайомитись із синтаксисом, зокрема основними функціями, мови JavaScript.

#### *Завдання*

1. Засобами JavaScript доповнити розроблений раніше Web-сайт елементами та функціоналом згідно варіанту.

#### *Варіанти до виконання поставленого завдання*

Варіант 1. Розмістити на сторінці поточний Київський, Лондонський, Паризький і Нью-Йоркський час.

Варіант 2. Створити конвертер популярних валют.

Варіант 3. Розробити засоби для розрахунку площі та периметра низки геометричних фігур.

Варіант 4. Реалізувати виведення зображення, адреса до якого вводиться у текстове поле. Коректними вважаються як адреси на локальному диску, так і у глобальній мережі.

Варіант 5. Розробити простий калькулятор, який складається з одного текстового поля і п'яти кнопок (додавання, віднімання, множення, ділення і дорівнює).

Варіант 6. Створити інструментарій для транслітерації введеного з клавіатури тексту згідно сучасних правил транслітерації.

Варіант 7. Реалізувати відображення 10-ти картинок із різними фруктами. Поряд із ними мають розміщуватись текстові поля. Туди користувач вводить назви фруктів. Знизу сторінки має знаходитись кнопка, при натисненні на яку виводиться список невідгаданих фруктів.

Варіант 8. Реалізувати збільшення-зменшення розміру зображення, розміщеного на Web-сторінці. Використати для цього текстові поля та кнопку.

Варіант 9. Створити Web-сторінку, на якій користувач має можливість введення своїх анкетних даних у відповідні поля. Якщо все введено коректно, то виводиться автобіографія користувача, інакше – з'являється повідомлення про невірно введені дані.

Варіант 10. Створити гру «Холодно-тепліше», в якій комп'ютер «загадує» випадкове число. Користувач же намагається відгадати це число. Якщо чергова спроба є більш вдалою, ніж попередня, то користувачу виводиться повідомлення «тепліше», інакше – «холодно».

Варіант 11. Використовуючи тег `input` із типом `radio` реалізувати виведення на екран деякого зображення із різними параметрами ширини, висоти і прозорості.

Варіант 12. Написати скрипт, який дозволяє при натисненні на кнопку `button` у парні секунди (наприклад, о 13:43:10) виводити одне з п'яти наперед заданих похвальних повідомлень, а у непарні (наприклад, о 13:43:11) – єдине гнівне повідомлення.

Варіант 13. Розмістити на сторінці фрейм та 3 кнопки для переключання мови. При натисненні на кнопку у фреймі має з'являтися контент на відповідній мові.

Варіант 14. Розмістити на Web-сторінці кнопку, при натисненні на яку у довільних позиціях на сторінці з'являються ще 2 кнопки. Натиснення на будь-яку з них призводить до появи ще 2 кнопок і т.д.

Варіант 15. Створити форму анкетування лише засобами JavaScript (блок `body` файлів `*.html` повинен бути порожнім). При натисканні на кнопку «Відправити» відбувається перехід на іншу Web-сторінку, де розміщується повідомлення про те, що форму відправлено.

Варіант 16. Розмістити на сторінці кнопку, при натисканні на яку випадковим чином формується певна адреса сайту; сам сайт поміщається у фрейм. Адреса сайту формується на основі випадковим чином згенерованих  $n$  символів;  $n$  задається користувачем сайту у текстовому полі. Передбачити випадок неіснування відповідного сайту.

Варіант 17. Створити «Калькулятор програміста», який складається із двох текстових полів і кнопки. Якщо у перше з текстових полів користувач вводить двійкове число і натискає на кнопку, то в другому текстовому полі бачить десяткове число. Якщо ж навпаки, користувач вводить десяткове число у друге поле, то при натисканні на кнопку має з'являтися відповідне двійкове число у першому полі.

Варіант 18. Написати скрипт, за допомогою якого кожні 10 секунд з'являється певне запитання. Користувач за відведений час повинен дати відповідь. Інакше автоматично задається наступне запитання. У результаті виводиться кількість правильно даних відповідей.

Варіант 19. Створити скрипт, який протягом 10-ти секунд рахує кількість натискань кнопки мишки на кнопку button та кількість одночасно введених символів у текстове поле. Результати виводяться в інших текстових полях.

Варіант 20. Розробити скрипт, який кожну секунду генерує нову цифру і очікує від користувача введення (вгадування) цієї цифри. У позитивному випадку протягом наступної секунди користувач відгадує попередню (невідгадану раніше) цифру і т.д. Ознакою того, що вдалось вгадати цифру є здійснене введення цієї цифри у текстове поле (інакше – візуально нічого не відбувається).

*Вимоги та рекомендації до програмної реалізації*

Уважно

подивитись

відео

<https://www.youtube.com/watch?v=Bluxbh9CaQ0> у часових



проміжках 0:00:00–1:43:50 та 4:41:00–5:47:26. Тут розповідається не лише про основи JavaScript, але й про деякі сучасні тенденції у Web-програмуванні.

Для отримання максимальної оцінки реалізація Web-сторінки повинна повністю відповідати постановці завдання, а також не повинно з'являтися жодних помилок у процесі користування сайтом.

Для отримання оцінки добре реалізація Web-сторінки повинна в більшості відповідати постановці завдання і справлятися із більшістю помилок.

Для отримання оцінки задовільно реалізація Web-сторінки повинна в більшості відповідати постановці завдання і виконувати просту перевірку на коректність.

Для успішного виконання даної лабораторної роботи Вам може знадобитись:

- стиль `position`, який визначає спосіб позиціонування об'єкта на сторінці;
- функція `document.getElementById`, яка повертає посилання на елемент по його ідентифікатору (`id`);
- властивості `елемент.innerHTML`, `елемент.innerText`, `елемент.textContent` для роботи з HTML-розміткою;
- об'єкт `Date`, який призначений для роботи із часом;
- функція `substr`, за допомогою якої можна отримати фрагмент деякого рядка;
- функції `alert`, `prompt` та `confirm`, за допомогою яких можна працювати із модальними вікнами різного виду;
- функції виду `parseInt`, `parseFloat`, `toString` тощо для конвертації типів даних;
- функція `Math.random()` для генерації випадкових чисел;
- запис виду `кнопка.onclick = function() {...}` для програмування реакції натиснення на кнопку.

### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріал за наступним посиланням:

1. <https://www.youtube.com/watch?v=Bluxbh9CaQ0>.

### *Контрольні питання*

1. Що таке JavaScript?
2. Чому мову JavaScript називають C-подібною?
3. Які способи підключення JavaScript Ви знаєте?
4. Які основні функції JavaScript Ви знаєте?
5. Яким чином у процесі роботи з Web-сайтом можна змінити текст, який відображається на кнопці button?
6. Яким чином при порожньому контейнері `<body>...</body>` можливо створити просту Web-сторінку?
7. Яким чином засобами JavaScript можна недопустити введення окремих символів у текстове поле?
8. Що таке `getElementById`?
9. Що означає наступний запис `document.getElementById("Example").textContent = Date()`?
10. Чим відрізняються одна від одної властивості `innerHTML`, `innerText`, `textContent`?
11. Яка різниця між функціями `alert`, `prompt` та `confirm`?
12. Яка різниця між функціями і властивостями?
13. Як згенерувати випадкове число з сегменту `[7.1,25]`?
14. Яка різниця між `==` та `===`?
15. Яка різниця між лапками `«"»` і `«'»`?
16. Яка різниця між `let`, `var` та `const`?
17. Які Ви знаєте типи змінних JavaScript?
18. Як задаються коментарі в JavaScript?
19. Чи виконається без помилок рядок `btn.style.left+btn.style.width/2`? Чому?
20. Навіщо виконувати додавання `"px"` у виразі `btn.style.top=btnTop-dy+5+"px"`?

## **Лабораторна робота №6**

### **Мова JavaScript для забезпечення динамічності сайту**

#### *Мета*

Оволодіти навичками щодо забезпечення динамічності сайту інструментами мови JavaScript.

#### *Завдання*

1. Забезпечити динамічність елементів, поміщених на Web-сторінку у попередній лабораторній роботі. Вони повинні рухатись та/або змінюватись динамічно. Тут мають задіюватись події мишки, наприклад, для перетягування об'єктів на Web-сторінці. Дані в полях (якщо такі передбачені у Вашому варіанті) повинні оновлюватись та перевірятись на коректність у момент введення. Використання тегів `input` та `button` є забороненим.

2. Реалізувати «втікаючу» від курсора кнопку. При натисканні на цю кнопку має з'являтися якесь повідомлення. Цю кнопку можна додати на вкладку «Контакти» (від цього дана вкладка стане менш «нудною»).

3. Реалізувати кнопки додавання фото з локального диску у галерею та сортування фото за певними критеріями.

#### *Вимоги та рекомендації до програмної реалізації*

Не повинно з'являтися жодних помилок у процесі користування сайтом. «Втікаюча» кнопка не повинна рухатись поза межами вікна.

Для успішного виконання даної лабораторної роботи Вам може знадобитись:

- стиль `position`, який визначає спосіб позиціонування об'єкта на сторінці;

- події `onmousemove` та `onmousedown`, за допомогою яких можна визначати реакцію об'єкта на рух мишки та натискання на кнопку мишки, відповідно;

- метод `setInterval`, який призначений для виконання певних дій через кожні задані проміжки часу;

- подія `onmousemove`, в яку присвоюється код реакції на рух мишки понад тим чи іншим елементом;

- інструментарій консолі у режимі розробника для перевірки допущених помилок при написанні скриптів.

Для отримання максимальної оцінки реалізація Web-сторінки повинна повністю відповідати постановці завдання, а також не повинно з'являтися жодних помилок у процесі користування сайтом.

Для отримання оцінки добре реалізація Web-сторінки повинна в більшості відповідати постановці завдання і справлятися із більшістю помилок.

Для отримання оцінки задовільно реалізація Web-сторінки повинна в більшості відповідати постановці завдання і виконувати хоча б просту перевірку на коректність.

### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріал за наступним посиланням:

1. <https://www.youtube.com/watch?v=shXvkszxw0E>.

### *Контрольні питання*

1. З якими помилками Ви стикнулись у процесі виконання даної лабораторної роботи та як їх вирішували?

2. Яким чином доцільно організовувати та здійснювати сортування в JavaScript?

3. Яким чином можна організувати динамічне виведення бажаної кількості фото?

4. Які події клавіатури Ви знаєте?
5. Які події мишки Ви знаєте?
6. Як забезпечити плавне переміщення текстового поля із використанням подій мишки?
7. Які види позиціонування об'єктів на Web-сторінці підходять при бажанні розмістити даний об'єкт у довільних координатах Web-сторінки?
8. У якому випадку на порожній Web-сторінці запис `div.style.top = "125px"` не дасть ніякого ефекту?
9. Як унеможливити введення цифр у деяке текстове поле?
10. Як користуватись консоллю розробника для перевірки помилок JavaScript?
11. Що таке `MouseEvents`?
12. Що таке `addEventListener`?
13. Чи можна пов'язати декілька подій з деяким конкретним елементом?
14. Яка різниця між подіями `onclick`, `onmouseup` та `onmousedown`?
15. Яка різниця між подіями `onkeypress`, `onkeyup` та `onkeydown`?
16. Яка подія спрацьовує при русі курсора мишки поверх елемента?
17. Що означає запис `<p style="color:grey" onmouseover="this.style.color='white'">?`
18. Чи коректним є запис `button.onclick = helloFunc()`?
19. Як застосувати метод `setAttribute` при програмуванні реакції елемента на дії користувача?
20. Яку функцію використовують для видалення обробника тієї чи іншої події?

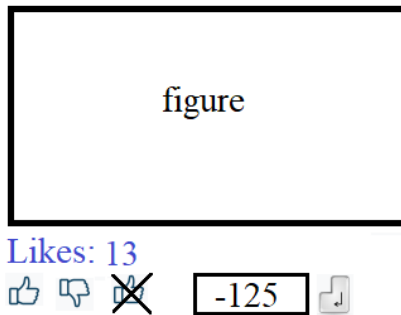
## Лабораторні роботи № 7-8 Фреймворк Vue

### *Мета*

Оволодіти навичками написання Web-сторінок інструментами сучасного фреймворку Vue.

### *Завдання*

1. Засобами Vue розмістити на раніше розробленому Web-сайті 4 кнопки для збільшення, зменшення, обнулення та задання з клавіатури кількості лайків певній фотографії (наприклад, у вигляді, зображеному на рис.).



2. Засобами Vue розробити інструментарій для динамічного керування елементами деякого списку на Web-сайті згідно варіанту.

### *Варіанти до виконання поставленого завдання*

Варіант 1. Реалізувати інтерфейс для додавання та видалення переліку мов і технологій, які вже Ви на сьогодні засвоїли та які ще плануєте засвоїти до кінця навчання.

Варіант 2. Реалізувати інтерфейс для редагування та видалення переліку мов і технологій, які вже Ви на сьогодні засвоїли та які ще плануєте засвоїти до кінця навчання.

Варіант 3. Реалізувати інтерфейс для додавання та редагування переліку мов і технологій, які вже Ви на сьогодні засвоїли та які ще плануєте засвоїти до кінця навчання.

Варіант 4. Реалізувати інтерфейс для додавання та видалення елементів у списку табелів успішності студентів Вашої академічної групи.

Варіант 5. Реалізувати інтерфейс для редагування та видалення елементів у списку табелів успішності студентів Вашої академічної групи.

Варіант 6. Реалізувати інтерфейс для додавання та редагування елементів у списку табелів успішності студентів Вашої академічної групи.

Варіант 7. Реалізувати інтерфейс для додавання та видалення елементів списку фотографій членів футбольної команди.

Варіант 8. Реалізувати інтерфейс для редагування та видалення елементів списку фотографій членів футбольної команди.

Варіант 9. Реалізувати інтерфейс для додавання та редагування елементів списку фотографій членів футбольної команди.

Варіант 10. Реалізувати інтерфейс для додавання та видалення відео-новин (у вигляді списку) деякого новинного сайту.

Варіант 11. Реалізувати інтерфейс для редагування та видалення відео-новин (у вигляді списку) деякого новинного сайту.

Варіант 12. Реалізувати інтерфейс для додавання та редагування відео-новин (у вигляді списку) деякого новинного сайту.

Варіант 13. Реалізувати інтерфейс для додавання та видалення елементів списку пісень довільної музичної групи.

Варіант 14. Реалізувати інтерфейс для редагування та видалення елементів списку пісень довільної музичної групи.

Варіант 15. Реалізувати інтерфейс для додавання та редагування елементів списку пісень довільної музичної групи.

Варіант 16. Реалізувати інтерфейс для додавання та видалення елементів списку заборонених на сьогодні програмних продуктів російського виробництва.

Варіант 17. Реалізувати інтерфейс для редагування та видалення елементів списку заборонених на сьогодні програмних продуктів російського виробництва.

Варіант 18. Реалізувати інтерфейс для додавання та редагування елементів списку заборонених на сьогодні програмних продуктів російського виробництва.

Варіант 19. Реалізувати інтерфейс для додавання та видалення переліку завдань, які Ви маєте виконати на сьогодні.

Варіант 20. Реалізувати інтерфейс для редагування та видалення переліку завдань, які Ви маєте виконати на сьогодні.

### *Вимоги та рекомендації до програмної реалізації*

Інформація на Web-сторінці має виглядати зв'язно. Для отримання максимальної оцінки студентам необхідно виконати усі завдання лабораторної роботи та вміти пояснити написаний код із використанням професійної термінології. Для оцінки добре – необхідно реалізувати хоча б половину із того функціоналу, що вимагається у



постановках кожного із завдань. Для задовільної оцінки достатньо повністю виконати перше завдання.

### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. <https://www.youtube.com/watch?v=94fHz4w65PY>.
2. <https://www.youtube.com/watch?v=p059z-0JTFg>.
3. <https://ru.vuejs.org/v2/guide/index.html>.

### *Контрольні питання*

1. Що таке фреймворк Vue?
2. Що таке JSON?
3. Який зв'язок між Vue та JSON?
4. Що означає послідовність дужок {{ у Vue?
5. Як підключити фреймворк Vue?
6. Що таке v-bind у Vue?
7. Як користуватись v-bind у Vue?
8. Що таке v-for у Vue?
9. Як користуватись v-for у Vue?
10. Що таке v-on у Vue?
11. Як користуватись v-on у Vue?
12. Що таке v-model у Vue?
13. Як користуватись v-model у Vue?
14. Що таке mount та як ним користуватись?
15. Як створюються константи у Vue?
16. Що таке фільтри у Vue?
17. Які Ви знаєте директиви у Vue?
18. Що таке Vue-ресурс?
19. Як створюється двостороння прив'язка у Vue?
20. Які структури даних доцільно використовувати для динамічного додавання, редагування і видалення списку осіб та їх номерів телефону на сторінці засобами Vue?

## Лабораторні роботи № 9-10

### Фреймворк Express

#### *Мета*

Навчитись створювати клієнт-серверні додатки на фреймворках Vue та Express.

Дана лабораторна робота є завершальною у курсі Web-програмування. Вона вимагає використання знань, вмінь та навичок, набутих на всіх попередніх лабораторних роботах. У результаті має бути реалізований клієнт-серверний додаток, який реалізує Frontend (клієнтську частину) та Backend (серверну частину) на відповідних фреймворках: Vue та Express.

#### *Завдання*

1. Інструментами фреймворків Express та Vue реалізувати можливість додавання коментарів до фотографій галереї, розробленої в лабораторній роботі №6.
2. Реалізувати можливість редагування та видалення коментарів до фотографій.
3. Засобами Express та Vue забезпечити можливість ставити та прибирати лайки біля фотографій.

#### *Вимоги та рекомендації до програмної реалізації*

Інформація на Web-сторінці має виглядати зв'язно. Для отримання максимальної оцінки студентам необхідно виконати усі завдання лабораторної роботи та вміти пояснити написаний код із використанням професійної термінології. Для оцінки добре – необхідно виконати перші 2 завдання. Для задовільної оцінки достатньо повністю виконати перше завдання.

Для успішного виконання лабораторної роботи необхідно:

- виконати усі попередні лабораторні роботи;
- розуміти архітектуру REST API;
- встановити Node.js з офіційного сайту <https://nodejs.org/en/>;
- створити файл виду frontend.js, де імпортувати Vue з віддаленого сервера;
- проініціалізувати проект;
- встановити пакети express та nodemon;
- підключити модулі express та path.

#### *Додаткові матеріали*

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. [https://www.youtube.com/watch?v=lzQIhjElV\\_g](https://www.youtube.com/watch?v=lzQIhjElV_g) (проміжок 0:00-18:00).
2. <https://www.youtube.com/watch?v=3aGSqasVPsI>.
3. <https://youtu.be/9EtkpCzqZu0>.
4. [https://www.youtube.com/watch?v=lzQIhjElV\\_g&t=1080s](https://www.youtube.com/watch?v=lzQIhjElV_g&t=1080s).

#### *Контрольні питання*

1. Що таке Backend?
2. Що Frontend?
3. Чим відрізняються Frontend та Backend?
4. Які Ви знаєте фреймворки стосовно JavaScript?
5. У чому принципова різниця між Vue та Express?
6. Яким чином пов'язані Node та Express?
7. Що таке архітектура REST API?
8. Яким чином ініціалізується проект на Express?
9. Які модулі для Node та Express Вам відомі?
10. Які пакети для Node та Express Вам відомі?

11. Яка різниця між "development" та "production" в Express?
12. Чи може функція-обробник бути асинхронною?
13. Що таке режим кластера?
14. Що означає `express().get(...)`?
15. Що означає `express().Router(...)`?
16. Які Ви знаєте спеціальні функції для обробки помилок?
17. Що таке механізм шаблонів в Express?
18. Яким може бути результат виконання рядка коду `app.listen(process.env.port || 3000)`?
19. Навіщо необхідно вказувати точку входу Web-застосунку?
20. Чим відрізняються методи `get` та `post`?

## Рекомендована література

1. Спірінцев В. В., Гнатушенко В. В., Волковський О. С. Web-технології та Web-дизайн: HTML, CSS : навчальний посібник. Дніпро : «Ліра», 2017. 163 с.
2. Мейер Э., Уэйл Э. CSS: полный справочник. Изд. 4-е / ред. В.Р. Гинзбурга; пер. И.В. Василенко. Санкт-Петербург : ООО «Диалектика», 2019. 1088 с.
3. Браун Э. Изучаем JavaScript: руководство по созданию современных вебсайтов. Изд. 3-е / ред. и пер. В. А. Коваленко. Санкт-Петербург : ООО «Диалектика», 2020. 368 с.
4. Дакетт Д. Javascript и jQuery. Интерактивная веб-разработка / пер. М.А. Райтман. Москва: Издательство «Э», 2017. 640 с.
5. Пасічник В. В., Пасічник О. В. Веб-дизайн. Львів : Магнолія 2006. 2018. 520 с.
6. Хэнчеп Э., Листуон Б. Vue.js в действии / пер. ООО Издательство «Питер». Санкт-Петербург : Питер, 2019. 304 с.
7. Macrae C. Vue.js: Up and Running. Sebastopol: O'Reilly Media, Inc., 2018. 159 p.
8. Янг А., Мек Б., Кантелон М. Node.js в действии. Изд. 2-е / пер. ООО Издательство «Питер». Санкт-Петербург : Питер, 2018. 432 с.
9. Mardan A. Practical Node.js. San Francisco: Apress Media LLC, 2018. 505 p.
10. Дронов В. А. Django 2.1. Практика создания веб-сайтов на Python. Санкт-Петербург : БХВ-Петербург, 2019. 672 с.
11. Кузьмініх В. О., Тараненко Р. А. Основи управління ІТ проектами : навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2019. 75 с.

12. Матвеева Л. Г., Никитаева А. Ю. Управление ИТ-проектами : учебное пособие. Таганрог : Издательство Южного федерального университета, 2016. 227 с.

13. Левушкина С. В. Управление проектами : учебное пособие. Ставрополь : Секвойя, 2017. 204 с.

14. <http://htmlbook.ru>.

15. <https://www.w3schools.com>.

16. <https://metanit.com/>.

17. <https://developer.mozilla.org/ru/docs/Learn>.

18. <https://w3schoolsua.github.io/js/index.html>.

### Корисні посилання

1. [http://validator.w3.org/#validate\\_by\\_upload](http://validator.w3.org/#validate_by_upload) – валідатор HTML.

2. [https://courses.prometheus.org.ua/courses/course-v1:Prometheus+CS50+2021\\_T1/course/](https://courses.prometheus.org.ua/courses/course-v1:Prometheus+CS50+2021_T1/course/) – Гарвардські курси по Web (HTML, CSS, JS, Python...).

3. <https://www.youtube.com/watch?v=PJRDpA9NpDs&list=PLL-k0Ff5RfqXnwdDG61WqZ2j3KXUPnfmq&index=1> – технологія ASP.

4. <https://wordpress.com>, <https://sites.google.com> – популярні «конструктори» власних сайтів.