

навчальна програма з веб-розробки

для вчителів інформатики





Шановний вчителю!

Перед Вами – навчальна програма курсу з веб-розробки. Дана програма була розроблена в рамках соціального освітнього проекту WOW Teachers з підвищення кваліфікації шкільних вчителів інформатики.

Програма має на меті навчити слухача основам розробки сучасних веб-сторінок із застосуванням найпоширеніших методик та інструментів.

Програма виклажається у форматі [blended learning](https://uk.wikipedia.org/wiki/%D0%97%D0%BC%D1%96%D1%88%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F) та передбачає проведення викладачем 16 аудиторних занять тривалістю 2 асторономічні години кожне протягом 8 тижнів (2 місяців). Передбачається, що на кожне таке заняття слухачі приходитимуть самостійно вивчивши передбачені цією програмою обсяги теоретичного матеріалу та (там де це передбачено) виконавши доманє завдання.

В залежнсті від складності та логічності викладання, кожне аудиторне заняття покриває від 1 до 4 виокремлених тем. За потреби деякі заняття викладач може ділити на окремі частини, зважаючи на успішність та рівень засовєння матеріалу групою.

Кожне із занять містить практичні рекомендації щодо його проведення, теоретичний матеріал для самостійного опрацювання слухачами та домашні завдання для них. Крмі того, в планах тем ви знайдете і посилання на приклади завдань для опрацювання під час аудиторних занять в групі.

Сподіваємось, що дана програма стане першим кроком у виведенні уроків інформатики на новий якісний рівень!

З повагою,

Команда Фонду BrainBasket

# ЗМІСТ

[ЗМІСТ 3](#_Toc516676514)

[ЗАНЯТТЯ 1 Знайомство з групою. Знайомство з веб-розробкою 4](#_Toc516676515)

[ЗАНЯТТЯ 2 Більше теорії про HTML. Знайомство зі стилями 7](#_Toc516676516)

[ЗАНЯТТЯ 3 Заглиблення в CSS 10](#_Toc516676517)

[ЗАНЯТТЯ 4 Продовження селекторів. Посилання. Хостінг 13](#_Toc516676518)

[ЗАНЯТТЯ 5 Блоки в HTML. Модель «ящику» (box model). Основи позиціонування 16](#_Toc516676519)

[ЗАНЯТТЯ 6 Продовження блоків та позиціонування. Float 18](#_Toc516676520)

[ЗАНЯТТЯ 7 Flexbox 20](#_Toc516676521)

[ЗАНЯТТЯ 8 Адаптивний дизайн. Знайомство з bootstrap 22](#_Toc516676522)

[ЗАНЯТТЯ 9 Продовження bootstrap 26](#_Toc516676523)

[ЗАНЯТТЯ 10 Знайомство з Javascript 30](#_Toc516676524)

[ЗАНЯТТЯ 11 Огляд синтаксису javascript та роботи з DOM 33](#_Toc516676525)

[ЗАНЯТТЯ 12 Масиви та об’єкти в javascript. JSON 36](#_Toc516676526)

[ЗАНЯТТЯ 13 Отримання даних з серверу. Обробка подій. Основи HTML форм 40](#_Toc516676527)

[ЗАНЯТТЯ 14 Робота з даними користувача. Продовження форм. Огляд відправки даних на сервер 43](#_Toc516676528)

[ЗАНЯТТЯ 15 Огляд поточного стану технологій веб-розробки. Заключне спілкування 46](#_Toc516676529)

[КУРСИ ДЛЯ ПАРАЛЕЛЬНОГО ВИВЧЕННЯ 49](#_Toc516676530)

[КОПІРАЙТ 50](#_Toc516676531)

# ЗАНЯТТЯ 1 Знайомство з групою. Знайомство з веб-розробкою

**ТЕМИ**

**1. Знайомство з WEB та розробкою сайтів**

* Огляд та історія WEB та HTML
* Поверхнево схема роботи інтернет та WWW
* HTML - це лише структура
* Інструментарій front-end розробника

**Практика:**

Продемонструвати роботу з рекомендованими редакторами коду, мінімальна сторінка “привіт, світ!” ([приклади виконаних завдань](https://drive.google.com/drive/u/1/folders/1RI8HptvWKQtnRUgxL7RQ_gDfeUzJr5dd))

**2. Категорії контенту та базові елементи**

* Content categories
* Div, span
* Заголовки
* Семантичні елементи

**Практика:**

Створити 3 типи заголовків на веб-сторінці ([приклади виконаних завдань](https://drive.google.com/drive/u/1/folders/1RI8HptvWKQtnRUgxL7RQ_gDfeUzJr5dd))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Знайомимось з групою. Ментор розповідає про себе та просить розповісти про себе учасників курсу.** Базові варіанти запитань:
   * Хто звідки (з якої школи, чому навчає (якщо вчитель) або які улюблені заняття взагалі по життю та які улюблені предмети/галузі знань (якщо учень)
   * Наскільки комфортно відчуває себе з тематикою курсу (можливо, вже щось вивчав, або робив практичне…)
   * Що очікує від курсу, чому сподівається, що йому буде цікаво – корисно.
2. **Обговорення на тему термінів *internet*, *web* та *html***

* Обговорюємо, хто як розуміє ці слова. Коротко згадуємо історію виникнення інтернет як мережі для військових та науковців. Наголошуємо, що інтернет – це мережа. Коротко описуємо історію виникнення web (разом з html та http).

<https://en.wikipedia.org/wiki/History_of_the_Internet><https://webfoundation.org/about/vision/history-of-the-web/>

* Обговорюємо поняття та роль браузера. Згадати давнину «коли сайти були інформаційними сторінками», та розповісти про еволюцію браузерів та сайтів від простих статичних сторінок до повноцінних «товстих застосувань» в середині браузера, якими є багато сучасних сайтів, та браузерних ігор. Взяти який-небудь відомий приклад, який вам подобається (facebook, веб-інтерфейс gmail, або навіть сторінку пошуку гугла) – і трохи відкрити, скільки  там всього відбувається під ковдрою.

1. **Crash-знайомство з html**

* Не вдаючись в подробиці, демонструємо створення мінімалістичної веб-сторінки. Як варіант, вона складається з лише з html, body та напису «Hello, Teachers!». Пропонуємо всім повторити на своїх комп’ютерах. Вводимо (не заглиблюючись в «html це xml», хоча і сказавши про це) поняття тегу, кажемо про те, що вони можуть бути вкладеними. Занурюємо текст у тег h1.
* Коротко розповідаємо про те, що використання зручного редактору дуже допомагає зекономити час та зменшити ймовірність помилок. Пропонуємо всім редактор, який вважаємо найкращим. Для тих, хто не має власного обґрунтованого вибору редактору для веб-розробки, використання вказаного ментором редактору вказуємо як обов’язкове.
* Поверхнево демонструємо, що подивитись структуру html можна використовуючи консоль браузера.
* Поверхнево демонструємо ще кілька базових тегів з тим, щоб можна було дати домашнє завдання по створенню простенької заготовки веб-сторінки. Як варіант – додаємо до сторінки блок тексту, картинку та список - тобто демонструємо теги:
  + - * заголовок (h1)
      * картинка (img) з атрибутом src (без подробиць кажемо, що це атрибут, і там ім’я файлу)
      * список без нумерації (ul)
      * абзац (параграф) (p)

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Mozilla Developer Network (MDN) «Як працює мережа» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/How_the_Web_works)
* [Mozilla Developer Network (MDN) «Вступ до HTML» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML)
* [Mozilla Developer Network (MDN) «Категорії вмісту» (укр/англ)](https://developer.mozilla.org/uk/docs/Web/Guide/HTML/Content_categories)
* [Mozilla Developer Network (MDN) «HTML text fundamentals» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals)
* [Mozilla Developer Network (MDN) «Advanced text formatting» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting)
* [Mozilla Developer Network (MDN) «Дебагінг HTML» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML/Debugging_HTML)
* [Mozilla Developer Network (MDN) «Document and website structure» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure)
* [Курс з веб-розробки на W3Schools, тема “HTML Elements” (англ)](https://www.w3schools.com/html/html_elements.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Attributes” (англ)](https://www.w3schools.com/html/html_attributes.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Headings” (англ)](https://www.w3schools.com/html/html_headings.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Paragraphs” (англ)](https://www.w3schools.com/html/html_paragraphs.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Text Formatting” (англ)](https://www.w3schools.com/html/html_formatting.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Comments” (англ)](https://www.w3schools.com/html/html_comments.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Images” (англ)](https://www.w3schools.com/html/html_images.asp)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* Попрактикуватися в створенні власного html документу із різними заголовками та базовими елементами у різних редакторах.
* Як варіант домашнього завдання можна запропонувати створити прототип «сторінки товару з веб-магазину» - а саме сторінку, яка містить заголовок товару, картинку товару, блок з текстовим його описом та блок із списком характеристик.
* Наголошуємо, що не потрібно (якщо учасник курсу не мав попередніх вмінь роботи з html-css) боротися за гарний його вигляд – головна мета створити сторінку, спробувати понабирати теги, попрацювати з редактором, інспектором і тд.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Історія мережі Інтернет на Вікіпедії (англ)](https://en.wikipedia.org/wiki/History_of_the_Internet)
* [Історія мережі Інтернет на WWW Foundation (англ)](https://webfoundation.org/about/vision/history-of-the-web/)
* [Текстовий редактор Notepad++ (офіційний веб-сайт, англ)](https://notepad-plus-plus.org/)
* [Оглядова відеоінструкція з коритування Notepad++ (рос)](https://www.youtube.com/watch?v=yuJQDqlT2_M)
* [Про організацію W3C (англ. World Wide Web Consortium, W3C) на Вікіпедії (укр)](https://uk.wikipedia.org/wiki/W3C)
* [Про організацію WHATWG (англ. Web Hypertext Application Technology Working Group) на Вікіпедії (укр)](https://uk.wikipedia.org/wiki/WHATWG)
* [Актуальні рекомендації (специфікації) до вмісту та структури HTML документів від W3C “Semantics, structure, and APIs of HTML documents” (англ)](https://www.w3.org/TR/html52/dom.html#kinds-%C2%AD%E2%80%90of-%C2%AD%E2%80%90content)
* [Актуальні стандарти HTML від WHATWG (англ)](https://html.spec.whatwg.org/multipage/)
* [Онлайн валідатор HTML від W3C (англ)](https://validator.w3.org/)
* [Війна браузерів на Вікіпедії (англ)](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B9%D0%BD%D0%B0_%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80%D1%96%D0%B2)
* [Світова статистика популярності браузерів від W3Schools (англ)](https://www.w3schools.com/browsers/default.asp)
* [Браузер Google Chrome (рос/англ)](https://support.google.com/chrome#topic=7438008)
* **Також можна починати опрацьовувати матеріали корисних безкоштовних курсів:**
  + - * [Coursera, курс “Responsive Website Basics: Code with HTML, CSS, and JavaScript” від University of London (англ, є український транскрипт)](https://www.coursera.org/learn/website-coding)
      * [Курс «Основи Web UI розробки» від Prometheus (укр)](https://courses.prometheus.org.ua/courses/course-v1:LITS+114+2017_T4/courseware/b52fdaef66554d7cbb90805509f00c03/8a916d3d0ed94fad9af8b9b38be217ba/)
      * [edX, курс “Microsoft Professional Orientation: Front End Web Development” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV237x+4T2017/course/)

# ЗАНЯТТЯ 2 Більше теорії про HTML. Знайомство зі стилями

**ТЕМИ**

**3. Базові елементи – 2**

* Більше про блоки - div, span
* Нумеровані та не-нумеровані списки
* HTML entities

**Практика:**

Створити різні варіанти списків та їх ієрархії ([приклади виконаних завдань](https://drive.google.com/open?id=1lrpZonAuXe12eumPyRsWTkSWjjDuyRdk))

**4. Вступ до CSS (каскадних таблиці стилів)**

* Синтаксис CSS
* Прості селектори

**Практика:**

Здійснити просту стилізацію вмісту html сторінки ([приклади виконаних завдань](https://drive.google.com/open?id=1OlHuLPyhUVAuTz9aaoaHtXgLGaSelCyc))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково (в охочих) дивимось та обговорюємо разом домашнє завдання**
   * В тому числі показуємо, що варто періодично перевіряти коректність html – наприклад, валідатором, або налаштувавши редактор (це може бути складно). Можна подумати про безкоштовну версію WebStorm для учнів та викладачів.
   * Часто учасники використовують «теги форматування», наприклад «b», «i», «center» - підкреслити, що це вважається застарілим підходом, і рекомендовано для цього використовувати styles, про які поговоримо пізніше.
2. **Коротко, але детальніше про склад структуру html документу та поняття тегу (в теорії всі повинні були вивчити це самостійно, наголосити на цьому!)**
   * Теги html, head, title, body. Знову про можливість вкладення тегів, показати трохи глибше вкладення. Коректність вкладення.
   * Атрибути тегів, на прикладі img src (який вже бачили та використовували).
   * Тег «посилання» - «a» - з атрибутом «src» - без заглиблення в деталі, базовий вигляд з посиланням на файлик html, який лежить на тому ж рівні, що і вихідний.
3. **Crash-вступ до css**
   * Трохи мотивації – про те, що при будь-якій розробці важливо розділяти концепції, і в тому числі структуру документу від властивостей зовнішнього вигляду – і для цього було створено окремий механізм стилів, який дозволяє один і той самий тег змусити виглядати по-різному.
   * Показуємо, що є атрибут style, який дозволяє задавати багато параметрів стилю. Приклади – як мінімум, колір, розмір шрифту та вирівнювання тексту по центру.
   * Теги div та span як варіанти обгорток будь-якого вмісту для застосування стилів. Без деталей – просто сказати, що коли хочемо, щоб було на окремому рядочку – то div, а коли потрібно задати стиль, наприклад, слову в середині речення – то span.
   * Експериментуємо на тему стилізації разом з групою, над сторінкою з попереднього заняття.
4. **Застосування стилю через атрибут class**
   * Звертаємо увагу групи, що, коли параметрів стилю стає багато, дуже не зручно працювати з ними, коли всі вони в атрибуті style одного тегу.
   * Демонструємо варіант прописування тегу атрибуту class, та винесення опису класу в блок style в середині блоку head. В даний момент не заглиблюємось в тему селекторів, просто кажемо, що насправді те, що ми зробили – це лише частинка дуже потужного механізму селекторів. Переробляємо наш приклад, виносячи все з вбудованих стилів в заголовок сторінки, та назначаючи класи відповідним тегам. Наголошуємо на тому, що імена класів важливі, і повинні бути змістовними – це підвищує читабельність та можливість підтримки коду.
   * Показуємо, як в консолі хрому дивитись, які стилі застосовано до елементу html – в першу чергу з метою пошуку помилок (ситуацій, коли «мій стиль не спрацьовує!»).

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “HTML Block and Inline Elements” (англ)](https://www.w3schools.com/html/html_blocks.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Lists” (англ)](https://www.w3schools.com/html/html_lists.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Entities” (англ)](https://www.w3schools.com/html/html_entities.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Syntax and Selectors” (англ)](https://www.w3schools.com/css/css_syntax.asp)
* [Mozilla Developer Network (MDN) «CSS syntax» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Syntax)
* [Mozilla Developer Network (MDN) «Simple selectors» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Simple_selectors)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* Переробити та покращити приклад з попереднього завдання, використовуючи стилі, повністю позбавившись від застарілих тегів форматування. Стилі застосовувати, використовуючи селектор class, і розміщувати в заголовку документа.
* Як варіант – спробувати стилізувати сторінку товару з відгуками так, щоб вона була дещо схожа на сторінку товару з відгуками якого-небудь відомого інтернет магазину. Знову таки – не в усьому, але добитись хоча б часткової подібності. Використовувати документацію по css.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [База зразків CSS стилів “CSS Zen Garden” (англ)](http://csszengarden.com/)
* [Ресурс “Frontend Masters”, онлайн туторіал “ Front-End Developer Handbook 2018”, розділ “Learn HTML & CSS” (англ)](https://frontendmasters.com/books/front-end-handbook/2018/learning/html-css.html)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Что такое CSS?» (рос)](https://devionity.com/ru/courses/css-fundamentals/what-is_css)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Селекторы (базовые)» (рос)](https://devionity.com/ru/courses/css-fundamentals/selectors-simple)

# ЗАНЯТТЯ 3 Заглиблення в CSS

**ТЕМИ**

**5. Більш потужні селектори**

* Дочірній елемент
* Нащадок в ієрархії
* Селектор класу
* Групування селекторів

**Практика:**

Здійснити більш складну стилізацію вмісту сторінки ([приклади виконаних завдань](https://drive.google.com/open?id=11ZRFBki8gFnWumm7OZ0reKNFDP5iyEG5))

**6. Варіанти розміщення стилів**

* Розміщення стилів в атрибуті style, в блоку style та в зовнішньому файлі
* Переваги та недоліки кожного з варіантів

**Практика:**

Опановуємо роботу з атрибутом style з різними підходами ([приклади виконаних завдань](https://drive.google.com/open?id=1bTEwN-MNiaVwd9i_-46uxTyObnVoCmUZ))

**7. Приорітетність та успадкування стилей**

* Каскадність та специфічність стилю
* Успадкування стилю

**Практика:**

Застосовуємо каскадність та успадкованість стилів ([приклади виконаних завдань](https://drive.google.com/open?id=1BM_Uladv1yaki42fYDlIOlWXZPQAJSOg))

**8. Стилізація тексту**

* Параметри шрифта
* Стилізація тексту

**Практика:**

Працюємо зі шрифтами в документі([приклади виконаних завдань](https://drive.google.com/open?id=1LOwlwpOs6wE_fvvSuy9TDBcq8862dpXk))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково (в охочих) дивимось та обговорюємо разом домашнє завдання**

* Обговорити найчастішу проблему – коли порушено синтаксис опису класів та правил css (забута або зайва фігурна дужка, крапка з комою і тд. Показати, що браузер при цьому не видає помилки, але користуючись консолью браузера, її легше відслідкувати.

1. **Повторюємо частину щодо селекторів по класу з попереднього заняття.**

* Ще раз про те, що можна завжди обгортати в div та span потрібні нам шматки html, щоб застосувати стиль, та про те, що назви класів обираємо свідомо.

1. **Демонструємо варіант виносу правил css в окремий файл.**

* Описуємо мотивацію, яка за цим стоїть. Разом з групою переробляємо попередній приклад на варіант «файл стилів окремо».

1. **Вводимо поняття селектору css.**

* Розповідаємо про те, що до цього моменту використовували селектор по атрибуту class. Демонструємо селектор по id. Селектор по тегу. Розповідаємо, в яких випадках який краще застосовувати, і чому (на погляд ментора).
* Демонструємо проблему «конфлікту» селекторів (коли підходять кілька одночасно). Вводимо поняття специфічності селекторів (поки що базово – на рівні «селектор за ІД перемагає в селектору за тегом» і т.д.
* Демонструємо ситуацію успадкування стилю.
* Ретельно граємось с цим усім, весь час підкреслюючи та показуючи, як важливо та зручно використовувати консоль браузера, щоб розбиратися, які стилі та чому застосувались до того чи іншого елементу.
* По ходу додаємо нові властивості css, які потрібні для нашого прикладу. Підкреслюємо, що на заняттях ми не будемо розбирати «всі можливі властивості css» - вони залишаються на самостійне опрацювання, на основі документації.
* Як варіант прикладу, можна створювати горизонтальне верхнє меню.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “CSS Combinators” (англ)](https://www.w3schools.com/css/css_combinators.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Attribute Selectors” (англ)](https://www.w3schools.com/css/css_attribute_selectors.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Styles - CSS” (англ)](https://www.w3schools.com/html/html_css.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Colors” (англ)](https://www.w3schools.com/html/html_colors.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML Styles” (англ)](https://www.w3schools.com/html/html_styles.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Text” (англ)](https://www.w3schools.com/css/css_text.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Fonts” (англ)](https://www.w3schools.com/css/css_font.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Text Effects” (англ)](https://www.w3schools.com/css/css3_text_effects.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Web Fonts” (англ)](https://www.w3schools.com/css/css3_fonts.asp)
* [Mozilla Developer Network (MDN) «Attribute selectors» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Attribute_selectors)
* [Mozilla Developer Network (MDN) «Combinators and groups of selectors» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Combinators_and_multiple_selectors)
* [Mozilla Developer Network (MDN) «How CSS Works» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works)
* [Mozilla Developer Network (MDN) «Cascade and inheritance» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Cascade_and_inheritance)
* [Mozilla Developer Network (MDN) «Fundamental text and font styling» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_text/Fundamentals)
* [Mozilla Developer Network (MDN) «Web fonts» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_text/Web_fonts)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Перестилізувати сторінку з попереднього завдання – використати селектори за тегом та успадкування так, щоб зменшити кількість наборів правил.
  + Винести стилі в окремий файл
  + Додати то цієї сторінки верхнє меню (якщо на занятті ментор обрав його як приклад. Інакше – додати щось інше, на розсуд ментора).
  + Для просунутих учасників – розібратися як, і додати випадаюче підменю (при наведенні на пункт меню випадають підпункти). Сказати, що для цього можна використати псевдо-класс, який буде змінювати властивість display – а далі «дивитись документацію». Для середньо – просунутих – нехай при наведенні мишею на меню, елемент змінює колір тексту (також псевдо-клас).

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Подключение CSS к HTML документу» (рос)](https://devionity.com/ru/courses/css-fundamentals/applying-css-to-html-document)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Селекторы (сложные)» (рос)](https://devionity.com/ru/courses/css-fundamentals/selectors-composite)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Форматирование текста с помощью CSS» (рос)](https://devionity.com/ru/courses/css-fundamentals/formatting-text-with-css)

# ЗАНЯТТЯ 4 Продовження селекторів. Посилання. Хостінг

**ТЕМИ**

**9. Посилання**

* Зовнішні посилання
* Внутрішні посилання
* Посилання всередині сторінки
* Посилання картинок

**Практика:**

Опрацювати різні варіанти використання посилань ([приклади виконаних завдань](https://drive.google.com/open?id=1PRtVGwuy3gkV5Lm0MwfKlwwTG5ljJWcX))

**10. Псевдо-класи**

* Pseudo-classes

**Практика:**

Порівнюємо документи до та після застосування псевдо-класів ([приклади виконаних завдань](https://drive.google.com/open?id=1V7c7cSCfLWdZLxYR5n0vUdzH_qDCNRv5))

**11. Хостінг**

* Варіанти розміщення сайту (прості). Можливо, згадка про спец-сервіси, типу AWS та Azure.
* Хостінг на github pages як швидкий простий та безкоштовний варіант розміщення.
* Мінімальні відомості про git та github.

**Практика:**

Разом з учнями створюємо обліковку та викладаємо сторінку.

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково (в охочих) дивимось та обговорюємо разом домашнє завдання.**
2. **Розповідаємо та демонструємо складні (складені) селектори.**

* Тег + клас, «діти елементу» і т.д. Комбінування селекторів.
* По ходу знову згадуємо та детальніше пояснюємо специфічність. Експериментуємо з нею в розумних межах. Повні правила визначення специфічності та комбінування селекторів залишаємо на самостійне вивчення по документації.
* Демонструємо, що нормальною практикою є вказувати елементу кілька класів одночасно.
* Як варіант, все це нормально демонструється на стилізації верхнього горизонтального меню з попереднього заняття. Наприклад, class=”menu-item active”.

1. **Псевдо класи. Призначення, ідея, кілька прикладів. Детально про псевдо клас «hover».**

* Експериментуємо з ним на нашому прикладі, з усіма бажаючими. Також непогано демонструється на прикладі меню – можна зробити підсвітку при наведенні, а потім більш цікавий приклад - випадання підменю при наведенні. Доводиться зробити елементи з тегами «а».

1. **Посилання (тег «a»)**

* Детальніше розповідаємо та експериментуємо з посиланнями. Види посилань, приклади (можна продовжувати те саме меню).
* Трошки сказати про протоколи, про проблеми при змішуванні http https, про абсолютні посилання, що починаються на //.
* Посилання в середині сторінки. Робимо сторінку з купою контенту (тексту, картинок) та внутрішні посилання, щоб воно скролилось.

1. **Використання хостінга.**

* Розповідаємо про те, що таке хостінг і чому це важливо.
* Демонстрація (можливо, з тим, що бажаючі повторюють) варіанту використання в якості хостінгу github pages.
* Озвучити прохання всім завести собі там обліковки, і почати використовувати його для наступних домашніх завдань.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “HTML Links” (англ)](https://www.w3schools.com/html/html_links.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Pseudo-elements” (англ)](https://www.w3schools.com/css/css_pseudo_elements.asp)
* [Mozilla Developer Network (MDN) «Creating hyperlinks» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks)
* [Mozilla Developer Network (MDN) «Pseudo-classes and pseudo-elements» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Pseudo-classes_and_pseudo-elements)
* [Mozilla Developer Network (MDN) «Styling links» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_text/Styling_links)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Спробувати далі мінімізувати кількість стилів, використовуючи комбінування та хитрі селектори.
  + Зробити нову сторінку для сайту, і на неї повинно вести посилання з попередньої (основної сторінки з попередніх завдань).
  + Пропонуємо учасникам спробувати на цій новій сторінці розташувати деякий набір зображень «плиткою». Наприклад, це може бути заготовка сторінки типу «каталогу інтернет магазину», з картинками товарів, поки що без тексту та інших елементів. Для цього вони повинні самостійно почитати про блоки та позиціонування.
  + Створюємо (обов’язково!) обліковку на GitHub

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Популярний сервіс репозиторіїв коду GitHub (англ)](https://github.com/)
* [Огляд роботи з системою контролю версій Git (рос)](https://ru.hexlet.io/courses/intro_to_git)
* [Репозиторій веб-сторінок GitHub Pages (англ)](https://pages.github.com/)
* [Стаття про хостінг на Вікіпедії (англ)](https://uk.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3)
* [Огляд процесу розміщення верстки на GitHub Pages. Безкоштовний хостінг GitHub (рос)](https://www.youtube.com/watch?v=YhseegRCn0U)

# ЗАНЯТТЯ 5 Блоки в HTML. Модель «ящику» (box model). Основи позиціонування

**ТЕМИ**

**12. Модель позиціонування CSS - Box model**

* Базові параметри блоку (width, height, padding etc.)
* Спеціальні параметри блоку (overflow, background)
* Відступи (margin, padding) та границі (border)

**Практика:**

Застосовуємо Box model до документу ([приклади виконаних завдань](https://drive.google.com/open?id=1I5iwEc-vX56LO0MBlw3FTgEDRYe-WY8k))

**13. Фон**

* Поняття фону, його прив’язка до елементу
* Фонове зображення, розташування, прокрутка та повторення
* Градієнт
* Можливість задати декілька фонів одночасно

**Практика:**

Працюємо з фоном та його варіаціями ([приклади виконаних завдань](https://drive.google.com/open?id=1I5iwEc-vX56LO0MBlw3FTgEDRYe-WY8k))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково (в охочих) дивимось та обговорюємо разом домашнє завдання**

* Цікавимось в тих, хто щось зробив, як і на основі чого розбирався з блоками, як можуть пояснити.

1. **Обговорюємо та пробуємо блоки та їх властивості.**

* Розповідаємо про поняття блочних та рядкових елементів, в основному апелюючи до «div» та «span». Демонструємо поведінку на банальних прикладах, тимчасово додаючи стилем границю – до речі, вказавши, що це непоганий прийом для аналізу розташування елементів.
* Демонструємо основні властивості блоку – margins, paddings, borders, width, height.
* Демонструємо базові варіанти використання фону (атрибут стилю background).
* Робимо разом з групою рекомендований варіант домашнього завдання – з картинками, розташованими «плиткою». Тут ще непогано виходить згадати про псевдо-клас hover – наприклад, можна змінювати параметри блоку при наведенні, і картинка буде братися в рамку або збільшуватись
* Варіанти відображення елементу (атрибут display) block, inline, inline-block.

1. **Розповідаємо про поняття позиціонування (атрибут стилю position) –** в чому ідея, які основні варіанти.

* Експериментуємо разом з варіантами relative та absolute. Робимо, щоб при наведенні наші плитки зсувалися. Одно картинку позиціонуємо абсолютно над іншими.
* Створюємо заготовку для розміщення плиткою не просто картинок, а блоків «картинка із заголовком згори, та текстом та посиланням знизу».
* Поки що без float та flex.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “CSS Box Model” (англ)](https://www.w3schools.com/css/css_boxmodel.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Margins” (англ)](https://www.w3schools.com/css/css_margin.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Padding” (англ)](https://www.w3schools.com/css/css_padding.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Height and Width” (англ)](https://www.w3schools.com/css/css_dimension.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Outline” (англ)](https://www.w3schools.com/css/css_outline.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Borders” (англ)](https://www.w3schools.com/css/css_border.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Multiple Backgrounds” (англ)](https://www.w3schools.com/css/css3_backgrounds.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Colors” (англ)](https://www.w3schools.com/css/css3_colors.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Gradients” (англ)](https://www.w3schools.com/css/css3_gradients.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Shadow Effects” (англ)](https://www.w3schools.com/css/css3_shadows.asp)
* [Mozilla Developer Network (MDN) «The box model» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Introduction_to_CSS/Box_model)
* [Mozilla Developer Network (MDN) «Box model recap» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_boxes/Box_model_recap)
* [Mozilla Developer Network (MDN) «Changing background styles using CSS» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_boxes/Backgrounds)
* [Mozilla Developer Network (MDN) «Advanced box effects» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/Styling_boxes/Advanced_box_effects)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо вдосконалити сторінку «каталогу» так, щоб в кожній плитці була не лише картинка, а і текстовий заголовок. Кажемо, що наша мета – вміти розташувати плиткою картки з будь-якою інформацією.
  + Бажано показати пару цікавих прикладів з реальних сайтів – я правило, каталог плиткою є в більшості інтернет-магазинів. Можливо, мотиваційним буде приклад сайту типу «каталог зірок» - <https://www.influencerdb.net/db/?page=2>

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Блочная модель в CSS» (рос)](https://devionity.com/ru/courses/css-fundamentals/css-box-model)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Внешние и внутренние отступы» (рос)](https://devionity.com/ru/courses/css-fundamentals/margin-and-padding)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Границы» (рос)](https://devionity.com/ru/courses/css-fundamentals/css-borders)
* Навчальна платформа Devionity, курс Основы CSS, заняття «Фоновые цвета и изображения» (рос)

# ЗАНЯТТЯ 6 Продовження блоків та позиціонування. Float

**ТЕМИ**

**14. Елементи, що плавають (float)**

* CSS атрибути float та clear
* Побудова макету з кількома колонками на базі float
* Clearfix
* display: inline-block для керування розташуванням блоків

**Практика:**

Застосовуємо нові атрибути та елементи до документу ([приклади виконаних завдань](https://drive.google.com/open?id=1zGoVhn13QYwcyeXGOS39m1KVE-47uAnG))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково дивимось та обговорюємо разом домашнє завдання**

* Демонструємо свій варіант виконання домашнього завдання (розстановки плиткою блоків заголовок-інформація)
* Обговорюємо варіанти тих, хто зробив, порівнюємо.

1. **Повторюємо блоки з їх властивостями і глибше демонструємо позиціонування.**

* Намагаємось провокувати питання та демонструвати відповіді на них у вигляді експериментів.
* Ще раз експериментуємо з абсолютним позиціонуванням.
* Демонструємо варіант, як можна до наших «карток», розташованих плиткою, додати не лише заголовок згори, а і різні елементи під картинкою, використовуючи position: absolute.
* Демонструємо позиціонування fixed, можна на прикладі фіксованого верхнього меню. Показуємо класичний підхід до виходу з ситуації, коли це меню накриває контент (відступ контенту згори).

1. **Потік документу та властивість стилю float.**

* Розповідаємо про потік документу.
* Вводимо та демонструємо на штучних прикладах базові варіанти використання float.
* Переробляємо розташування елементів в середині наших плиток, використовуючи float.
* Демонструємо варіант розстановки самих плиток-карток на сторінці, використовуючи float

1. **Семантичні теги**

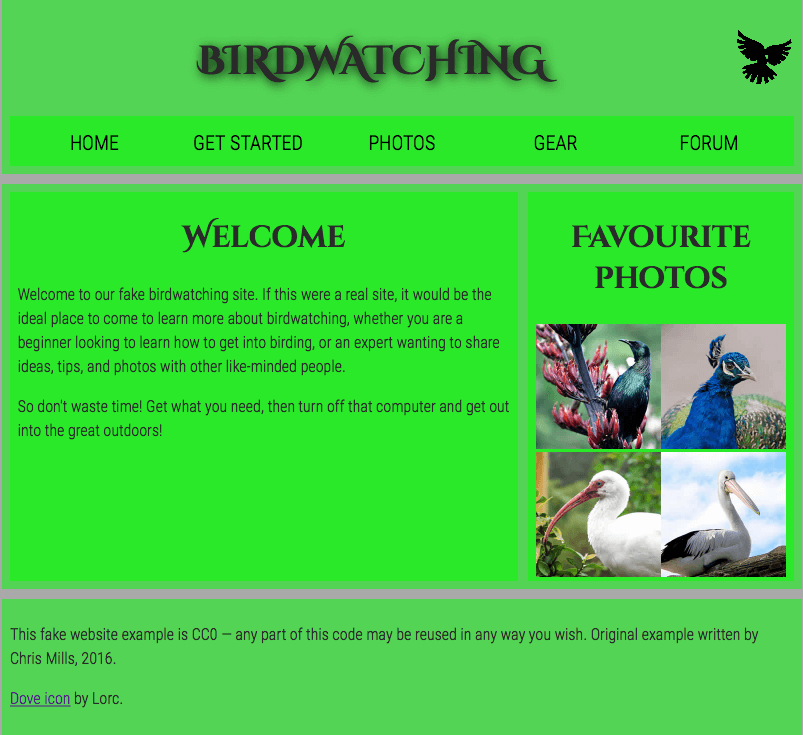
* Розповідаємо про семантичні теги – більшою мірою мотиваційно та поверхнево.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “CSS Layout - float and clear” (англ)](https://www.w3schools.com/css/css_float.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Layout - display: inline-block” (англ)](https://www.w3schools.com/css/css_inline-block.asp)
* [Mozilla Developer Network (MDN) «CSS syntax» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/CSS_layout/Floats)
* [Mozilla Developer Network (MDN) «Positioning» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/CSS_layout/Positioning)
* [Mozilla Developer Network (MDN) «Practical positioning examples» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/CSS_layout/Practical_positioning_examples)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо всім детальніше прочитати про семантичні теги, та попрактикуватися з блоками і позиціонуванням, створивши структуровану сторінку по типу наступної:  
    <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Structuring_a_page_of_content>
  + Блок з тестом (там, де «Welcome» - зробити вузьким, на користь блоку з фото. А в блоці з фото розмістити ті самі плитки, які робились в попередніх ДЗ.
  + Використати семантичні теги хоча б для заголовку, головного меню, контенту та підвалу.
  + Термін цього завдання – тиждень (тобто воно на «два заняття»).



**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Плавающие элементы» (рос)](https://devionity.com/ru/courses/css-fundamentals/css-float)

# ЗАНЯТТЯ 7 Flexbox

**ТЕМИ**

**15. Позиціонування елементів**

* Стандартне розміщення в потоці документа
* Відносний ссув (position:relative)
* Незалежне розташування в блоці (position: absolute)
* Фіксоване розташування на екрані (position: fixed)
* Позиціонування за допомогою flexbox

**Практика:**

Застосовуємо прийоми позиціонування та flexbox до документу ([приклади виконаних завдань](https://drive.google.com/open?id=1L8Rn0wQFeoOyhqp1y-q_K7qVJk6oG3St))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково дивимось та обговорюємо разом домашнє завдання**

* По завданню з минулого заняття (макет сторінки з семантичними тегами) – лише відповідаємо на питання, щоб дати можливість тим, хто ще не робив (не зробив), доробити до наступного разу.
* По завданню з поза-минулого заняття – розстановки блоків плиткою – якщо в когось з тих, хто намагався його зробити, все ще не вийшло - розбираємо ще раз та експериментуємо.

1. **Розповідаємо про варіант розстановки елементів flexbox.**

* Проблематика (навіщо створено), які можливості надає.
* Тут можна навести класичну історію про те, що варіанти розв’язків дуже природньої на перший погляд задачі - «відцентрувати div по вертикалі» - завжди шокували своєю складністю веб-розробників, які вперше з цією задачею стикалися.

1. **В спокійному темпі проводимо експерименти з тим, як робити розстановки блоків за допомогою flex, на мінімалістичних прикладах.**

* Можна йти за канвою   
  <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox>  
  <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox>
* В ідеалі просити всіх повторювати експерименти.

1. **Демонструємо, як можна використати flexbox для виконання нашої улюбленою задачі розстановки блоків плиткою.**

* Демонструємо, що flexbox допомагає розмістити елементи в середині плитки (наприклад, кнопки або іконки дій під картинкою).

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “CSS Flexbox” (англ)](https://www.w3schools.com/css/css3_flexbox.asp)
* [Mozilla Developer Network (MDN) «Flexbox» (англ)](https://developer.mozilla.org/uk/docs/Learn/CSS/CSS_layout/Flexbox)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Залишається завдання з минулого разу
  + Плюсом пропонуємо використати flex там, де буде бажання – і особливо для розстановки плиток та елементів всередині плитки.
  + Нагадуємо слухачам про те, що програмою передбачено захист фінальних робіт і що зараз саме час визначитися з тематикою майбутньої фінальної роботи.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Просвітницький веб-ресурс «CSS-TRICKS», стаття “A Complete Guide to Flexbox” (англ)](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)

# ЗАНЯТТЯ 8 Адаптивний дизайн. Знайомство з bootstrap

**ТЕМИ**

**16. Медіа запити**

* Медіа запити як інструмент значно підвищити гнучкість макета
* media types
* media features

**Практика:**

Застосовуємо медіа інструменти до документу ([приклади виконаних завдань](https://drive.google.com/open?id=1wDXiwIHKjCP1Rssx-kEhY7soonXNx4Fu))

**17. Адаптивний макет**

* Проблема багатьох пристроїв. Поняття адаптивного макету.
* Область перегляду (viewport).
* Медіа запити для побудови адаптивного макету. Точки зміни (breakpoint)

**Практика:**

Наділяємо документ властивостями адаптивного макету ([приклади виконаних завдань](https://drive.google.com/open?id=1NxMUwroO7qYu_aF5qk-ZgAPhRcPRlEkc))

**18. Twitter bootstrap - знайомство**

* Каркаси CSS (CSS Frameworks) - мета створення і проблематика
* Bootstrap як один з основних каркасів
* Підключення. Залежність від JQuery
* Параметри області відображення - мета параметр viewport
* Основний контейнер - класси container та container-fluid

**Практика:**

Змінюємо підхід до стилізації документу та підключаємо bootstrap ([приклади виконаних завдань](https://drive.google.com/open?id=1bB_z-ZKH0fVmunhEEAsjsRYMBgm5ikOB))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Вибірково дивимось та обговорюємо разом домашнє завдання**

* Ментору бажано мати свою реалізацію сторінки з семантичними тегами та розстановкою блоків (“зелена” сторінка з сайту мозіла, завдання з позаминулого заняття), з використанням flex та з розстановкою каталогу плиток всередині.
* Вибірково дивимось та обговорюємо варіанти в тих, в кого вийшло або майже вийшло. Скоріше за все, доводити до ідеалу займе занадто багато часу - тому лише окреслюємо шлях.
* Демонструємо та аргументуємо свій варіант.

1. **Мотивація щодо створення та використання css frameworks (з прицілом на bootstrap).**

* Як варіант мотивуючих факторів: швидкість розробки, базування на найкращих практиках, звичний для користувача інтерфейс.
* Окремо наголошуємо на потребі створення сайтів, які гарно виглядають на мобільних пристроях. Демонструємо, що наш “каталог плиткою” або інші ростановки з попередніх прикладів або домашніх завдань не дуже гарно поводять себе при перегляді з телефону (можна використати симулятор в консолі розробника chrome).
* Згадуємо про те, що насправді аудиторія, що заходить на сайт з пристроїв, часто важливіша та більша, ніж ті, хто заходить з десктоп. Розповісти про поняття mobile-first design.

1. **Ліричний відступ - про те, як робити адаптивні макети вручну з використанням сss media queries.**

* Демонстрація синтаксису
* Поняття “точки зміни” - breakpoint.
* Поняття viewport (області перегляду). Непагано пояснено ось тут:

<https://developer.mozilla.org/ru/docs/Mozilla/Mobile/Viewport_meta_tag>

* Демонстрація того, як зробити, щоб “каталог плиткою” при зменшенні екрану спочатку зменшував картинки, а при досягненні зовсім маленької ширини - розставляв все в одну колонку.

1. **Знайомство з бутстрап.**

* Робимо огляд платформи bootstrap на базі їх сайту (є сайти-переклади, можна демонструвати їх). Кажемо, що глобально - це величезна збірка стилів, які задаються селекторами за тегами та за класами, які ми можемо використовувати.
* Пока, що їх приклади гарно виглядають на мобільних, та ще й змінюють вигляд в залежності від розміру екрану. Пояснити поняття adaptive (responsive) web design (якщо ментор глибоко володіє цим аспектом теми, можна пояснити різницю - але головне, щоб всі усвідомили важливість і складність нормального вигляду сайту на різних пристроях).
* Демонстрація підключення бутстрап до сторінки.
  + - * Варто сказати, що для функціонування деяких інтерфейсних елементів та ефектів використовується javascript та бібліотека jquery, але деталізацію відкласти.
      * Варто сказати про SAAS та LESS (так як бустрап пробонує скачати себе у вигляді saas) - про те, навіщо це, і що це - для загального розвитку.
  + Експерементування по створенню сторінки, на якій використувується кілька елементів бутстрап, разом з групою.
    - * Як варіант - створюємо сторінку з мінімалістичним верхнім (горизонтальним) меню, основною частиною, в якій є заголовок, текст (jumbotron?), та кнопка. Можливо, також підвал (footer).
      * Можна повторити один з стартових шаблонів з сайту бутстрап – наприклад:

<https://getbootstrap.com/docs/3.3/examples/jumbotron-narrow/>

* + - * По ходу розповідаємо про класси container та container-fluid
      * Показуємо, що для “тонкої” стилізації елементів бібліотека надає набір класів зі змістовними іменами, про які можна прочитати в документації до відповідного елементу. Як варіант прикладу для їх демонстрації - зробити кнопку маленькою та великою, червоною - зеленою - жовтою (класс “btn” та його варіаціїї).
  + Поверхнева демонстрація bootstrap grid - розставимо щось у вигляді пари колонок та рядочків.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “CSS Media Queries” (англ)](https://www.w3schools.com/css/css3_mediaqueries.asp)
* [Курс з веб-розробки на W3Schools, тема “CSS Media Queries - Examples” (англ)](https://www.w3schools.com/css/css3_mediaqueries_ex.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Introduction” (англ)](https://www.w3schools.com/css/css_rwd_intro.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Media Queries” (англ)](https://www.w3schools.com/css/css_rwd_mediaqueries.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Images” (англ)](https://www.w3schools.com/css/css_rwd_images.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Videos” (англ)](https://www.w3schools.com/css/css_rwd_videos.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Frameworks” (англ)](https://www.w3schools.com/css/css_rwd_frameworks.asp)
* [Курс з веб-розробки на W3Schools, тема “Responsive Web Design - Templates” (англ)](https://www.w3schools.com/css/css_rwd_templates.asp)
* [Курс з веб-розробки на W3Schools, тема “Bootstrap 3 Tutorial” (англ)](https://www.w3schools.com/bootstrap/default.asp)
* [Курс з веб-розробки на W3Schools, тема “Bootstrap Get Started” (англ)](https://www.w3schools.com/bootstrap/bootstrap_get_started.asp)
* [Mozilla Developer Network (MDN) «Using media queries» (англ)](https://developer.mozilla.org/uk/docs/Web/CSS/Media_Queries/Using_media_queries)
* [Mozilla Developer Network (MDN) «The building blocks of responsive design» (англ)](https://developer.mozilla.org/en-US/Apps/Progressive/Responsive/responsive_design_building_blocks)
* [Mozilla Developer Network (MDN) «Responsive images» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images)
* [Офіційний веб-сайт відкритого фреймворку Bootstrap, розділ “Getting started” (англ)](https://getbootstrap.com/docs/3.3/getting-started/)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Ознайомлення” (укр)](https://twbs.docs.org.ua/getting-started/)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Огляд” (укр)](https://twbs.docs.org.ua/css/)
* [Навчальна платформа FructCode, курс Bootstrap, Урок 1 (рос)](https://fructcode.com/ru/courses/bootstrap/getting-started/)
* [Навчальна платформа FructCode, курс Bootstrap, Урок 2 (рос)](https://fructcode.com/ru/courses/bootstrap/video-container-and-container-fluid/)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Переробити попередні завдання з максимальним використанням елементів bootstrap.
  + Повторити шаблон “зеленої сторінки”. Задача - зробити те саме по суті на елементах бутстрап. Підкреслити, що НЕ потрібно намагатись витримати розміри і взагалі точне співпадіння - головне лишити схему розташування.

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Structuring_a_page_of_content>

* + Повторити каталог плиткою.

Як варіант, запропонувати використати bootstrap thumbnails:

<https://getbootstrap.com/docs/3.3/components/#thumbnails-custom-content>

* + Вірогідно, це завдання краще планувати на два заняття
  + Нагадуємо про вибір тематики фінального проекту

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Стаття про Bootstrap на Вікіпедії (укр)](https://uk.wikipedia.org/wiki/Bootstrap)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Компоненти” (укр)](https://twbs.docs.org.ua/components/)
* [Стаття про jQuery на Вікіпедії (укр)](https://uk.wikipedia.org/wiki/JQuery)
* [Офіційний веб-сайт фреймворку jQuery, розділ “Download” (англ)](http://jquery.com/download/)
* **Також можна починати опрацьовувати матеріали корисних безкоштовних курсів:**
  + - * [Серія уроків по Bootstrap від програміста Назара Токаря (укр/рос)](https://tokar.ua/read/6901)

# ЗАНЯТТЯ 9 Продовження bootstrap

**ТЕМИ**

**19. Bootstrap - сітка позиціонування (grid system)**

* Підхід до позиціонування на основі рядків та колонок.
* Сітка bootstrap (bootstrap grid)
* Примусове позиціонування на сітці - push, pull. Offset
* Адаптивність - xs, xl, ….

**Практика:**

Застосовуємо сітку позиціонування до документу ([приклади виконаних завдань](https://drive.google.com/open?id=16NddO9KYlXiu6kmB7QbQaNr0Fn7lBcwv))

**20. Основи побудови головного навігаційного меню з bootstrap**

* Поняття навігаційного меню та вимоги до нього
* navbar в bootstrap

**Практика:**

Створюємо навігаційне меню ([приклади виконаних завдань](https://drive.google.com/open?id=1Oz8cqTNQcf9cOrqQN_KWEuImalZ2cDhZ))

**21. Наповнення навігаційного меню. Адаптивність**

* Додавання логотипу в navbar
* Додавання пунктів меню
* Створення класичного “бутерброду” для мобільної версії

**Практика:**

Додаємо пункти меню та доповнюємо його логотипами

[(приклади виконаних завдань №1)](https://drive.google.com/open?id=14dRP2I5j-2np108KZXNICCWh7rFBTyBn)

[(приклади виконаних завдань №2)](https://drive.google.com/open?id=1wbbDUrR_JZxNpnFu5wVIP_eL8Ggtl2Ra)

[(приклади виконаних завдань №3)](https://drive.google.com/open?id=1ek-O3MRC3R9sE-jIreSyKZHgmk0NRPcf)

**22. Робота з фоном**

* Додавання та позиціонування фонового зображення
* Додавання відтіняючого overlay

**Практика:**

Доповнюємо робочий макет фоном з відповідними ефектами ([приклади виконаних завдань](https://drive.google.com/open?id=1M31qzXNWribOfakGrAYU4ZKtt3YOeIhG))

**23. Кілька корисних речей для початку розробки сайту**

* Дротовий макет як важливий інструмент вибору вірного шляху
* Інструменти для створення прототипу.
* Використання стандартних елементів дизайну. Шрифти гугл

**24. Створення основного блоку**

* Додавання другого container
* Розташування блоків з використанням grid
* Стилізація

**Практика:**

Видозмінюємо сторінку та створюємо основний блок ([приклади виконаних завдань](https://drive.google.com/open?id=10Tp5b4Vdb0nZzvfFROvEepiITY7_yOmI))

**25. Створення підвалу (footer) і остаточне оформлення сторінки**

* Додавання container для підвалу
* Розташування блоків з використанням grid
* Стилізація

**Практика:**

Видозмінюємо сторінку та створюємо підвал ([приклади виконаних завдань](https://drive.google.com/open?id=19TSUhAxUcFRylQXiofMxYEBsvUW5kL5Y))

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Обговорюємо разом домашнє завдання**

* Повністю свій варіант не надаємо, але тим, хто щось зробив і має питання, окреслюємо шлях їх вирішення.
* Якщо буде доцільно, окреслюємо стандартні проблеми при виконанні завдання.
* Даємо загальні рекомендації, як доробити (зробити) це завдання.

1. **Глибше демонструємо bootstrap grid.**

* Разом з групою ретельно експерементуємо з грід розстановкою бутстрап.
* Ще раз на її прикладі показуємо, що таке адаптивний макет і як керувати тим, коли колонки стають одна під одною.
* Можна згадати, що в сучасному стандарті css додали режим відображення display: grid, яких дозволяє зручно робити потрібні розстановки - але ще не всі браузери його підтримують.
* Можна дати як корисне посилання <https://www.w3schools.com/css/css_templates.asp> - кілька варіантів базових шаблонів, без бутстрап, але з float та flex.

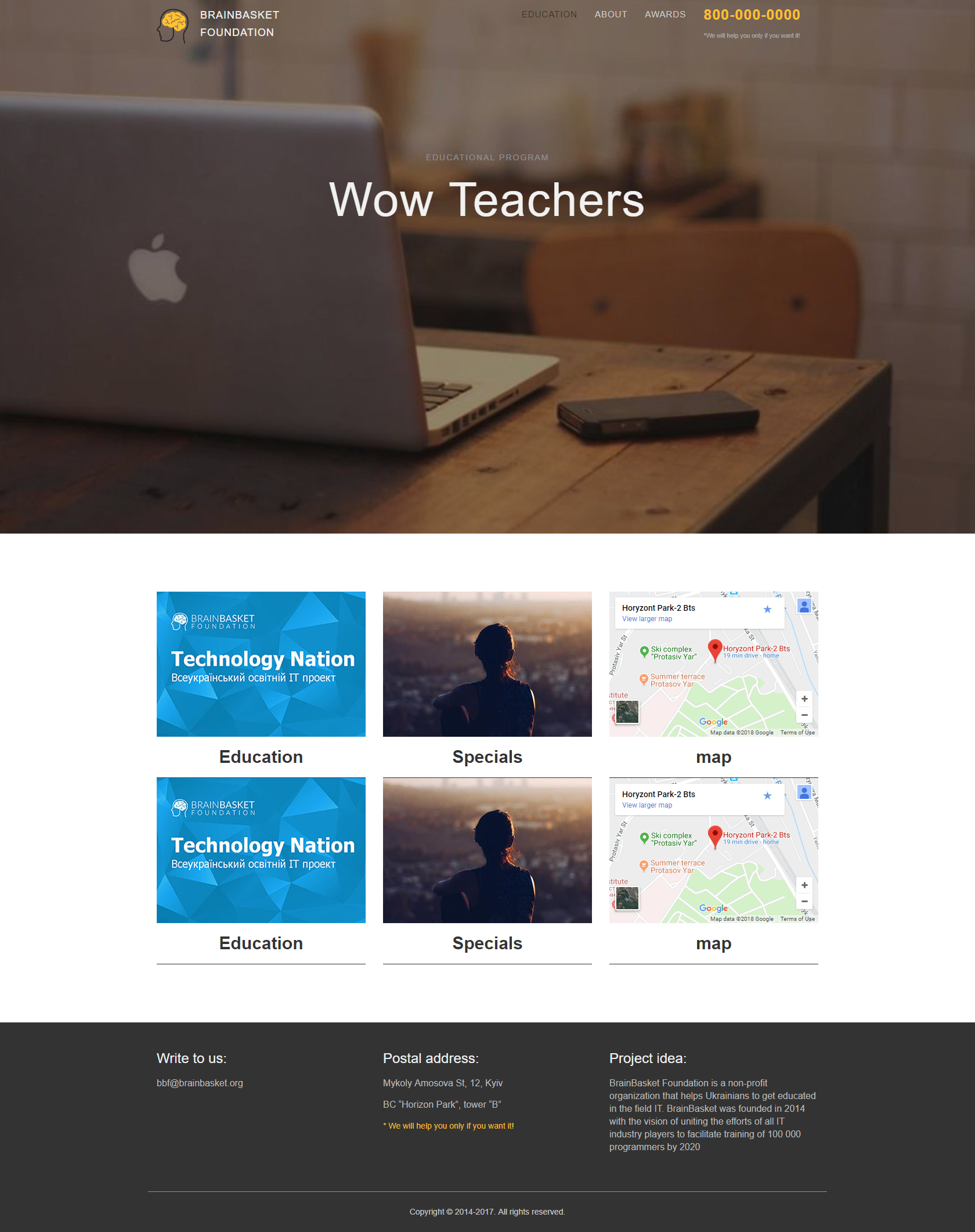
1. **Класична навігаційна панель bootstrap**

* Разом з групою на основі документації bootstrap створюємо верхню навігаційну панель з меню, яке перетворюється на “бутерброд” при зменшенні ширини екрану (в тому числі на мобільних).
* Додаємо в панель логотип-картинку та кілька (2+) пунктів меню

1. **Бустрап як основа створення “сучасно виглядаючого” односторінкового сайту-візитки.**

* Додаємо фонове зображення, яке займає весь перший “екран” (див. скрін нижче).
* Додаємо основний блок на “наступному екрані”, додавши ще один container. Непогано там розташувати кілька плиток, що допоможе потім усім в домашньому завданні.
* Додаємо підвал (нижній інформаційний блок, footer).
* Все це розташовується з використанням bootstrap grid.

1. **В результаті повинно вийти повинно щось на кшталт наступного скріна,** причому відкривши сторінку, користувач повинен бачити лише “перший екран” з великим фоновим зображенням - а за всім іншим він повинен скролити донизу або натискати на пункти меню.



**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “Bootstrap Grids” (англ)](https://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp)
* [Курс з веб-розробки на W3Schools, тема “Bootstrap Grid System” (англ)](https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp)
* [Курс з веб-розробки на W3Schools, тема “Bootstrap Navigation Bar” (англ)](https://www.w3schools.com/bootstrap/bootstrap_navbar.asp)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Система сітки” (укр)](https://twbs.docs.org.ua/css/#grid)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Приклади Bootstrap сітки” (укр)](https://twbs.docs.org.ua/examples/grid/)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Підготовлена навігаційна панель” (укр)](https://twbs.docs.org.ua/components/" \l "navbar-default)
* [Україномовна адаптація матеріалів офіційного веб-сайту відкритого фреймворку Bootstrap, розділ “Зображення бренду” (укр)](https://twbs.docs.org.ua/components/" \l "navbar-brand-image)
* [Навчальна платформа FructCode, курс Bootstrap, Урок 5 (рос)](https://fructcode.com/ru/courses/bootstrap/grid-system-class-row/)
* [Навчальна платформа FructCode, курс Bootstrap, Урок 8 (рос)](https://fructcode.com/ru/courses/bootstrap/grid-system-class-cols/)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо доробити - зробити - покращити завдання з попереднього заняття, використовуючи отримані знання.
  + Просимо всіх остаточно визначитися з фінальним проектом. Як мінімальний варіант, можна запропонувати створити інформаційний сайт на цікаву для учасника тематику, з сучасним оформленням та з використанням бутстрап.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Бібліотека шрифтів Google (англ)](https://fonts.google.com/)
* **Також можна починати опрацьовувати матеріали корисних безкоштовних курсів:**
  + - * [edX, курс “ Microsoft: DEV203x Introduction to Bootstrap” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV203x+4T2017/course/)

# ЗАНЯТТЯ 10 Знайомство з Javascript

**ТЕМИ**

**26. Javascript як основний інструмент WEB 2.0+**

* Історія та мета створення JS
* JS як мова програмування - оглядово
* Консоль JS в браузерах
* Приклад обробки натиснення на кнопку (без глибокого пояснення)
* Базове поняття про DOM, найпростіший приклад отримання елементу за ІД та зміни його тексту
* Вплив місця розміщення тегу <script> в документі
* setTimeout та setInterval (без глибокого пояснення)
* Мотиваційна демонстрація можливостей маніпуляції елементами DOM - наприклад, зміна кольору та розміру тексту та зміна розміру зображення за таймером.

**Практика:**

* Спробувати зробити простий код (на вибір ментора)
* Спробувати зробити код, що реагуватиме на елемент сторінки слухача (на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Обговорюємо разом домашнє завдання**

* На цей момент бажано, щоб в ментора був свій варіант.
* Дивимось варіанти учасників. Тим, хто робив, але не зробив все, або зробив, але не зміг вирішити частину проблем - намагаємось допомогти, по можливості залучаючи до цього групу. Там, де це можна зробити швидко - робимо виправлення. В інших варіантах - пропонуємо ідеї та напрямки руху.
* Демонструємо та описуємо власний варіант.

1. **Знайомство з javascript.**

* Робимо мотиваційний вступ - розповідаємо про мету створення та поверхневу історія розвитку JS. Як варіант, можна згадати, що “колись сайти були занадто статичними”, про зміщення балансу між логикою на сервері і на клієнті (туди-назад час від часу), про зростання “апетитів” користувачів.
* Демонструємо вбудовування найпростішого JS в сторінку, просимо всіх повторити. Робимо вбудованим тегом script. Мабуть, демонструємо виклик alert-у. Після цього пояснюємо недоліки alert (наприклад, “некрасиво”, “якщо забудемо - побачить користувач”, “лише строкова інфа”...), і замінюємо його на console.log - без заглиблення в поняття об’єкт, window і так далі - просто сказавши, що за цим є багато всього, але поки що ми просто віримо, що воно працює, і знаємо як використати.
* Демонструємо базовий варіант работи з DOM на javasccript. Просимо групу робити разом з нами.
  + - * Демонструємо отримання елементу за ІД, зміну якого-небудь простого атрибуту (стилю). Наголошуємо на те, що для того, щоб маніпулювати документом в скрипті, сам скрипт потрібно розмістити в кінці документу.
      * На даному етапі намагаємось уникати пояснень глибоких речей, описуємо на рівні “ось така функція дозволяє отримати доступ до елементу з HTML”.
      * Як варіант, можно взяти приклад з заняття, коли робили нарис каталогу плиткою - а саме сторінку з картинками плиткою (без тексту та додаткових елементів) - дати якій з них Id, отримати через getElementById, та змінити, наприклад, style.width.
      * Важливо - ментор повинен мати і перед демонстрацією роздати всім код сторінки-заготовки, з якої починати - не факт, що в усіх вона збережеться, навіть якщо була на одному з попередніх занять.
      * Впевнитись, що в усіх бажаючих з групи вийшло повторити. Допомогти.
      * Варто ще раз підкреслити, що таким чином можна змінювати будь-які доступні стилі будь-якого елементу.
* Демонструємо, як можна додавати елементи до DOM, використовуючи document.createElement, setAttribute та appendChild
  + - * Створюємо цикл (синтаксис циклу пояснюємо мінімалістично), який генерує цей “каталог плиткою” - тобто додає до контейнеру картинки, наприклад, з іменами “img1.jpg”, “img2.jpg” і т.д. - по суті, в результаті буде така сама сторінка, але картинки-плитки будуть генеруватися динамічно.
      * Мінімалістично пояснюємо, що таке документ та елемент.
      * Намагаємось, щоб більшість з групи також реалізувало цю функціональність.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “JavaScript Tutorial” (англ)](https://www.w3schools.com/js/default.asp)
* [Курс з веб-розробки на W3Schools, тема “JavaScript Introduction” (англ)](https://www.w3schools.com/js/js_intro.asp)
* [Mozilla Developer Network (MDN) «Основи JavaScript» (укр./англ)](https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics)
* [Mozilla Developer Network (MDN) «JavaScript» (укр./англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript)
* [Mozilla Developer Network (MDN) «What is JavaScript?» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо учасникам взяти одну з їх сторінок з попередніх завдань, та спробувати:
    - * Прописавши в ній атрибут Id трьом HTML тегам на вибір, додати до сторінки скрипт, котрий змінить стилі цим елементам.
      * Як варіант, запропонувати взяти сторінку з “каталогом плиткою”, але більш пізню версію, де в плитці вже є не лише картинка, а й текст, і в одній плитці зробити зображення більшим, в другій - зробити текст червоним, а в третій - додати товсту зелену рамку навколо картинки.
  + Зробити сторінку, на якій буде розміщено список, елементи якого повинні мати вигляд “Елемент-1, Елемент-2, …, Елемент-100.” (звичайні тексти). При цьому сама сторінка та тег списку (ul або ol) створюється статично, а от самі елементи в список (li) повинні додаватись скриптом.
  + Нагадуємо про необхідність вибору ідеї для фінального проекту. При потребі, обговорюємо ідеї з тими, хто вже їх має.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Курс Javascript и jQuery, заняття «Использование DOM. Доступ к элементам» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-using-dom-accessing-elements)
* [Проект "JavaScript довідка українською" (укр)](http://xn--80adth0aefm3i.xn--j1amh/)
* [Онлайн книга «Выразительный Javascript» (рос)](https://karmazzin.gitbooks.io/eloquentjavascript_ru/?q=)
* **Також можна починати опрацьовувати матеріали корисних безкоштовних курсів:**
  + - * [edX, курс “PennX: SD4x Programming for the Web with JavaScript” (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV211.1x+4T2017/course/)
      * [edX, курс “Microsoft: DEV211.1x Introduction to HTML and JavaScript” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV211.1x+4T2017/course/)

# ЗАНЯТТЯ 11 Огляд синтаксису javascript та роботи з DOM

**ТЕМИ**

**27. Javascript як мова програмування**

* Виділення скрипта в окремий файл. Структура файлу js - функціональний стиль.
* Основні синтаксичні елементи JS - змінні, умовні конструкції, цикли.
* Типи змінних. Слабка типізація.
* Функції - базовий синтаксис, передача параметрів

**Практика:**

* Розглядаємо приклади роботи зі змінними та маніпулювання з DOM (на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Обговорюємо разом домашнє завдання**

* Дивимось варіанти учасників. Якщо вдається, то дороблюємо (демонструючи групі та закликаючи групу допомогти) два-три варіанти тих, в кого щось не вийшло (але напрацювання є).

1. **Підключення зовнішніх скриптів**

* Пояснюємо, чому не варто писати більшість коду в середині тегу <script> в основному HTML.
* Демонструємо, як підключати скрипт з окремого файлу, і разом з групою переробляємо приклад з попереднього заняття (виносимо код в окремий файл, і його підключаємо).

1. **Базовий синтаксис javascript.**

* Розраховуємо на те, що учасники мають базові знання з програмування хоча б якоюсь мовою - тобто поняття “змінна”, “if”, “цикл” та “функція” вони розуміють.
* Пропонуємо групі повторювати експерименти ментора (ймовірно, варто робити максимально прості демонстрації - без маніпуляції документом, на рівні арифметики та виводу значень в консоль браузера).
* Демонструємо базові конструкції мови, без ООП та масивів. Змінні, деякі оператори (варто зупинитись на операторах “присвоєння” та “порівнняння” (=, ==, згадати про ===)) , if, for.
* Після демонстрації змінних робимо теоретичний відступ щодо того, що js - це мова із слабкою типізацією.
* Також можна згадати, що синтаксис подібний до мови “С”.
* Демонструємо роботу з функціями на JS (в базовому вигляді, без анонімних функцій). Показуємо створення, виклик, передачу параметрів та повернення результату.

1. **Утилітарно вводимо поняття “об’єктів, які надає браузер” (window, document).** Якщо раптом група знайома з ООП - то все просто. Якщо, що більш імовірно, ні - то кажемо, що для зручності JS і багато інших мов дозволяють згрупувати змінні та функції разом, і таке групування іноді називають об’єктом (на розсуд ментора, чи варто намагатись пояснювати різницю між класом та об’єктом). Далі, як варіант, можна розповісти та продемонструвати приблизно наступним чином:

* Якщо в деякій змінній зберігається об’єкт, то ми можемо через крапку отримувати чи змінювати значення його змінних (які часто називають полями (fields) або властивостями (properties)), а також викликати його функції (які часто називають методами (methods)).
* При цьому поля об’єкта часто, в свою чергу, містять деякі об’єкти. І методи об’єкта також часто повертають об’єкти.
* Сам браузер забезпечує, що в будь-якому місці коду ми можемо використовувати деякі вбудовані об’єкти. Насправді, доступний об’єкт window, а більшість іншого - це змінні в ньому, які є також об’єктами - але історично склалося, що ми можемо звертатися до них не лише як window.щось, а і просто як до “щось”. Наприклад, наш знайомий виклик console.log(“Hello, teachers!”) - насправді є викликом window.console.log(...) - тобто змінна з іменем window, яка надається нам браузером, містить об’єкт, в якому є змінна console, яка містить об’єкт, у якого є метод (функція) log(), яка виводить свої аргументи (те, що ми їй передамо) в консоль браузера.
* зокрема в window (а отже і глобально) доступна змінна document, яка містить об’єкт, через який ми можемо викликати купу корисних функцій - знову продемонструвати отримання елементу по ІД.
* також варто продемонструвати зміну або отримання location.

1. **Трохи глибше, ніж минулого разу, торкаємось поняття DOM.**

* Розповідаємо, що браузер “парсить” html - тобто аналізує його, і створює всередині себе структуру, схожу на дерево, яку називають ДОМ. І надає нам, як розробникам, можливість цим “домом” маніпулювати, знаходячи в ньому елементи, змінюючи їх властивості, а також додаючи нові та видаляючи існуючі елементи.
* Через цю призму ще раз дивимось на наш приклад, в якому ми заповнювали картинки в сторінці, або список з домашнього завдання.
* Демонструємо, що можна будувати в такий спосіб і більш складні конструкції (блоки HTML) - наприклад, генеруємо список, кожен елемент якого - це <div> з текстом та картинкою в середині - приблизно такий:
  + - * <div><h1>Image-1</h1><img src=”img-1.jpg” /></div>
      * <div><h1>Image-2</h1><img src=”img-2.jpg” /></div>
      * …………

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “JavaScript Where To” (англ)](https://www.w3schools.com/js/js_whereto.asp)
* [Курс з веб-розробки на W3Schools, тема “JavaScript Syntax” (англ)](https://www.w3schools.com/js/js_syntax.asp)
* Курс з веб-розробки на W3Schools, тема “JavaScript Functions” (англ)
* [Mozilla Developer Network (MDN) «A first splash into JavaScript» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/A_first_splash)
* [Mozilla Developer Network (MDN) «JavaScript building blocks» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Building_blocks)
* [Mozilla Developer Network (MDN) «Making decisions in your code — conditionals» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Building_blocks/conditionals)
* [Mozilla Developer Network (MDN) «Looping code» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Building_blocks/Looping_code)
* [Mozilla Developer Network (MDN) «Functions — reusable blocks of code» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Building_blocks/Functions)
* [Mozilla Developer Network (MDN) «Storing the information you need — Variables» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/Variables)
* [Mozilla Developer Network (MDN) «Basic math in JavaScript — numbers and operators» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/Math)
* [Mozilla Developer Network (MDN) «Handling text — strings in JavaScript» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/Strings)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Реалізувати генератор “каталогу плиткою” на javascript. Маючи сторінку, яка містить весь html окрім самого набору елементів каталогу (можна взяти сторінку з відповідного завдання, та вирізати ці елементи з html), підключити до неї скрипт-файл, в якому написати код, який додасть в потрібне місце 100 елементів-плиток. Кожна плитка повинна як мінімум містити заголовок (текст вигляду “Елемент каталогу номер N”, де N змінюється для кожного елементу) і картинку. Окрім того, кожен другий елемент повинен мати заголовок зеленого коліру та картинку в рамочці також зеленого коліру.
  + Для того, щоб не створювати сто реальних картинок, можна використати сервіс “picsum” - можна генерувати посилання на картинки вигляду [https://picsum.photos/290/290/?N](https://picsum.photos/290/290/?2) (тут 290 - це висота і ширина картинки, їх можна задати такі, які потрібно, а замість “N” в кінці може бути будь-яке ціле число, і тоді картинки будуть різними). Або ж можна кинути в папку кілька картинок з іменами img1.jpg, img2.jpg і т.д, і змусити скрипт генерувати їх по колу - тобто, картинки будуть повторюватись.
  + Обговорюємо фінальні проекти та формуємо список тем робіт слухачів до захисту.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Основы синтаксиса. Переменные и константы» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-syntax-basics-variables-and-constants)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Типы данных. Undefined и Null» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-data-types-undefined-and-null)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Управляющие конструкции: условия» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-conditional-statements)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Циклы. Цикл for()» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-loops-and-iterations-for-loop)
* Навчальна платформа Devionity, курс Основы CSS, заняття «Основы функций» (рос)

# ЗАНЯТТЯ 12 Масиви та об’єкти в javascript. JSON

**ТЕМИ**

**28. Структурування даних в Javascript, взаємодія з DOM**

* Масиви
* Об’єкти
* Побудова блоків DOM за масивом об’єктів JS.

**29. JSON як механізм для обміну даними в веб**

* Історія, причини та мета створення формату JSON
* JSON як більш складний варіант масиву об’єктів
* Побудова front-end логіки на основі даних.
* Теоретичний огляд архітектури взаємодії web-сторінки з api. Теоретична можливість отримання даних зовні у вигляді JSON.

**Практика:**

* Опрацьовуємо приклади масивів та об’єктів, практикуємося з JSON (на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

* + - 1. **Обговорюємо разом домашнє завдання**
* Дивимось варіанти учасників. Якщо вдається, то дороблюємо (демонструючи групі та закликаючи групу допомогти) два-три варіанти тих, в кого результат близький до потрібного, але щось не вийшло.
  + - 1. **Масиви**
* Розповідаємо та демонструємо роботу з масивами простих типів (чисел, рядків) - на штучних найпростіших прикладах.
* Обговорюємо синтаксис та використання двовимірного масиву. Як варіант мотивації та прикладу - масив базової інформації про товар, який для кожного товару містить назву, посилання на картинку та ціну :

[  
 [“Iphone X 32”, ”/images/iphonex32.jpg”,27513],  
 [“Samsung S9”, ”/images/sams9.jpg”,17500]  
]

* Вказуємо, що не дуже то зручно весь час пам’ятати, що елемент з індексом “0” - це назва, а елемент з індексом “2” - це ціна.
  + - 1. **Об’єкти**
* Вводимо поняття об’єкту - як структури даних, яка містить іменовані значення.
  + - * Можна як мотивацію взяти той самий приклад - об’єкт “товар”, вигляду   
        {  
         name:“Iphone X 32”,  
         image\_url: ”/images/iphonex32.jpg”,  
         price: 27513  
        }
      * Глибше обговорюємо те, що ці “змінні в середині об’єкту” називаються властивостями (properties) або ж полями (fields).
      * І що вони можуть містити не лише прописані в коді значння - а що завгодно. І що вони є змінними, тобто ми можемо їх змінювати в коді.
      * Демонструємо доступ до полів об’єкту через крапку (product.price) та через імена-індекси, як до масивів (product[‘price’]). Варто обговорити, що іноді об’єкти навіть називають асоціативними масивами - тобто можно об’єкт розглядати як масив, індексами якого є рядки (імена полів (властивостей)).
      * Окремо демонструємо, що поля об’єкту можуть в свою чергу бути об’єктами або масивами. Як варіант прикладу:

{  
 name:“Iphone X 32”,  
 image: {  
 url: ”/images/iphonex32.jpg”,  
 title: “Iphone X 32Gb main image”  
 },  
 price: 27513,  
 colors: [‘white’, ‘gold’, ‘black’]  
}

* Кажемо, що в об’єкті так само можуть міститися і функції, які також можна буде викликати та які спеціальним чином матимуть доступ до полів об’єкту.
* Демонструємо створення функції - методу об’єкту
* Демноструємо та обговорюємо ключове слово this.
* Зауважуємо та демноструємо, що нормальна і широко вживана конструкція - це масив об’єктів. Можна продемонструвати все той самий масив товарів.
* Рекомендовані посилання:
  + - * <https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects/Basics> - є переклад.
      * Для просунутих та самостійного опанування:  
        <https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects/Object-oriented_JS>
      1. **JSON**
* Робимо мотиваційний вступ - щодо важливої задачі обміну даними, в тому числі між клієнтом та сервером. Обговорюємо властивості, які хотілося б мати від формату даних для такого обміну (розмір, швидкість пакування-розпакування, читабельність для людини тощо).
* Коротко розповідаємо, що історично спочатку білше використовували XML (і зараз також використовують). Вказуємо недоліки.
* Розповідаємо про JSON, його переваги. Можна і про проблеми. Вказуємо, що на даний момент це найпоширеніший формат передачі даних у веб та мобільних застосуваннях.
* Демонструємо приклад JSON - логічно той самий масив товарів.
* Демонструємо, що багато постачальників даних дозволяють отримувати свої дані у форматі JSON. Як приклад можна взяти отримання курсів валют НБУ  
  <https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json>
* <https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects/JSON>  
  (є переклад російською)
  + - 1. **Демонструємо (якщо вистачає часу, то виконуємо разом з групою) генерацію DOM на базі масиву об’єктів.**
* Ментору бажано мати та роздати усім заготовку. Як варіант - сторінка, в якій десь є тег <ol>, та скрипт, якій містить масив об’єктів, кожен з яких містить заголовок та текст.  
  (можна взяти як тут, і прибрати зайві поля <https://jsonplaceholder.typicode.com/posts> )
* Пишемо скріпт, якій перебирає цей масив, та генерує по кожному елементу масиву елемент списку приблизно як на минулому занятті
  + - * <div><h1>Заголовок</h1><p>Тут йде текст…</p></div>
      * <div><h1>Інший Заголовок</h1><p>Тут йде інший текст…</p></div>
      * ………..

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “JavaScript Arrays” (англ)](https://www.w3schools.com/js/js_arrays.asp)
* [Курс з веб-розробки на W3Schools, тема “JavaScript Objects” (англ)](https://www.w3schools.com/js/js_objects.asp)
* [Курс з веб-розробки на W3Schools, тема “JSON - Introduction” (англ)](https://www.w3schools.com/js/js_ajax_intro.asp)
* [Курс з веб-розробки на W3Schools, тема “AJAX Introduction” (англ)](https://www.w3schools.com/js/js_ajax_intro.asp)
* [Mozilla Developer Network (MDN) «Arrays» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/First_steps/Arrays)
* [Mozilla Developer Network (MDN) «Вступ до об'єктів JavaScript» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects)
* [Mozilla Developer Network (MDN) «Основи об'єктів в JavaScript» (укр/англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects/Basics)
* [Mozilla Developer Network (MDN) «Working with JSON» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Objects/JSON)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Даємо групі javascript файл з змінною - масивом об’єктів, що містять інформацію про товари (як варіант - назву, посилання на картинку, ціну та кількість на складі).
  + Задача буде за цією інформацією побудувати каталог плиткою аналогічний попередньому завданню.
  + Є надія, що три підряд глобально схожі завдання дадуть можливість всім, хто докладе достатньо зусиль, застрибнути в потяг JS та маніпуляції документом.
  + Для тих, для кого це завдання буде очевидно простим, пропонуємо додатково почитати про поля вводу та виклик функції при натисканні кнопки, і виконати наступний варіант завдання:
  + На сторінці розміщуємо текстове поле вводу тексту для пошуку (searchMask) та кнопку “Шукати” (Search).
  + По натисканні кнопки, викликаємо функцію, яка генерує каталог плиткою за наданим масивом об’єктів, як і в основному варіанті завдання - але при цьому обирає з масиву лише ті об’єкти, заголовок або текст яких містить той текст, що користувач ввів до поля вводу.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Mozilla Developer Network (MDN) «Image gallery» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Building_blocks/Image_gallery)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Типы данных. Массив» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-data-types-array)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Типы данных. Объект» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-data%20types-object)

# ЗАНЯТТЯ 13 Отримання даних з серверу. Обробка подій. Основи HTML форм

**ТЕМИ**

**30. Взаємодія з власним або відкритим API**

* Огляд задачі, аргументація необхідності роботи з зовнішніми даними.
* Отримання даних з серверу за допомогою jquery ajax.
* Відправка команд та даних на сервер.
* Приклади взаємодії з серверами, що надають інформацію в публічний доступ (новини google або NY Times, котування валют, тендери prozorro).

**Практика:**

* Практикуємося з кодом для отримання зовнішніх даних та їх обробкою (на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

1. **Обговорюємо разом домашнє завдання**

* Ментору бажано мати свій варіант реалізованого завдання.
* Дивимось варіанти учасників. Якщо є варіант, який можна швидко доробити - дороблюємо разом.
* Якщо на кількох з працюючих учасників можна виділити типові проблеми - обговорюємо і, при потребі, повторно демонструємо щось з попередніх занять.

1. **Отримання даних ззовні - мотивація та постановка задачі для заняття**

* Окреслюємо задачу та мотивацію обміну даними між клієнтом та сервером в веб. Оглядово проходимось по поняттях “клієнт” та “сервер”, варто потребу в базі даних (в принципі, той самий інтернет магазин є зрозумілим прикладом, але варто описати процес роботи - закупщики вносять товари, менеджери обробляють замовлення, користувачі шукають товари та роблять замовлення… Також варто навести ще які-небудь видовищні приклади - пошуковий сервер гугла, соц. мережа фейсбук…)
* Нагадуємо про те, що для самої передачі даних в мережі часто використовується JSON. Можна розповісти про те, що для цього дані перетворюються саме в рядкове представлення, тому для кожної технології потрібні “серіалізатори” та “парсери”. Продемонструвати, як виглядає рядок JSON.
* Окреслюємо задачу, яку пропонуємо зробити разом. Задача повинна включати прийняття деякого вводу від користувача, отримання даних з сервера і використання цих даних для виконання дій за командою користувача. Як варінат - можна реалізувати сторінку - конвертер валют, отримуючи дані про курси з сайту НБУ в json-форматі. На сторінці буде поле вводу суми, випадаючий список для вибору валюти, та кнопка “порахувати”. Варіант результату і мінімалістичної реалізації (HTML + JS):

<https://sergdeni33.github.io/WebDevDemoShop/JsStart/js-ajax-pure.html>  
<https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/js-ajax-pure.html>  
<https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/js/course-converter.js>

1. **Наголошуємо, що для виконання задачі нам необхідно базово освоїти використання обробників подій та елементів керування - та пропонуємо групі створювати сторінку разом з ментором.**

* Створюємо сторінку, в яку додаємо поле вводу (input type=’text’) (для суми), поле вибору валюти (select + options), кнопку та поле виводу результату.
* По ходу базово розповідаємо про поля вводу тексту та поля вибору в HTML.
  + - * За поглибленим вивченням рекомендуємо користуватись документацією, а також кажемо, що детальніше про форми та поля вводу поговоримо на наступному занятті.
      * В поле вибору валюти вбиваємо руками два - три варіанти для демонстрації - звертаючи увагу на різницю між заголовком варіанту (текстом, наприклад може бути “Доллар США”) та значенням, яке ми будемо використовувати в коді (наприклад, код валюти “USD”).
* Базово розповідаємо про обробники подій. Варто згадати, що саме реакція на події (особливо на дії користувача) була однією з основних цілей створення скриптових мов в браузерах (javascript). Разом з групою створюємо функцію і призначаємо її обробником події натиснення кнопки “порахувати”.
* Приходимо до того, що для подальшої реалізації конвертера валют нам необхідно при показі сторінки користувачу мати в доступі нашого скрипта масив з курсами - щоб при натисненні кнопки у обробнику подій виконати підрахунки. Крім того, в момент показу сторінки ми повинні заповнити список доступних валют, щоб користувач міг реально їх обирати.

1. **Отримання даних з сервера - реалізація**

* Розповідаємо про те, що задача виконати код відразу після завантаження сторінки є важливою та часто зустрічається, і для цього можна також використати обробник події - але вже не такої, яка генерується користувачем, а вбудованої.
* Демонструємо (реалізуємо разом з групою) виклик нашої функції на подію onload в body. Варто згадати, що як правило використовується більш потужний (але і більш складний) варіант призначення цього обробника - і відіслати бажаючих до документації.
* Розповідаємо концепцію AJAX. В тому числі про мотивацію виникнення та значення.
* Демонструємо в мінімальному вигляді використання вбудованого об’єкту XMLHttpRequest (вірогідно, варто обійтись без пояснення оператора new - кажемо, що так створюються), без сумісності з 6-м експлорером та повної перевірки проблем. Один з варіантів коду - за посиланням вище.
* Розповідаємо про проблему синхронності та асинхронності (сторінка не повинна підвисати, поки очікує виконання ajax-запитів), базово пояснюємо поняття колбеків (функцій зворотнього виклику), на прикладі призначення обробника onreadystatechange.
* Разом з групою реалізуємо отримання даних про курси валют з сайту НБУ, і   
  в обробнику подіі отримання результату заповнюємо випадаючий список вибору валюти для користувача (можна зробити окремою функцією, яку викликати та дати їй масив курсів з сервера). Також в цьому обробнику запам’ятовуємо отримані дані в глобальну змінну.
* Нарешті разом реалізуємо до кінця реакцію на кнопку “Порахувати” - так як теперь при завантаженні сторінки ми отримуємо всі потрібні дані. По ходу показуємо, як отримати обране значення в випадаючому списку валют, та прочитати введену в текстове поле суму.
* Варіант ресурсу з документацією по ajax:

<https://developer.mozilla.org/ru/docs/Web/Guide/AJAX/Getting_Started>  
(є переклад російською)

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “jQuery - AJAX Introduction” (англ)](https://www.w3schools.com/jquery/jquery_ajax_intro.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery - AJAX get() and post() Methods” (англ)](https://www.w3schools.com/jquery/jquery_ajax_get_post.asp)
* [Mozilla Developer Network (MDN) «Third party APIs» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Client-side_web_APIs/Third_party_APIs)
* [Mozilla Developer Network (MDN) «Fetching data from the server» (англ)](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data)
* [Офіційний веб-сайт фреймворку jQuery, навчальний розділ “jQuery’s Ajax-Related Methods” (англ)](http://learn.jquery.com/ajax/jquery-ajax-methods/)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо групі додати до свого “каталогу плиткою, якій генерується за масивом даних про товари” (попереднє завдання), наступну функціональність:
    - * При завантаженні сторінки, отримати курси валют НБУ, і запропонувати користувачу список вибору валют та кнопку “Застосувати”.
      * По натисненню цієї кнопки, перегенерувати каталог товарів, показавши ціну в обраній валюті (вважати, що в масиві даних для каталога ціна задана в національній валюті).
  + Якщо ментору вдасться знайти яке-небудь інше відкрите джерело не складних даних, можна запропонувати групі створити сторінку, яка буде генерувати свій контент на їх основі. Наприклад, це може бути сторінка із списком новин з гугл (апі відкрите, але на даний момент потребує пройти процедуру реєстрації та отримання ключа).
  + Можна також запропонувати вивести “курси валют плиткою”, на основі тих самих даних - так само, як в сторінці - прикладі за посиланням.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Стаття про OpenProcurement на Вікіпедії (укр)](https://uk.wikipedia.org/wiki/OpenProcurement)
* [Сайт документації по OpenProcurement, розділ «Туторіал. Розглянемо основні правила» (укр)](http://api-docs.openprocurement.org/uk_UA/latest/tutorial.html#creating-tender)
* [Галерея API на офіційному веб-сайт NY Times (англ)](https://developer.nytimes.com/)
* [Google Maps API на офіційному веб-сайті Google (рос)](https://developers.google.com/maps/?hl=ru)

# ЗАНЯТТЯ 14 Робота з даними користувача. Продовження форм. Огляд відправки даних на сервер

**ТЕМИ**

**31. Отримання даних від користувача та робота з ними**

* Форми вводу даних. Стандартні елементи.
* Валідація (перевірка) даних на боці клієнта.
* Збереження даних на боці клієнта - localstorage, session storage.
* Побудова логіки сторінки, орієнтованої на роботу з даними.

**Практика:**

* Створюємо форму входу на сайт та інтегруємо її у сторінку (на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

* + - 1. **Обговорюємо разом домашнє завдання**
* Бажано, щоб ментор мав свій варінат розв’язку.
* Дивимось варіанти учасників. Якщо є підходящі варіанти, то дороблюємо разом з групою один-два з них (там, де є напрацювання, але щось не вийшло, і можна доробити досить швидко).
* Якщо доцільно, демонструємо свій варіант, та обговорюємо його з групою.
  + - 1. **Надання користувачу можливості вводу даних**
* Робимо мотиваційний вступ
  + - * Наголошуємо на тому, що майже кожен веб-сайт так чи інакше включає сценаріі, де користувач повинен вводити інформацію (від логіну та паролю до форм зворотнього зв’язку)
      * Аргументуємо, що в більшості випадків, введені дані повинні відправлятися з клієнта (браузера) на сервер, та впиливати на поведінку системи як на боці клієнта, так і на боці сервера (дані про замовлення товарів, профілі користувача, логін пароль для входу, відгуки, обрані фільтри…).
* Описуємо дві частини задачі
  + - * дати можливість користувачу ввести дані
      * власне, відправка даних на сервер в зрозумілому для нього форматі
* Розповідаємо про форми та експерементуємо разом з групою
  + - * Показуємо базовий синтаксис форми, приклад з текстовими полями вводу
      * Демонструємо стилізацію за допомогою bootstrap (також базово, без розбору всіх варіантів - просто щоб форма виглядала сучасно)
      * Додаємо кілька найбільш вживаних типів полів вводу, окрім текстових (select, password, checkbox…)
      * Розповідаємо про поняття “відправки форми” - submit, і про необхідність задання дії (action) форми.
      * Поверхнево згадуємо про метод відправки форми (method), та про різницю між GET та POST.
      * Створюємо тестову сторінку - “приймач” - ту, на яку відправляємо нашу форму, використовуючи метод GET та атрибут action з відповідним url. Додаємо кнопку “відправити”.
      * Показуємо, що при методі GET параметри форми потрапляють в адресний рядок браузера - додаються до посилання.
      * Обговорюємо принципи, за якими це відбувається, пояснюємо атрибут name у полів вводу.
      * Демонструємо, що відбувається, якщо ввести в поле текст українською, або ввести слова латиницею але з пробілом. Обговорюємо поняття url encoding.
* Пропонуємо групі разом з ментором розробити сторінку “входу на сайт” (логіну) - і “захищену сторінку” сайту, яка імітує перевірку імені користувача та паролю, виконуючи цю перевірку на боці клієнта. Приклад реалізованих сторінок:

<https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/login.html>  
<https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/login-result.html>  
<https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/js/check-login.js>

* + - * Створюємо просту бутстрап-стилізовану форму з полями “ім’я користувача”, “пароль” та “запам’ятати мене” (checkbox). По натисненню кнопки “Увійти” переходимо на іншу нашу сторінку - “дані користувача”.
      * На сторінці результату реалізуємо скріпт, який відразу при її відкритті виймає з адресного рядка браузера передані параметри, “парсить” їх (попередньо зробивши urlDecode), звіряє ім’я та пароль з деякими тестовими, які можна прописати прямо в коді сторінки - наприклад, зробити там масив допустимих користувачів.
      * В залежності від “вірного” чи “невірного” логіну, або показуємо сторінку - щось на зразок “Вітаємо, Олексій!” - або ж перенаправляємо браузер назад на сторінку вводу даних (по ходу демонструємо роботу з об’єктом location).
      * Якщо вистачає часу, можна додати до форми також деяке поле вибору, і на сторінці результату відображати обране значення.
      * Якщо група достатньо сильна, можна реалізувати функціональність чекбоксу “Запам’ятати мене”, використовуючи localstorage (див. приклад за посиланням).
      1. **Обговорюємо отриманий результат, намагаємось провокувати питання в учасників та обговорення.** Ще раз наголошуємо, що це була демонстрація технологій - і в реальності не можна зберігати та перевіряти логін-пароль на клієнті - при цьому принципи пердачі отримання параметрів і т.д. залишаються тими самими і в варіанті з реальною серверною частиною. Описуємо, як відбувається взаємодія зі справжнім сервером. Можна показати запити на якомусь прикладі, використовуючи консоль браузера або інструмент типу Fiddler.

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “JavaScript Forms” (англ)](https://www.w3schools.com/js/js_validation.asp)
* [Курс з веб-розробки на W3Schools, тема “HTML5 Web Storage” (англ)](https://www.w3schools.com/html/html5_webstorage.asp)
* [Mozilla Developer Network (MDN) «Your first HTML form» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Forms/Your_first_HTML_form)
* [Mozilla Developer Network (MDN) «Form data validation» (англ)](https://developer.mozilla.org/uk/docs/Learn/HTML/Forms/Form_validation)
* [Mozilla Developer Network (MDN) «Web Storage API» (англ)](https://developer.mozilla.org/uk/docs/Web/API/Web_Storage_API)
* [Mozilla Developer Network (MDN) «Third party APIs» (англ)](https://developer.mozilla.org/uk/docs/Learn/JavaScript/Client-side_web_APIs/Third_party_APIs)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Пропонуємо створити сторінку входу до нашого “магазину”, на якій користувач вводить логін пароль, та обирає валюту. Після цього він потрапляє на сторінку “каталогу плиткою” на основі попереднього завдання, де додаємо перевірку логіну та паролю. При невірному введенні - повертаємо його на сторінку входу. При вірному - відображаємо каталог та ціни в обраній валюті, і біля кожного товару кнопку “придбати”, а також відображаємо десь на сторінці “вітання” - “Ви зайшли як …”, та кнопку “Вийти”.
  + Сторінка каталогу повинна, як і раніше, допускати перегляд і без логіну та паролю - але у такому випадку біля товарів не повинно бути кнопки “придбати”, ціна повинна бути в гривні, і на сторінці не повинно бути “вітання” та “вийти” - навпаки, повинне бути посилання “Увійти”, яке веде на сторінку вводу логіну та паролю.

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Пример из жизни: валидация формы при помощи jQuery» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/jsquery-real-example-HTML-form-validation-via-jquery)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Пример из жизни: валидация формы при помощи JS» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/js-form-validation-js)

# ЗАНЯТТЯ 15 Огляд поточного стану технологій веб-розробки. Заключне спілкування

**ТЕМИ**

**32. Сучасні підходи та інструменти веб-розробника**

* CMS
* Системи контроля версій
* Системи генерації CSS
* Інструменти збирання проектів
* JS frameworks

**33. Бібліотека jquery**

* Проблематика, історія виникнення, переваги та недоліки.
* Підключення. Точка входу для логіки сторінки.
* Маніпулювання DOM за допомогою jquery.
* jquery як база для стандартних рішень задач front-end

**34. Заключне заняття**

* Обговорення питань по проектам
* Окреслення напрямків подальшого розвитку
* Подяка і вільне спілкування

**Практика:**

* Практикуємося з однією із систем CMS (на вибір ментора), пробуємо маніпулювати з DOM за допомогою jquery (приклади на вибір ментора)

**РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ**

**Обговорюємо разом домашнє завдання**

* Бажано, щоб ментор мав свій варінат розв’язку.
* Дивимось варіанти учасників, коментуємо, якщо потрібно, розповідаємо, як їх доробити.
* Якщо доцільно, демонструємо свій варіант, та обговорюємо його з групою.

**Ментор розповідає про своє бачення галузі веб-розробки, та вказує місце того матеріалу, який було розглянуто в курсі, серед усіх задач та підходів.**

* Ще раз коротко нагадуємо про історію розвитку веб сайтів від простих статичних до повноцінних “товстих” застосувань всередині браузера. Описуємо виклики, яке це створює для розробників та індустрії.
* Вказуємо, що “простий старий html+css+js”, які ми розглядали, є лише базою - і використовуючи їх в чистому вигляді, для створення складних сайтів доведеться витратити забагато часу. Плюс підтримка і доробка, а також командна взаємодія (в тому числі між дизайнером - верстальником та програмістом) на великих проектах потребує більш потужних інструментів.
* Розповідаємо про одну з надбудов над CSS - бажано, щоб ментор підготував яку-небудь демонстрацію, щоб швидко показати. На даний момент, вірогідно, доцільно розповідати про SAAS або LESS. Не забути наголосити на мотивації - навіщо це потрібно.
* Розповідаємо про сучасний стан розвитку javascript.
  + - * Про об’єктний стиль (ECMAScript 2015) - знову бажано мати маленьку демонстрацію.
      * Ще раз згадуємо про “browser hell” - ситуацію, коли в різних браузерах все працює по-різному. Описуємо, які виходи з цього використовує індустрія:
      1. jQuery (маленька демонстрація, мета створення та переваги над чистим JS). Кажемо, що останні роки вважається не рекомендованим підходом - тому що замість мислення на рівні реалізації бізнес логіки, розробник повинен мислити на рівні безпосереднього маніпулювання елементам інтерфейсу.
      2. Поліфіли (якщо ментор має досвід з якимось з них - також коротке демо).
      3. Розповідаємо, що бажано при розробці базуватися на перевірці наявних можливостей та реалізовувати плавну деградацію  
         <https://developer.mozilla.org/en-US/docs/Glossary/Graceful_degradation>
      4. Сучасні JS фреймворки.
         1. Розповідаємо про мотивацію створення - “чому React або Angular це не JQuery”. Якщо ментор має досвід з якимось з фреймворків, можна зробити мінімальну демонстрацію. Мабуть, найбільш зрозумілим і цікавим буде демо зв’язки з даними (data binding).
         2. Вказуємо і на недоліки при роботі з фреймворками - можливо, можна назвати ситуацію “framework hell”.
      5. Інструменти збірки, керування залежностями та автоматизації підготовки проекту. Можна розповісти зовсім мінімально, але щоб всі знали, що вони є. На прикладі чогось з webpack, gulp, grunt і т.д.
      6. NodeJS - сказати, що класна штука, і в тому числи можна виконувати javascript на сервері.
      * Окреслюємо напрямки та підходи до подальшого розвитку в напрямку веб-розробки для тих учасників, хто потенційно збирається розвиватися в ції галузі. Варто згадати, що на базі веб-застосувань часто створюються мобільні застосування в рамках підходу “створив один раз, працює всюди” (наприклад, React Native).

**Ведемо вільну бесіду з групою - питання, розмови на будь-які теми.**

**З бажаючими обговорюємо фінальні проекти, даємо рекомендації. Окреслюємо терміни та способи комунікації (звернення за допомогою та порадами) до ментора щодо доробки фінальних проектів.**

**Попередньо плануємо наступну зустріч для захисту проектів - дякуємо всім!**

**ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)**

* [Курс з веб-розробки на W3Schools, тема “jQuery Introduction” (англ)](https://www.w3schools.com/jquery/jquery_intro.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery Get Started” (англ)](https://www.w3schools.com/jquery/jquery_get_started.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery Syntax” (англ)](https://www.w3schools.com/jquery/jquery_syntax.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery Selectors” (англ)](https://www.w3schools.com/jquery/jquery_selectors.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery - Add Elements” (англ)](https://www.w3schools.com/jquery/jquery_dom_add.asp)
* [Курс з веб-розробки на W3Schools, тема “jQuery - Remove Elements” (англ)](https://www.w3schools.com/jquery/jquery_dom_remove.asp)
* Офіційний веб-сайт фреймворку jQuery, навчальний розділ “How jQuery Works” (англ)

**ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)**

* + Підготовка проектів до захисту

**ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ**

* [Навчальна платформа Devionity, курс Основы CSS, заняття «Что такое jQuery. Подключение jQuery» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/what-is-jquery-including-jquery)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «Основы jQuery. Селекторы» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/jquery-syntax-basics-selectors)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «jQuery: изменение элементов DOM и стилей» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/jquery-changing-elements-css)
* [Навчальна платформа Devionity, курс Основы CSS, заняття «jQuery: добавление и удаление элементов DOM» (рос)](https://devionity.com/ru/courses/js-fundamentals-and-jquery/jquery-adding-removing-dom-elements)

# КУРСИ ДЛЯ ПАРАЛЕЛЬНОГО ВИВЧЕННЯ

* Курс «Основи Web UI розробки» від Prometheus (укр)
* [Coursera, курс “Responsive Website Basics: Code with HTML, CSS, and JavaScript” від University of London (англ, є український транскрипт)](https://www.coursera.org/learn/website-coding)
* [edX, курс “Microsoft Professional Orientation: Front End Web Development” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV237x+4T2017/course/)
* [edX, курс “ Microsoft: DEV203x Introduction to Bootstrap” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV203x+4T2017/course/)
* [edX, курс “Microsoft: DEV211.1x Introduction to HTML and JavaScript” від Microsoft (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV211.1x+4T2017/course/)
* [edX, курс “PennX: SD4x Programming for the Web with JavaScript” (англ, є англ субтитри)](https://courses.edx.org/courses/course-v1:Microsoft+DEV211.1x+4T2017/course/)
* Серія уроків по Bootstrap від програміста Назара Токаря (укр/рос)

# КОПІРАЙТ

Дана програма є розробкою Благодійної організації «Благодійний фонд «Брейнбаскет Фаундейшн» та розповсюджується на умовах ліцензії Creative Commons CC BY-NC 4.0.



**Із Зазначенням Авторства — Некомерційна 4.0 Міжнародна (CC BY-NC 4.0)**

Це спрощений виклад (а не замінник) [ліцензії](https://creativecommons.org/licenses/by-nc/4.0/legalcode.uk). [Відмова від відповідальності](https://creativecommons.org/licenses/by-nc/4.0/deed.uk).

**Ви можете вільно:**

* **Поширювати**— копіювати і розповсюджувати матеріал у будь-якому вигляді чи форматі
* **Змінювати** — реміксувати, трансформувати, і брати матеріал за основу

Ліцензіар не може відкликати ці дозволи поки ви дотримуєтесь умов ліцензії.

**На наступних умовах:**

* **Зазначення Авторства** — Ви маєте [вказати автора](https://creativecommons.org/licenses/by-nc/4.0/deed.uk), розмістити посилання на ліцензію та [вказати чи було внесено зміни до твору](https://creativecommons.org/licenses/by-nc/4.0/deed.uk). Ви можете зробити це у будь-який розумний спосіб, але так, щоб не створювати враження стосовно того, що ліцензіар підтримує чи схвалює вас або ваше використання твору.
* **Некомерційна** — Ви не можете використовувати матеріал для [комерційних цілей](https://creativecommons.org/licenses/by-nc/4.0/deed.uk).

Без додаткових обмежень — Ви не можете висувати додаткові умови або застосовувати [технологічні засоби захисту,](https://creativecommons.org/licenses/by-nc/4.0/deed.uk) що обмежують права інших на дії дозволені ліцензією.