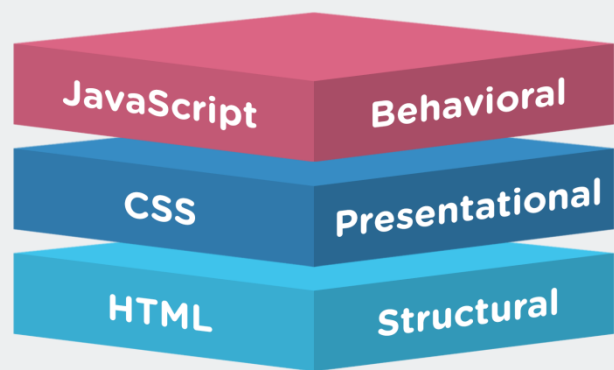


ЗАНЯТТЯ 11

АДАПТИВНИЙ ДИЗАЙН. ЗНАЙОМСТВО З BOOTSTRAP.



Курси "Веб-розробка"
за програмою
BrainBasket
"WOW Teachers"

ЗМІСТ ТЕМИ

1. Медіа запити.

- a) Медіа запити як інструмент значно підвищити гнучкість макета
- b) media types
- c) media features.

ЗМІСТ ТЕМИ

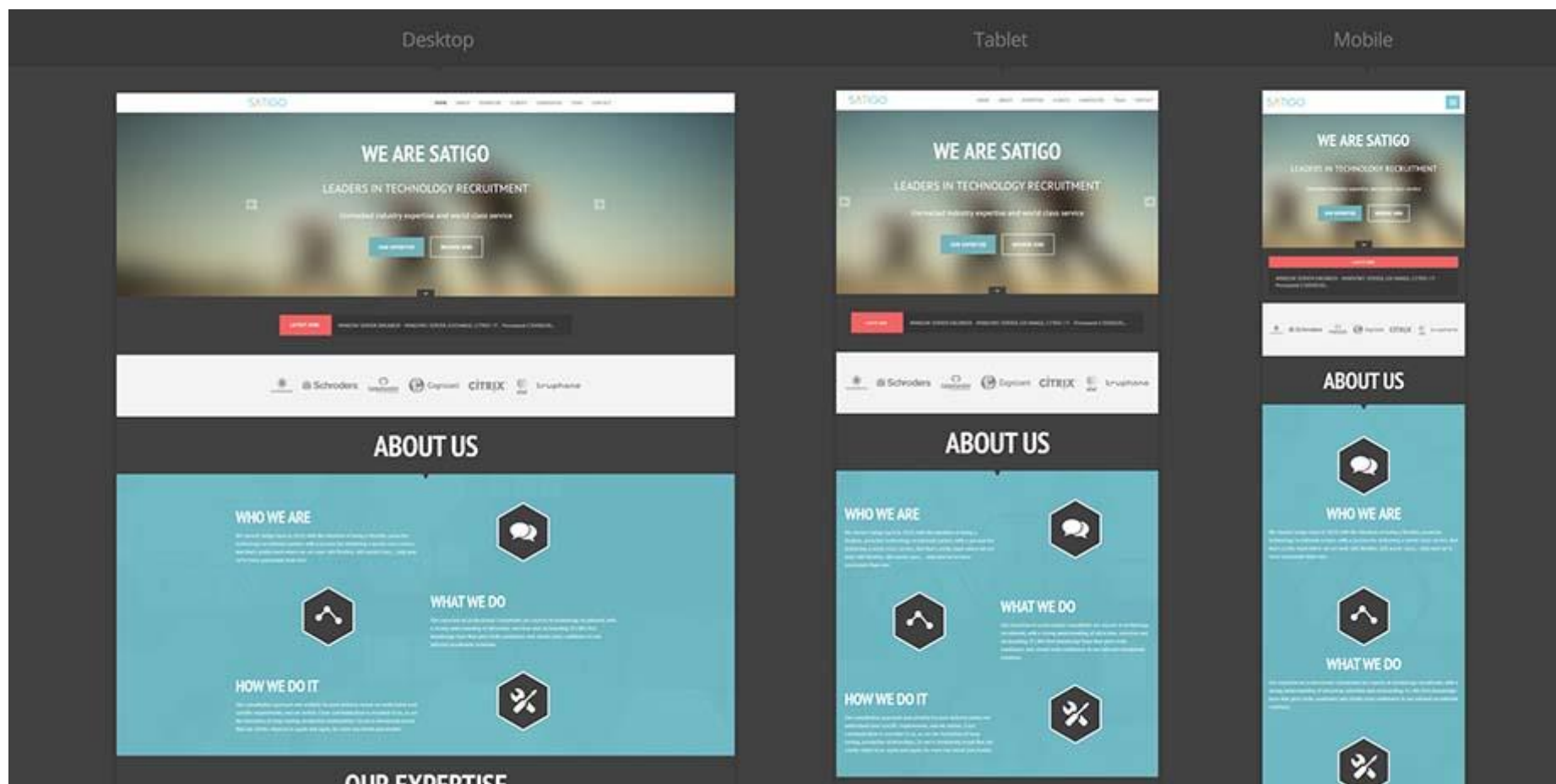
2. Адаптивний макет.

- a) Проблема багатьох пристроїв. Поняття адаптивного макету.
- b) Область перегляду (viewport).
- c) Медіа запити для побудови адаптивного макету. Точки зміни (breakpoint)

ЗМІСТ ТЕМИ

3. Twitter bootstrap - знайомство.

- a) Каркаси CSS (CSS Frameworks) - мета створення і проблематика
- b) Bootstrap як один з основних каркасів
- c) Підключення. Залежність від JQuery
- d) Система сітки в Bootstrap
- e) Основний контейнер - класи container та container-fluid



Відображення сайту на різних пристроях. Макет сторінки набуває змін.

1_a Медіа запити як інструмент значно підвищити гнучкість макета.

У 2001 році в HTML4 і CSS2 була введена підтримка апаратно-залежних таблиць стилів, що дозволила створювати стилі та таблиці стилів для певних типів пристроїв.

У загальному випадку **медіа-запит** складається з ключового слова, що описує тип пристрою (необов'язковий параметр) і виразу, який перевіряє характеристики даного пристрою. З усіх характеристик найчастіше перевіряється ширина пристрою **width**. Медіа-запит є логічним виразом, яке повертає значення істини або брехні.

Всі запити починаються з правила **@media**, після чого слідує умова, в якій використовуються типи носіїв, логічні оператори і медіа-функції.

1_a Медіа запити як інструмент значно підвищити гнучкість макета.

Медіа-запити можуть бути додані наступними способами:

1. За допомогою HTML:

```
<link rel="stylesheet" media="screen and (color)"  
href="example.css">
```

2. За допомогою правила `@import` всередині елемента `<style>` або зовнішньої таблиці стилів:

```
@import url(color.css) screen and (color);
```

3. Безпосередньо в коді сторінки:

```
<style> @media (max-width: 600px) { #sidebar  
{display: none;} } </style>
```

4. Всередині таблиці стилів `style.css`:

```
@media (max-width: 600px) { #sidebar {display: none;}  
}
```

1_b media types.

Тип	Опис
all	Всі типи. Це значення використовується по замовчуванню.
braille	Пристрої, засновані на системі Брайля, які призначені для читання сліпими людьми.
embossed	Принтери, що використовують для друку систему Брайля.
handheld	Смартфони і аналогічні їм апарати.
print	Принтери і інші друкарські пристрої.
projection	Проектори.
screen	Екран монітора.
speech	Мовні синтезатори, а також програми для відтворення тексту вголос. Сюди, наприклад, можна віднести мовні браузері.
tty	Пристрої з фіксованим розміром символів (телетайпи, термінали, пристрої з обмеженнями дисплея).
tv	Телевізори.

1_c media features.

Логічні оператори, що застосовуються в медіа-запитах:

and

Логічне І. Вказується для об'єднання декількох умов.

Приклад. Стиль для всіх кольорових пристроїв

@media all and (color) {...}

not

Логічне НЕ. Вказується для заперечення умови.

Приклад. Стиль для всіх пристроїв крім смартфонів

@media all and (not handheld) {...}

only

Застосовується для старих браузерів, які не підтримують медіа-запити.

У списку немає логічного оператора АБО, його роль виконує кома.

Перерахування кількох умов через кому говорить про те, що якщо хоча б одна умова виконується, то стиль буде застосований.

1_c media features.

Параметр	Опис
width	Перевіряє ширину області перегляду. Значення задаються в одиницях довжини, px, em і т.д., наприклад, (width: 800px). Зазвичай для перевірки використовуються мінімальні і максимальні значення ширини. min-width застосовує правило якщо ширина області перегляду більше значення, зазначеного в запиті, max-width - ширина області перегляду менше значення, зазначеного в запиті.
height	Перевіряє висоту області перегляду. Значення задаються в одиницях довжини, px, em і т.д., наприклад, (height: 500px). Зазвичай для перевірки використовуються мінімальні і максимальні значення висоти. min-height застосовує правило якщо висота області перегляду більше значення, зазначеного в запиті, max-height - висота області перегляду якого менше значення, зазначеного в запиті.
aspect-ratio	перевіряє співвідношення ширини до висоти області перегляду. Широкоекранний дисплей зі співвідношенням сторін 16: 9 може бути позначений як (aspect-ratio: 16/9). min-aspect-ratio перевіряє мінімальне співвідношення, max-aspect-ratio - максимальне співвідношення ширини до висоти області перегляду.
orientation	Перевіряє орієнтацію області перегляду. Приймає два значення: (orientation: portrait) і (orientation: landscape).
resolution	Перевіряє дозвіл екрана (кількість пікселів). Значення також можуть перевіряти кількість точок на дюйм (dpi) або кількість точок на сантиметр (dpcm), наприклад, (resolution: 300dpi). min-resolution перевіряє мінімальний дозвіл екрана, max-resolution - максимальне.
color	Перевіряє кількість біт на кожен з кольорних компонентів пристрою виведення. Наприклад, (min-color: 4) означає, що екран конкретного пристрою повинен мати 4-бітну глибину кольору. min-color перевіряє мінімальну кількість біт, max-color - максимальна кількість біт.
color-index	Перевіряє кількість записів в таблиці підстановки квітів. Як значення вказується позитивне число, наприклад, (color-index: 256). min-color-index перевіряє мінімальну кількість записів, max-color-index - максимальна кількість записів
monochrome	Перевіряє кількість бітів на піксель монохромного пристрою. Значення задається цілим позитивним числом, наприклад, (min-monochrome: 8). min-monochrome перевіряє мінімальну кількість бітів, max-monochrome - максимальна кількість бітів.
-webkit-device-pixel-ratio	Задає кількість фізичних пікселів пристрої на кожен CSS-піксель.

Чутливий і адаптивний дизайн сайту

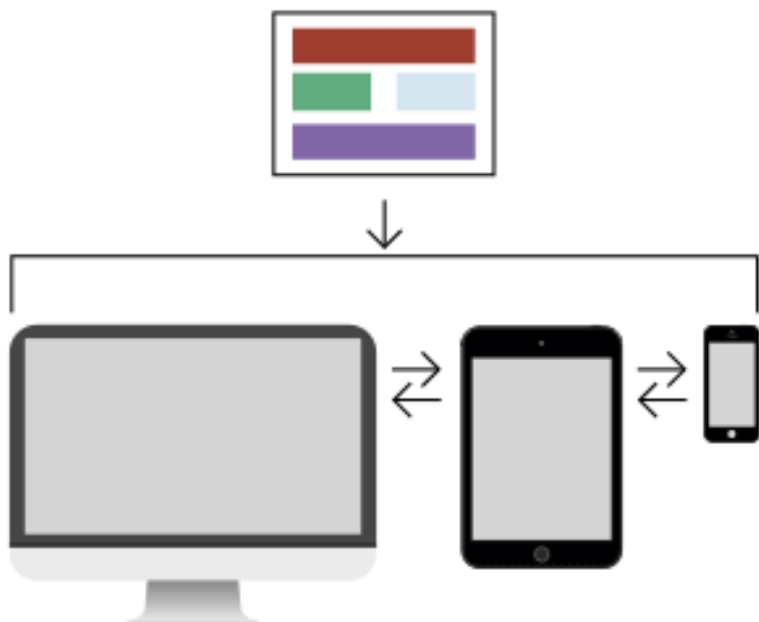
Інтернет-користувачі переглядають сайти на різних пристроях з екранами різних розмірів. Розміри екранів постійно змінюються, тому важливо щоб сайт адаптувався до будь-якого з них. Існує два основних підходи для створення сайтів, легко адаптуються для різних типів пристроїв:

Responsive Design (RWD) – чутливий (або гнучкий) дизайн - проектування сайту з певними значеннями властивостей, наприклад, гнучка сітка макета, які дозволяють одному макету працювати на різних пристроях;

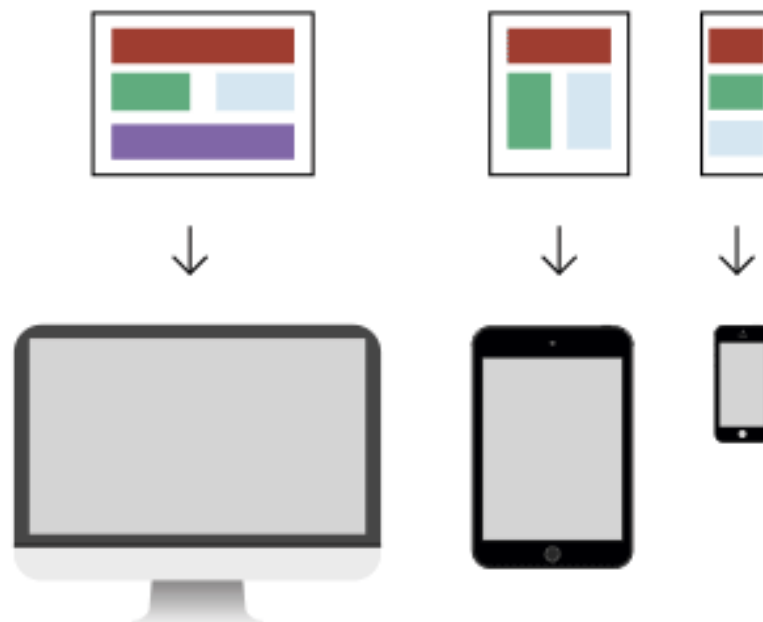
Adaptive Design (AWD) - адаптивний дизайн, або динамічний показ - проектування сайту з умовами, які змінюються в залежності від пристрою, базуючись на декількох макетах фіксованої ширини.

Відмінність між чутливим і адаптивним веб-дизайном

Responsive



Adaptive



2_а Проблема багатьох пристроїв. Поняття адаптивного макету.

Адаптивний дизайн (Adaptive Web Design) орієнтується на розміри пристроїв. Він використовує кілька статичних макетів для різних типів пристроїв (мобільні пристрої, планшети, настільні комп'ютери), базуючись на контрольних (переломних) точках. Тобто макети завантажуються при певних розмірах вікна браузера пристрою, а переходи між макетами відбуваються стрибкоподібно, а не плавно.

Зазвичай адаптивні макети мають шість варіантів макетів в залежності від ширини екрану:

320

480

760

960

1200

1600.

В адаптивних макетах головну роль грає функціональність, тобто при створенні дизайну враховуються особливості пристроїв, наприклад, сенсорне управління для мобільних пристроїв або великі простори для настільних моніторів.

2_b Область просмотра (viewport).

Для управления розміткою в мобільних браузерах використовується метатег viewport. Спочатку даний тег був представлений розробниками Apple для браузера Safari на iOS. Мобільні браузери відображають сторінки в віртуальному вікні перегляду, яке зазвичай ширше, ніж екран пристрою. За допомогою метатега viewport можна контролювати розмір вікна перегляду і масштаб.

Сторінки, адаптовані для перегляду на різних типах пристроїв, повинні містити в розділі <head> метатег viewport.

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Властивість width визначає віртуальну ширину вікна перегляду, значення device-width - фізичну ширину пристрою.

При першому завантаженні сторінки властивість initial-scale управляє початковим рівнем масштабування, initial-scale = 1 означає, що 1 піксель вікна перегляду = 1 піксель CSS.



2_с Медіа запити для побудови адаптивного макету.

```
/* Смартфони (портретний і альбомний режими) ----- */
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) {
/* Додаємо сюди стилі*/
}
/* Смартфони (альбомний режим) ----- */
@media only screen and (min-width : 321px) {
/* Додаємо сюди стилі*/
}
/* Смартфони (портретний режим) ----- */
@media only screen and (max-width : 320px) {
/* Додаємо сюди стилі*/
}
/* iPad (портретний і альбомний режими) ----- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) {
/* Додаємо сюди стилі*/
}
/* iPad (альбомний режим) ----- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) {
/* Додаємо сюди стилі*/
}
/* iPad (портретний режим) ----- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) {
/* Додаємо сюди стилі*/
}
/* Настільні комп'ютери і ноутбуки ----- */
@media only screen and (min-width : 1224px) {
/* Додаємо сюди стилі*/
}
```

2_с Медіа запити для побудови адаптивного макету.

```
/* Великі екрани ----- */
@media only screen and (min-width : 1824px) {
/* Додаємо сюди стилі*/
}
/* iPhone 5 (портретний і альбомний режими)----- */
@media only screen and (min-device-width : 320px) and (max-device-width : 568px) {
/* Додаємо сюди стилі*/
}
/* iPhone 5 (альбомний режим)----- */
@media only screen and (min-device-width : 320px) and (max-device-width : 568px) and (orientation :
landscape) {
/* Додаємо сюди стилі*/
}
/* iPhone 5 (портретний режим)----- */
@media only screen and (min-device-width : 320px) and (max-device-width : 568px) and (orientation : portrait)
{
/* Додаємо сюди стилі*/
}
```


2_d Точки зміни (breakpoint).

Mobile First або Desktop First?

desktop first:

1200px

992px

768px

480px

320px

`/* ----- */ /* ----- От маленьких экранов к большим ----- */ /* ----- */ /* Стили для очень маленьких экранов (телефоны в книжной ориентации, дисплей менее 576 пикселей). Записываются без медиа-запросов, поскольку в Bootstrap это значение по умолчанию */ /* Стили для маленьких экранов (телефоны в альбомной ориентации, дисплей 576 пикселей и более) */ @media (min-width: 576px) { ... } /* Стили для средних экранов (планшеты, дисплей 768 пикселей и более) */ @media (min-width: 768px) { ... } /* Стили для больших экранов (ПК, дисплей 992 пикселей и более) */ @media (min-width: 992px) { ... } /* Стили для очень больших экранов (ПК с большим монитором, дисплей 1200 пикселей и более) */ @media (min-width: 1200px) { ... } /* ----- */ /* ----- От больших экранов к маленьким ----- */ /* ----- */ /* Стили для очень больших экранов (дисплей 1200 пикселей и более). Записываются без медиа-запросов, так как контрольная точка для очень больших экранов не имеет верхней границы по ширине */ /* Стили для больших экранов (дисплей максимум 1199 пикселей) */ @media (max-width: 1199px) { ... } /* Стили для средних экранов (дисплей максимум 991 пикселей) */ @media (max-width: 991px) { ... } /* Стили для маленьких экранов (дисплей максимум 767 пикселей) */ @media (max-width: 767px) { ... } /* Стили для очень маленьких экранов (дисплей максимум 575 пикселей) */ @media (max-width: 575px) { ... }`

Кроме этого, в Bootstrap задействуются еще несколько вариаций контрольных точек для записи стилей под конкретный диапазон ширины экранов. При этом используются сразу оба условия — для минимума и для максимума:

`/* Стили для очень маленьких экранов (дисплей максимум 575 пикселей) */ @media (max-width: 575px) { ... } /* Стили для маленьких экранов (дисплей шириной от 576 пикселей до 767 пикселей) */ @media (min-width: 576px) and (max-width: 767px) { ... } /* Стили для средних экранов (дисплей шириной от 768 пикселей до 991 пикселей) */ @media (min-width: 768px) and (max-width: 991px) { ... } /* Стили для больших экранов (дисплей шириной от 992 пикселей до 1199 пикселей) */ @media (min-width: 992px) and (max-width: 1199px) { ... } /* Стили для очень больших экранов (дисплей минимум 1200 пикселей) */ @media (min-width: 1200px) { ... }`

3_а Каркаси CSS (CSS Frameworks) - мета створення і проблематика.

CSS-Framework ([акронім](#) [англ.](#) CSS FW) — [фреймворк](#), створений для полегшення роботи веб-розробника та веб-дизайнера, пришвидшення розробки та попередження максимальної кількості помилок.

Більшість сучасних фреймворків за стандартом містять так звані CSS файли-нормалізатори, призначення яких "онулення"(скидання) CSS до стандартного значення. Це дозволяє отримувати майже однаковий вигляд(дизайн) веб-сторінки, сайту в різних [браузерах](#) та [ОС](#).

Ознаками фреймворка вважають:

- він визначає правила побудови архітектури програмного коду;
- він зумовлює принципи структурування файлів;
- всередині себе має декілька бібліотек (файлів).

Види

- *чистий CSS-фреймворк* (або CSS-бібліотека)
- *Модульний CSS-фреймворк* (CSS-бібліотеки)
- Системи [CSS-сітки](#) (система CSS-розмітки)
- *UI CSS-фреймворк* (UI CSS FW)

[960 Grid System](#)
[Bootstrap](#)
[Foundation](#)
[Kube](#)
[Responsive Grid System](#)
[Yaml](#)

3_b Bootstrap як один з основних каркасів.

Bootstrap — це безкоштовний набір інструментів з відкритим [кодом](#), призначений для створення [веб-сайтів](#) та [веб-додатків](#), який містить [шаблони CSS](#) та [HTML](#) для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення [JavaScript](#). Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Bootstrap має модульну структуру і складається переважно з наборів [таблиць стилів LESS](#), які реалізують різні компоненти цього набору інструментів. Розробники можуть самостійно налаштовувати файли Bootstrap, обираючи компоненти для свого проекту.

Основні інструменти Bootstrap:

- **Сітки** (grid) — наперед задані, готові до використання колонки
- **Шаблони** (template) — фіксовані чи адаптивні шаблони сторінок
- **Типографіка** (typography) — опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо
- **Мультимедіа** (media) — засоби управління зображеннями та відео
- **Таблиці** (table) — засоби оформлення таблиць, які зокрема забезпечують сортування
- **Форми** (form) — класи для оформлення як форм, так і деяких подій
- **Навігація** (nav, navbar) — класи для оформлення вкладок, сторінок, меню і панелей навігації
- **Сповіщення** (alert) — класи для оформлення діалогових вікон, підказок і спливаючих вікон
- **Іконочний шрифт** (icon font) — набір іконок у вигляді шрифту, складається майже з 500 компонентів.

3_с Підключення. Залежність від JQuery.

tps://bootstrap-4.ru

Файлова структура Bootstrap

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphs-halflings-regular.eot
    ├── glyphs-halflings-regular.svg
    ├── glyphs-halflings-regular.ttf
    └── glyphs-halflings-regular.woff
```

Можна завантажити із сайту

<https://getbootstrap.com/>

Три варіанти завантаження:

1. Мінімізований код CSS, JavaScript і шрифти. Без документації та вихідних файлів.
2. Вихідний код Less, JavaScript, файли шрифтів і документація. Потрібно компілятор Less і інші настройки
3. Bootstrap перенесений з Less в Sass для легкого включення в Rails, Compass, або проекти Sass-only.

Підключення онлайн (Bootstrap CDN)

<!-- Latest compiled and minified CSS -->

<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">

<!-- Optional theme --> <link rel="stylesheet"

href="//netdna.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->

<script src="//netdna.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>

3_с Підключення. Залежність від JQuery.

Документація до Bootstrap:

<https://bootstrap-4.ru/>

У Bootstrap створено велику кількість компонентів для багатьох потреб: іконки символів, спадні меню, навігаційні панелі, повідомлення для попереджень, підказки та ще багато чого іншого. Детально на сайті:

<https://twbs.docs.org.ua/components/>

Зв'язок Bootstrap з JQuery

jQuery — популярна JavaScript-бібліотека з відкритим сирцевим кодом. Основне завдання jQuery — це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації [DOM](#) об'єктів за допомогою [CSS](#) та [XPath](#) селекторів. Також даний фреймворк надає інтерфейси для [Ajax-застосунків](#), обробників подій і простої анімації.

```
<script type="text/javascript" src="jquery.js"></script>
```

3_d Система сітки в Bootstrap.

Bootstrap має чутливу, "найперше – мобільну" рідинну систему сітки, яка масштабується до 12 колонок відповідно до розміру пристрою або до розміру оглядового вікна. Вона включає [предвизначені класи](#) для легкого налаштування шаблонів, а також потужні [mixins для генерації більш семантичних шаблонів](#).

Система сітки використовується для створення сторінок шаблону через серію рядків та колонок. Далі показано як працює система сітки Bootstrap:

- Для правильного вирівнювання та відступів, рядки повинні розміщуватись в межах **.container** (щоб мати фіксовану ширину) або **.container-fluid** (щоб мати повну ширину).
- Використовуйте **.row** для створення рядків.
- Класи, що починаються на **col-** використовуйте для створення колонок.
- Вміст повинен розміщуватись в межах колонок, та лише рядки можуть безпосередньо обгортати колонки.
- Використовуючи, наприклад **.row** та **.col-xs-4**, можна швидко створити шаблон сітки.
- Колонки мають внутрішні відступи (між їх рамками та вмістом) завдяки властивості **padding**. Ці внутрішні відступи є причиною того, чому рядки мають від'ємне значення **margin-left** та **margin-right**.
- Якщо проглянути [прикладі сітки нижче](#), то можна побачити, що вона ширша, за вміст, що не знаходиться в ній. Причина цього - згадане в попередньому пункті від'ємне значення в рядках. Також завдяки цьому, вміст в межах колонок сітки вишиковується в одну лінію з вмістом, що не знаходиться в сітці.
- Колонка сітки створюється через зазначення одного номера із загальної доступної "суми дванадцять" (тобто – загальна сума номерів колонок в одному рядку не повинна перевищувати **12**). Наприклад, щоб створити три однакові колонки, які розтягнуться на всю ширину рядка, у кожній із них необхідно встановити клас **.col-xs-4**.
- Якщо в одному рядку вміщено більше, ніж 12 колонок, кожену групу додаткових колонок [буде перенесено](#), як єдине ціле, на новий рядок.
- Класи сітки застосовуються до пристроїв, з шириною екрану більшою або рівною контрольній точці, яка встановлена для даного класу; і в той же час, класи сітки, призначені для пристроїв з вузьким екраном, ігноруються. Таким чином, застосування будь-якого класу **.col-md-** до певного елемента, буде впливати на його стилі не лише на пристроях середнього розміру, але також і на пристроях з широким екраном, якщо клас **.col-lg-** не представлено.

3_d Система сітки в Bootstrap.

Опції сітки

Прогляньте призначення певних аспектів системи сітки Bootstrap для різних пристроїв.

	Дуже вузькі пристрої Телефони (<768px)	Вузькі пристрої Планшети (≥768px)	Середні пристрої Монітори (≥992px)	Широкі пристрої Монітори (≥1200px)
Поведінка сітки	Горизонтальна завжди	Складена спочатку, але горизонтальна, коли ширша за контрольні точки		
Ширина контейнера	Немає (автоматична)	750px	970px	1170px
Префікс класа	.col-xs-	.col-sm-	.col-md-	.col-lg-
# колонок	12			
Ширина колонки	Автоматична	~62px	~81px	~97px
Ширина проміжку	30px (15px з кожної сторони колонки)			
Можуть вкладатись	Так			
Можуть зміщуватись	Так			
Порядок колонок	Так			

3_е Основний контейнер - класи container та container-fluid.

Можна перетворити будь-який шаблон сітки з фіксованою шириною, на шаблон з повною шириною, змінивши ваш самий верхній-зовні **.container** на **.container-fluid**.

```
<div class="container-fluid">  
  <div class="row">  
    ...  
  </div>  
</div>
```

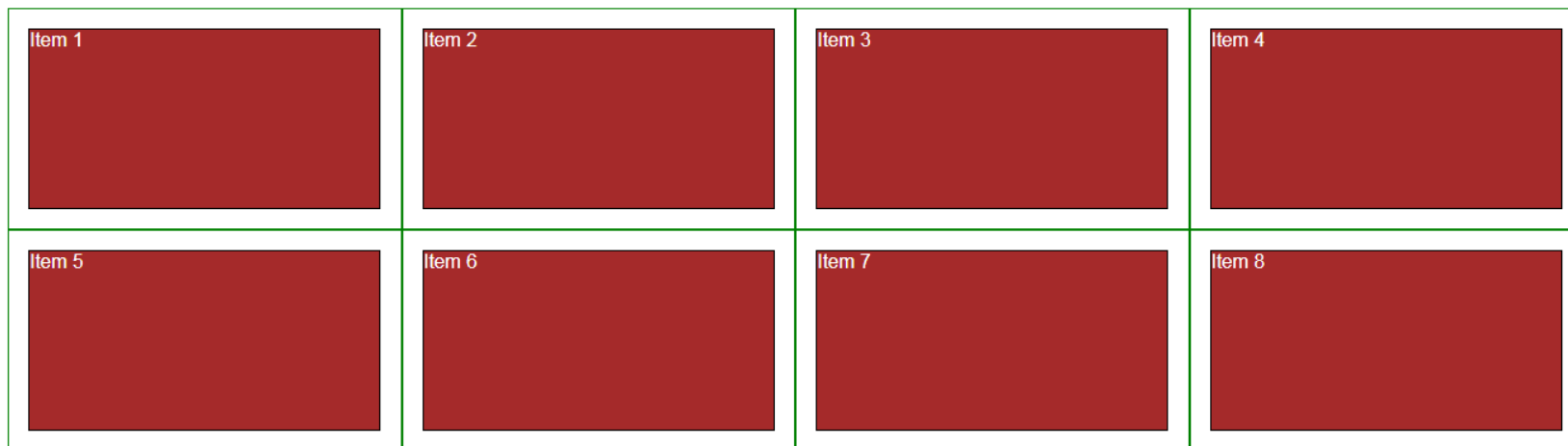

КЛАСНА РОБОТА (ПРАКТИКА)

Наділяємо документ властивостями адаптивного макету.

8 блоків на сторінці отримують різне відображення у залежності від розміру екрана.

Масштабування блоків градується за 12-рівнями від ширини.

Responsive Layout



Зразок можна знайти на хмарі

https://drive.google.com/drive/folders/1NxMUwroO7qYu_aF5qk-ZgAPhRcPRIKc

КЛАСНА РОБОТА (ПРАКТИКА)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Responsive Layout</title>
<style>

/***** Base styles *****/
* {
  box-sizing: border-box;
}
h1 {
  margin-bottom: 15px;
}

p {
  border: 1px solid black;
  background-color: #A52A2A;
  width: 90%;
  height: 150px;
  margin-right: auto;
  margin-left: auto;
  font-family: Helvetica;
  color: white;
}

/* Simple Responsive Framework. */
.row {
  width: 100%;
}
```

КЛАСНА РОБОТА (ПРАКТИКА)

/***** Large devices only *****/

```
@media (min-width: 1200px) {  
  .col-lg-1, .col-lg-2, .col-lg-3, .col-lg-4, .col-lg-5, .col-lg-6, .col-lg-7, .col-lg-8, .col-lg-9, .  
  col-lg-10, .col-lg-11, .col-lg-12 {  
    float: left;  
    border: 1px solid green;  
  }  
  .col-lg-1 {  
    width: 8.33%;  
  }  
  .col-lg-2 { width: 16.66%;  
  }  
  .col-lg-3 {  
    width: 25%; }  
  .col-lg-4 { width: 33.33%; }  
  .col-lg-5 {  
    width: 41.66%;  
  }  
  .col-lg-6 {  
    width: 50%;  
  }  
  .col-lg-7 {  
    width: 58.33%;  
  }  
  .col-lg-8 {  
    width: 66.66%;  
  }  
  .col-lg-9 {  
    width: 74.99%;  
  } .col-lg-10 {  
    width: 83.33%;  
  }  
  .col-lg-11 {  
    width: 91.66%;  
  }  
  .col-lg-12 {  
    width: 100%;  
  }  
}
```

КЛАСНА РОБОТА (ПРАКТИКА)

```
/****** Medium devices only *****/  
@media (min-width: 992px) and (max-width: 1199px) {  
  .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6, .col-md-7, .col-md-8, .col-md-9, .col-md-10, .col-md-11, .col-md-12 {  
    float: left;  
    border: 1px solid green;  
  }  
  .col-md-1 {  
    width: 8.33%;  
  }  
  .col-md-2 {  
    width: 16.66%;  
  }  
  .col-md-3 { width: 25%;  
  }  
  .col-md-4 {  
    width: 33.33%;  
  }  
  .col-md-5 { width: 41.66%;  
  }  
  .col-md-6 { width: 50%;  
  }  
  .col-md-7 {  
    width: 58.33%;  
  }  
  .col-md-8 { width: 66.66%;  
  }  
  .col-md-9 {  
    width: 74.99%;  
  }  
  .col-md-10 { width: 83.33%;  
  }  
  .col-md-11 { width: 91.66%;  
  }  
  .col-md-12 { width: 100%;  
  }  
}
```

</style>
</head>

КЛАСНА РОБОТА (ПРАКТИКА)

```
<body>
```

```
<h1>Responsive Layout</h1>
```

```
<div class="row">
```

```
  <div class="col-lg-3 col-md-6"><p>Item 1</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 2</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 3</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 4</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 5</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 6</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 7</p></div>
```

```
  <div class="col-lg-3 col-md-6"><p>Item 8</p></div>
```

```
</div>
```

```
</body>
```

```
</html>
```

ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)

- <https://html5book.ru/css3-mediazaproxy/>
- <http://htmlbook.ru/css/value/media>
- https://www.w3schools.com/css/css3_mediaqueries.asp
- <http://softwares.net.ua/css-media-zapyty-dlya-riznyh-prystroyiv/>
- https://html5css.ru/css/css3_mediaqueries.php
- <https://developer.mozilla.org/en-US/docs/Web/CSS/@media>
- <https://html5book.ru/adaptivnaya-vyorstka-sayta/>
- <https://html5book.ru/otzyvchivij-dizayn-saita/>
- <https://twbs.docs.org.ua/css/>
- <http://getbootstrap.ru/docs/v4-alpha/layout/grid/>
- <http://html-plus.in.ua/flex-utility-for-adaptive-layout-on-bootstrap4/>

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

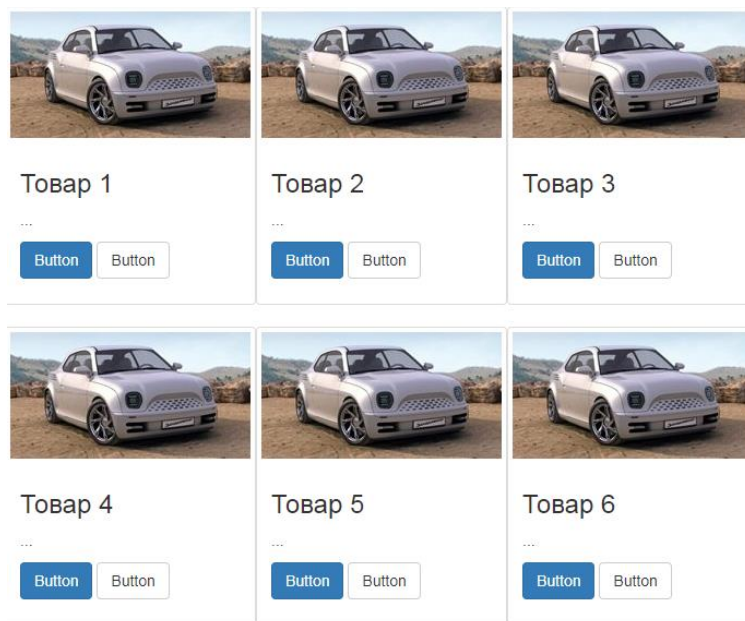
Варіант 1 (за зразком гіпертексту)

Створити веб-сторінку, на якій розміщено блоки плиткою, кожен з яких містить зображення, заголовок, текст та кнопки.

Усі елементи стилізовані засобами Bootstrap.

Використати bootstrap thumbnails:

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



ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

Варіант 2 (за зразком веб-сторінки)

Розробити власну веб-сторінку, використовуючи шаблон з бібліотеки Bootstrap

Можна повторити один з стартових шаблонів з сайту бутстрап – наприклад:

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

