



Адаптивна верстка

Модуль «Веб-технології»

Урок 12

Повторюємо

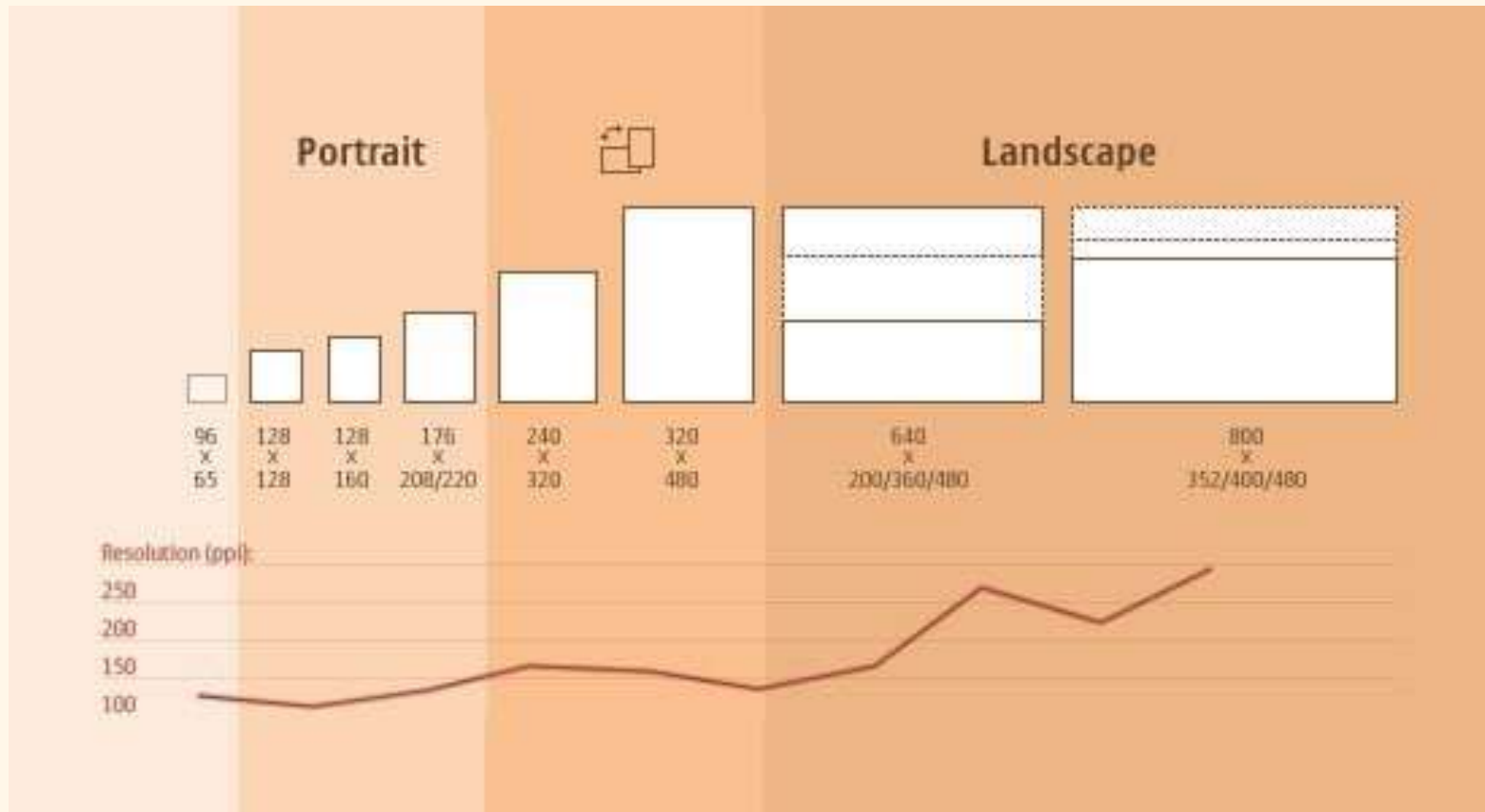


1. Сформулюйте правила структурування й іменування файлів.
2. Які типи макетів веб-сторінок ви знаєте? Які переваги та недоліки вони мають?
3. Назвіть етапи створення сайта
4. Що таке фрейворки?
5. Назвіть послідовність верстки веб-сторінки

Поміркуйте



- На рисунку представлена роздільність різноманітних девайсів.
- Як виглядатимуть сторінки сайту, які ви розробляли на минулих уроках на різних пристроях?



3 історії мобільного інтернету



- Веб-сайти десктопної версії мали широкий функціонал і яскраве оформлення, а отже довго завантажувалися та забирали занадто багато трафіку. Тому з'явилися **вап-сайти**, призначені для перегляду з мобільного телефону, які містили в основному текст і поодинокі зображення. Нагадаємо на той час мобільні телефони були слабкі, а мобільний Інтернет повільний і дуже дорогий.

The screenshot shows a mobile website with a blue header bar. The main content area is white and contains a list of links. The links are: [Sporting News](#) - Live sports 4 free - wap.upsnap.com/ - Call: 1-704-631-4060, [Find Headhunters in Seattle Now!](#), [\[1\] UEFA. Mobile Web page of European football](#), [\[2\] Eurosport. Latest sports news for mobile \(fr.\)](#), [\[3\] MLB. Sport news about baseball](#), and [\[4\] First4fl. Formula 1 news for mobile \(eng.\)](#). The text is in a small, sans-serif font, typical of early mobile web design.

SitiosWAP 2.0
wap.sitioswap.com

Sports News for Mobile

Ads by Google

[Sporting News](#) - Live sports 4 free -
wap.upsnap.com/ - **Call:**
1-704-631-4060

[Find Headhunters in Seattle Now!](#)

[1] UEFA. Mobile Web page of European football

[2] Eurosport. Latest sports news for mobile (fr.)

[3] MLB. Sport news about baseball

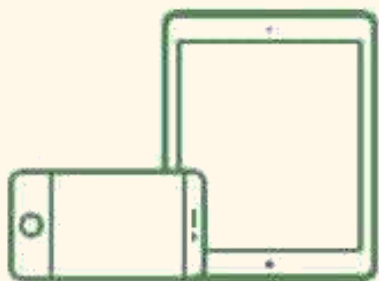
[4] First4fl. Formula 1 news for mobile (eng.)



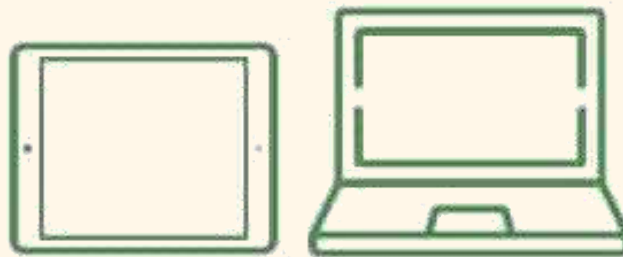
- Поява планшетів створила нові виклики – фіксована верстка не дозволяла змінювати розмір роздільності сайта. Великі сайти переглядати з планшетів було незручно, а мобільні сайти не були оптимізовані під екран великих розмірів.



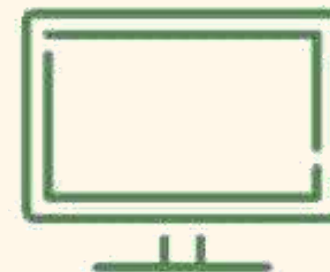
0-480



481-768



769-1279



1280+

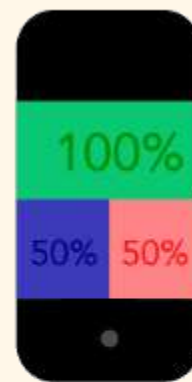
Гумова верстка



Макету і окремим його складовим надаються розміри, подані у відсотках, таким чином вміст сторінки розтягується. Для того, щоб розтягнути сайт і на 6-дюймовий екран смартфона і 42-дюймовий екран телевізора необхідно встановити максимальний і мінімальний розміри ширини (властивості max-width і min-width відповідно)



Фіксована



Гумова

Медіа-запити



- У 2006 році в інтернет-журналі A List Apart (<https://alistapart.com/>) вийшла стаття Марка ван ден Доббельстіна, який запропонував створити класи в залежності від пристрою та оголошувати кожному класу спеціальні стилі.
- Медіа-запити (*media queries*) — це правила CSS, які дозволяють керувати стилями елементів залежно від значень технічних параметрів пристроїв.
- У 2001 році в HTML4 і CSS2 була введена підтримка апаратнозалежних таблиць стилів, що дозволило створювати стилі й таблиці стилів для певних типів пристроїв.



Switchy McLayout: An Adaptive Layout Technique

by Marc van den Dobbelaer | December 10, 2009

Published in *Smashing*, 1710.

CSS-based liquid layout has proven successful during the reign of 800-pixel to 1024-pixel screens, but as we use a wider range of devices to access the web, we need more powerful and flexible ways of managing layout. If we want to serve devices whose viewports range from 240 pixels to about 1680 pixels—and with resolution ranging from 72 to 150 pixels per inch—we need a new method.

Медіа-запит є логічним виразом, який повертає істину або хибність.



```
@media(min-width:900px){p{color:red;}}
```



Оголошення
медіазапиту



Вираз перевіряє
відповідність розміру



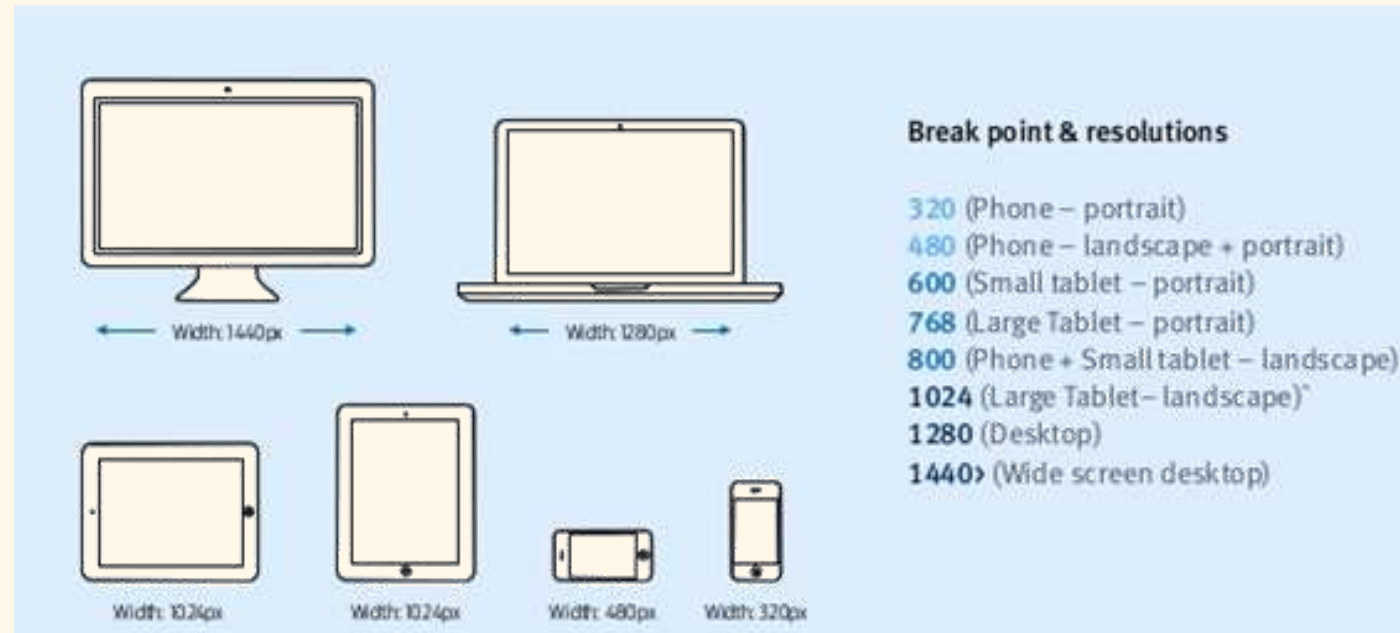
Стиль спрацює, якщо
умова виконується

Брейкпоінт

- прив'язка до конкретних пристроїв з певною роздільністю екрану. Стили перемикаються від одного брейкпоінта (контрольні точки, сталий термін у програмуванні, у випадку верстки — перемикання з одних умов на інші) до іншого, тобто насправді є фіксовані макети для кожного девайсу



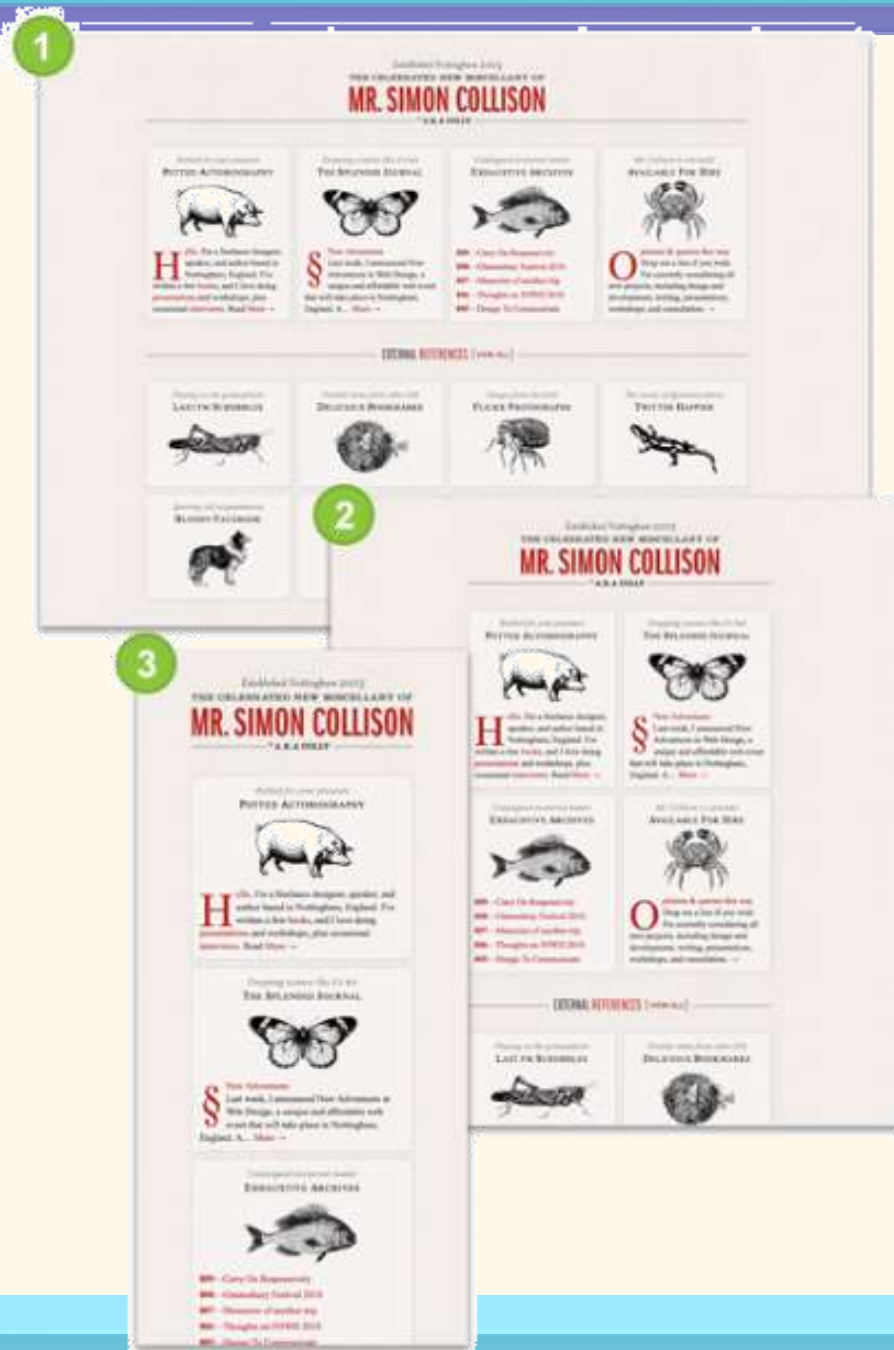
Breakpoints & Media queries



Поміркуйте

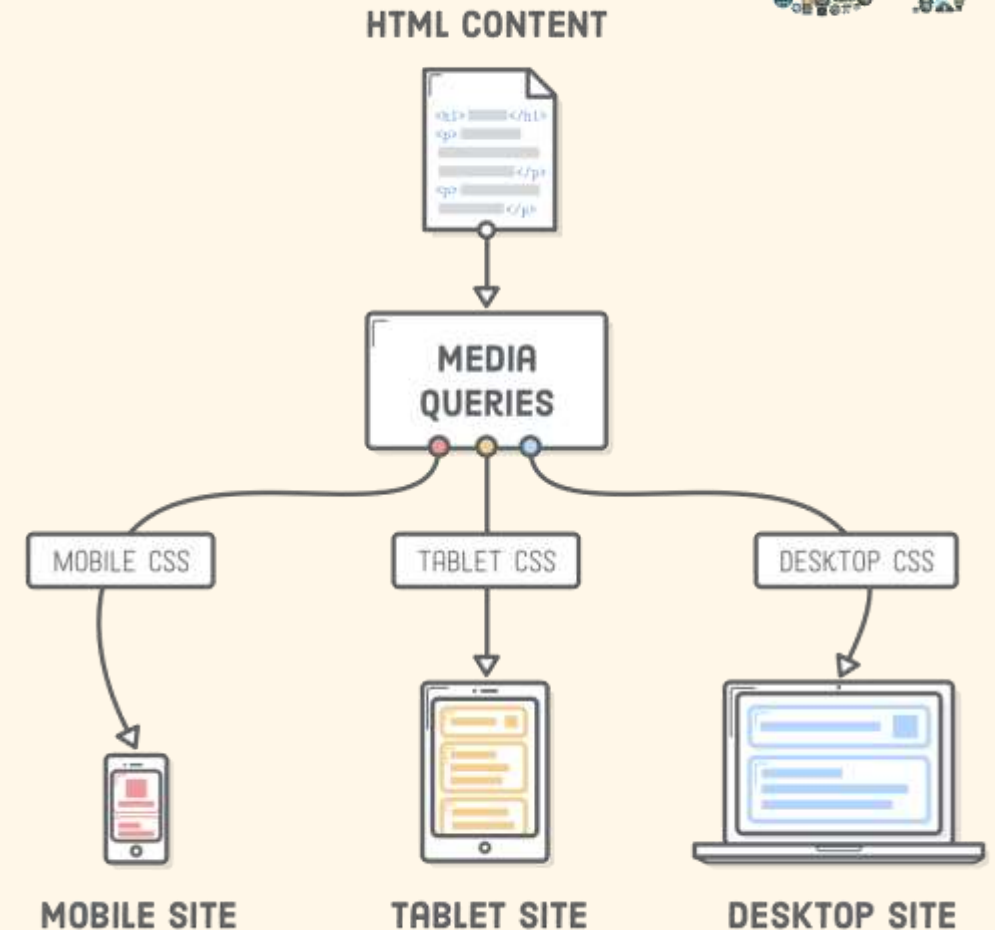
На рисунку
наведено
використання медіа-
запитів для різних
пристроїв

Визначте тип
прстроїв



Адаптивні сайти

- з'явилися завдяки зростанню попиту тих користувачів, які хочуть завантажувати веб-сторінки з ідеальною структурою на всіх своїх пристроях. Адаптивний дизайн стає обов'язковим. Він значно спрощує регулярний серфінг в Інтернеті незалежно від того, який пристрій вибрав користувач для доступу до сайтів, що його цікавлять



Цікаво



- Ідея розробки адаптивної верстки належить Аарону Густафсону. У своїй книзі «Адаптивний веб дизайн» він запропонував гнучко адаптувати сайти до можливостей пристроїв і браузерів.



Цікаво



- За Аароном Густафсоном адаптивність — це особливий підхід до розробки сайта, який дозволяє вже наявним веб-ресурсам підлаштовуватися під розміри екранів різних пристроїв. Інакше говорячи, сторінка повинна автоматично підлаштовуватися під екран, змінювати розмір картинок тощо. Це дозволило усунути потребу в розробці дизайну для кожного нового типу пристрою.



Цікаво

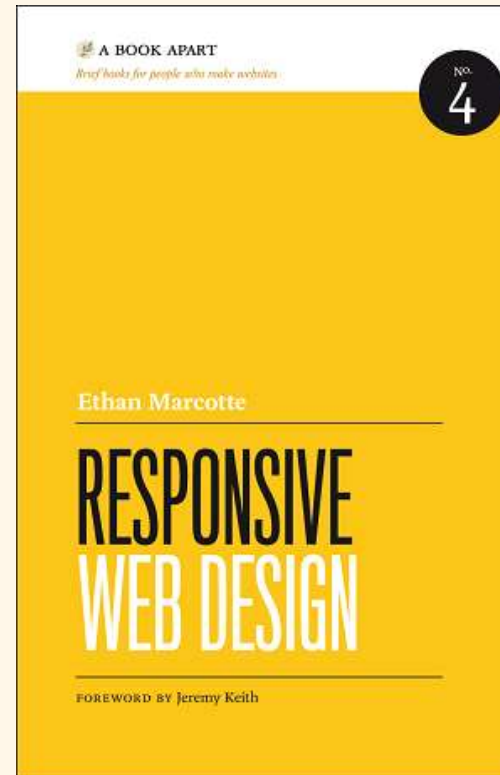
- Традиційно макет сайту спочатку розробляється для десктопної версії, а вже потім адаптується до мобільної. Люк Вроблевські у своїй книзі «**Спочатку мобільні**» запропонував піти від протилежного: розробити макет сайту для мобільної версії, а вже потім покращувати до десктопної. Вроблевські керувався тим, що **ускладнювати просте легше, ніж спрощувати складне.**



Цікаво



Концепцію чуйного дизайну запропонував Ітан Маркотт у 2010 році та описав у своїй книзі «Responsive Web Design». Особливістю такого підходу стала плавна зміна сайта, зорієнтована на вміст, а не на конкретні пристрої.



Основні особливості адаптивного дизайну



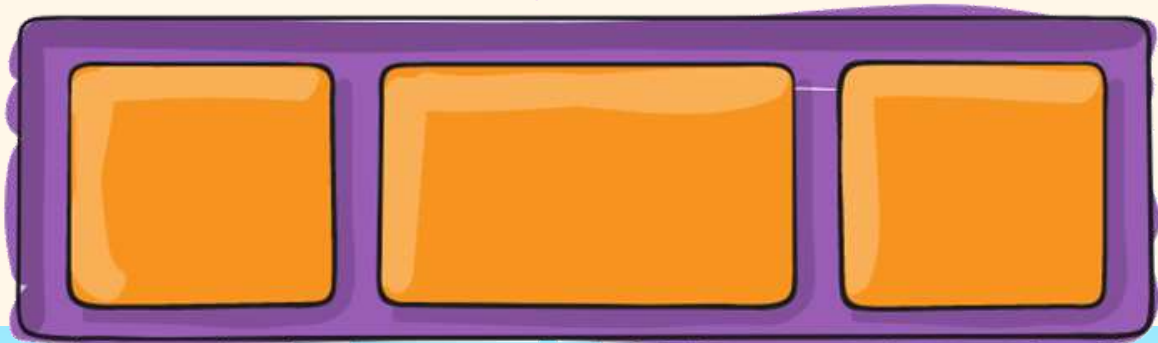
- застосування гнучкого макета на основі сітки (англ. *flexible, grid-based layout*);
- використання гнучких зображень (англ. *flexible images*);
- робота з медіа-запитами (англ. *media queries*);
- плавна перебудова блоків у разі зміни розміру екрана (наприклад, під час повертання планшету)

Модуль Flexbox

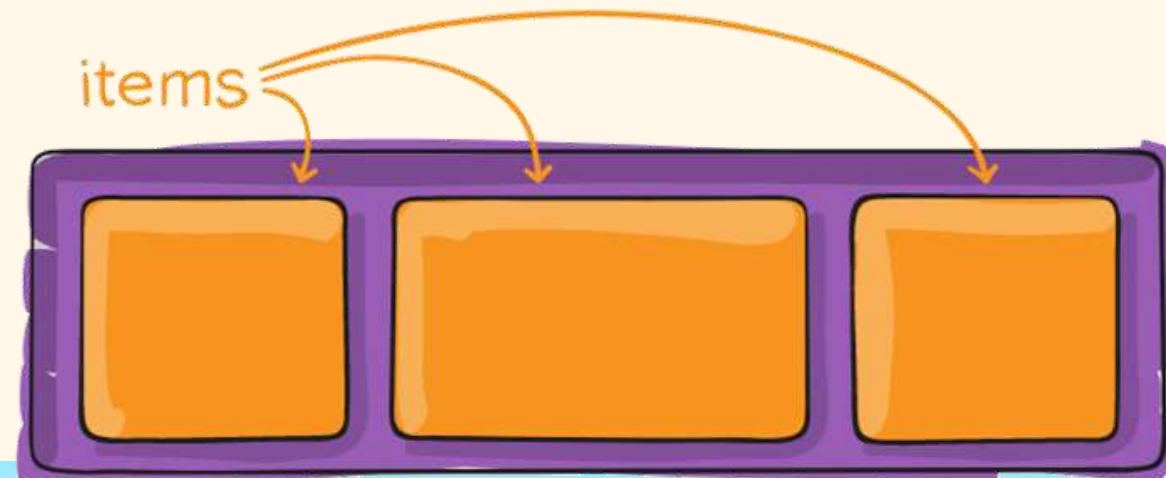


- CSS Flexbox (Flexible Box Layout Module) — модуль макета гнучкого контейнера, що являє собою спосіб компонування елементів. Технологія **flexbox** ставить на меті зробити шари гнучкими, а роботу з ними інтуїтивно зрозумілою. Flexbox складається з гнучкого контейнера (**flex container**) і гнучких елементів (**flex items**). Останні можуть вибудовуватися в рядок або стовпець, а вільний простір розподіляється між ними різними способами

• container



items



Flexbox дозволяє:



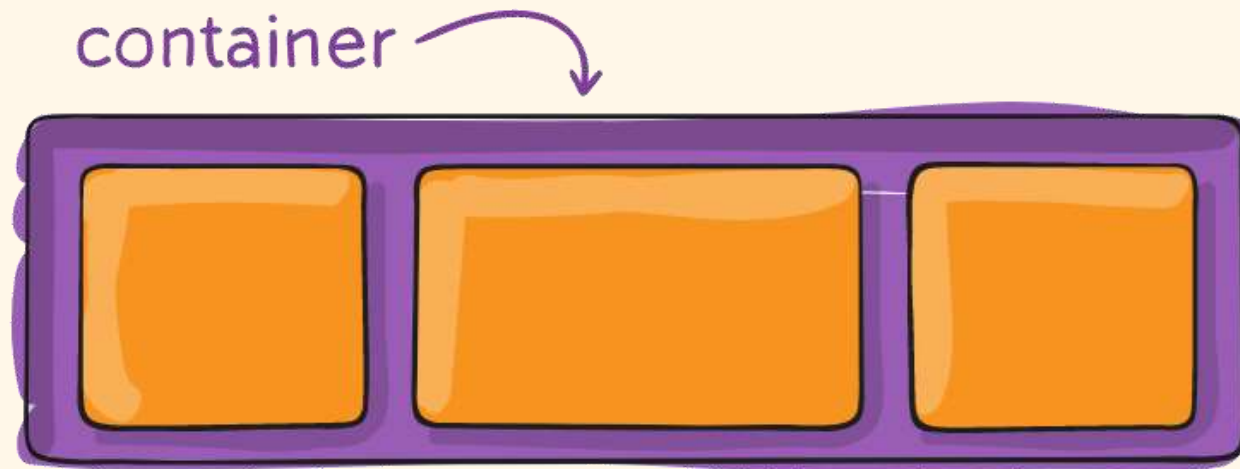
- розташовувати елементи в одному з чотирьох напрямків: зліва направо, справа наліво, зверху вниз або знизу вгору;
 - перевизначати порядок відображення елементів;
 - автоматично визначати розміри елементів так, щоб вони вписувалися в доступний простір;
 - розв'язувати проблему з горизонтальним і вертикальним центруванням;
 - переносити елементи всередині контейнера, не допускаючи його переповнення;
 - створювати колонки однакової висоти.
- керувати елементами в одновимірному просторі
-

Display



Щоб почати працювати з Flexbox, необхідно зробити контейнер flex-контейнером. Робиться це так:

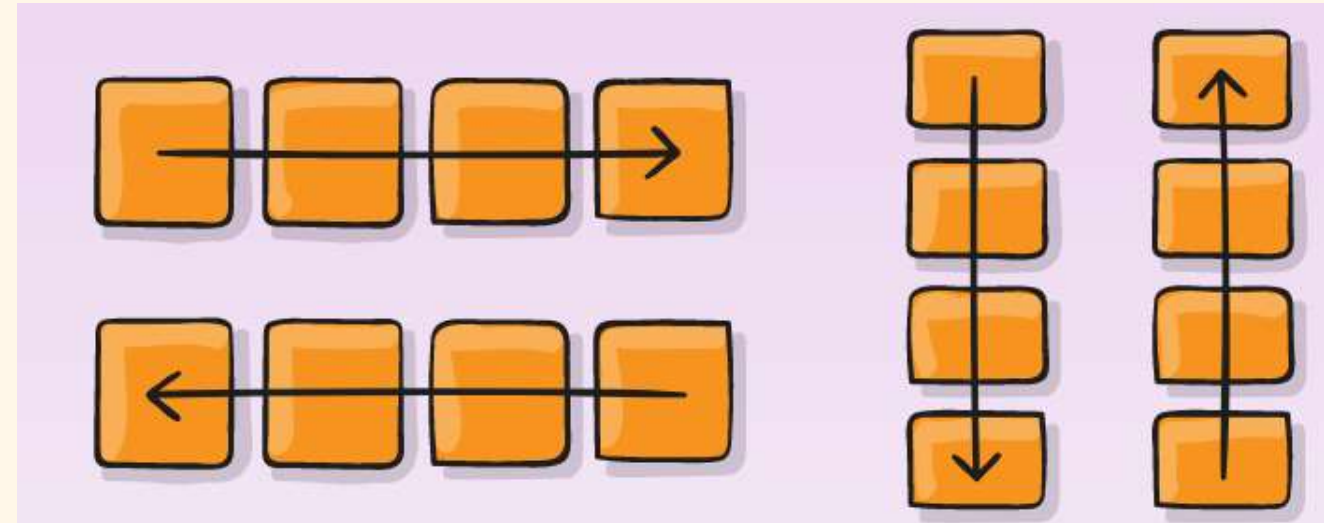
```
.container { display: flex; }
```



Flex-direction – напрямок головної вісі flex-контейнера.



- Головною віссю flex-контейнера є напрям, відповідно до якого розташовуються всі його дочірні елементи.
- Поперечною віссю називається напрям, перпендикулярний головній вісі.
- Головна вісь за замовчуванням розташовується зліва направо. Поперечна - згори вниз.



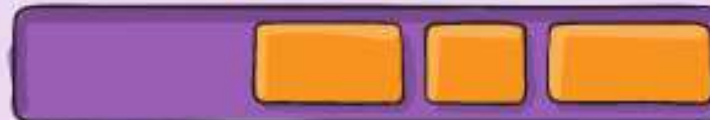
Justify-content

- визначає те, як будуть вирівняні елементи уздовж головної осі.

flex-start



flex-end



center



space-between



space-around



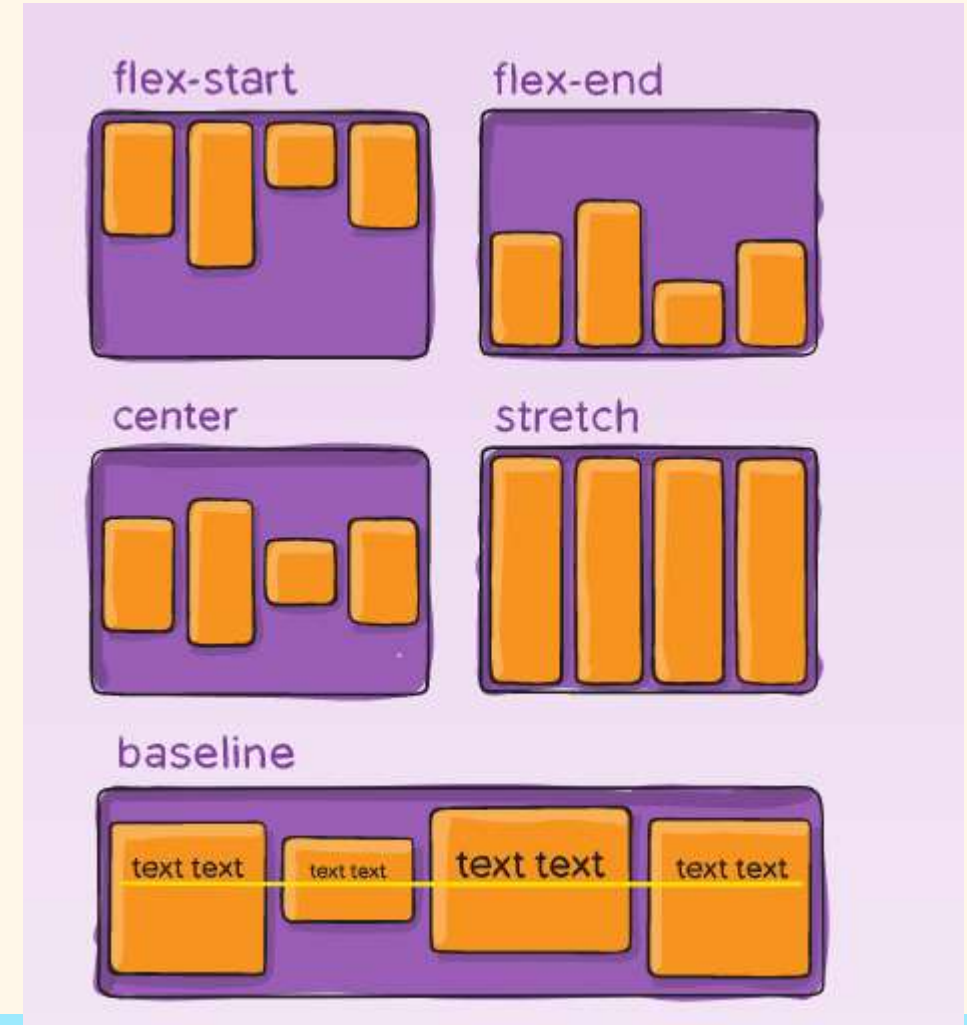
space-evenly



Align-items



- визначає те, як будуть вирівняні елементи уздовж поперечної осі.



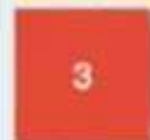


- Щоб зрозуміти, як працює модуль переглянемо наступне відео. У нас є заздалегідь створені 4 різнокольорових div'а різних розмірів, які знаходяться всередині сірого div'а. У кожного div'а є властивість `display: block`. Тому кожен квадрат займає всю ширину рядка.





`display: block;`



- Для більш детального вивчення властивостей модуля **Flexbox** радимо скористатись одним з кращих сайтів для вивчення каскадних таблиць стилів **CSS-TRICKS**



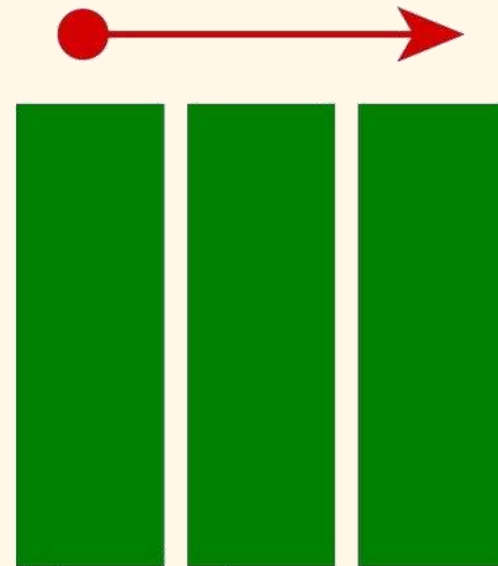
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

A screenshot of the CSS-Tricks website. The top navigation bar is dark with the CSS-Tricks logo (an orange asterisk) and links to Articles, Videos, Almanac, Snippets, and Newsletter. The main content area features a 'Guide Collection' section with the title 'A Complete Guide to Flexbox' in large white text. Below the title, it shows the publication date 'Apr 8, 2013' and the update date 'Sep 4, 2019'. The description reads: 'Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.'

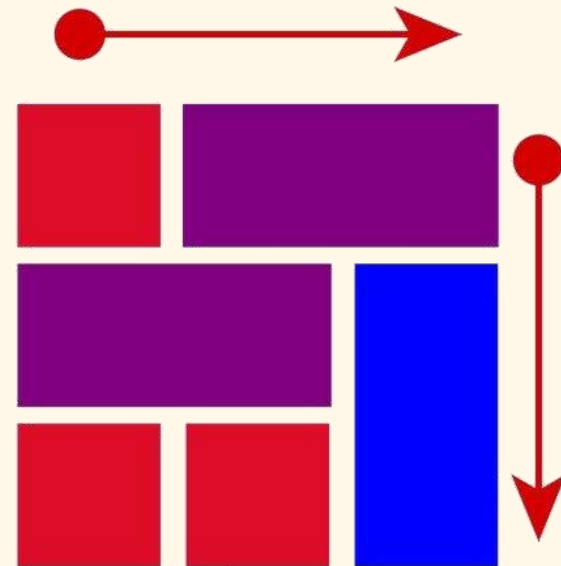
Модуль Grid

Grid дозволяє оперувати стовпцями та рядками.

Grid container (грід-контейнер) — це набір горизонтальних і вертикальних **grid-ліній**, що перетинаються. Ці лінії ділять простір на **grid-області**, де розташовуються **grid-елементи**. У середині grid-контейнера є два набори grid-ліній: один визначає вісь стовпців, інший — вісь рядків.



Flexbox
One Dimensions



CSS Grids
Two Dimensions

Аналізуємо. Обговорюємо

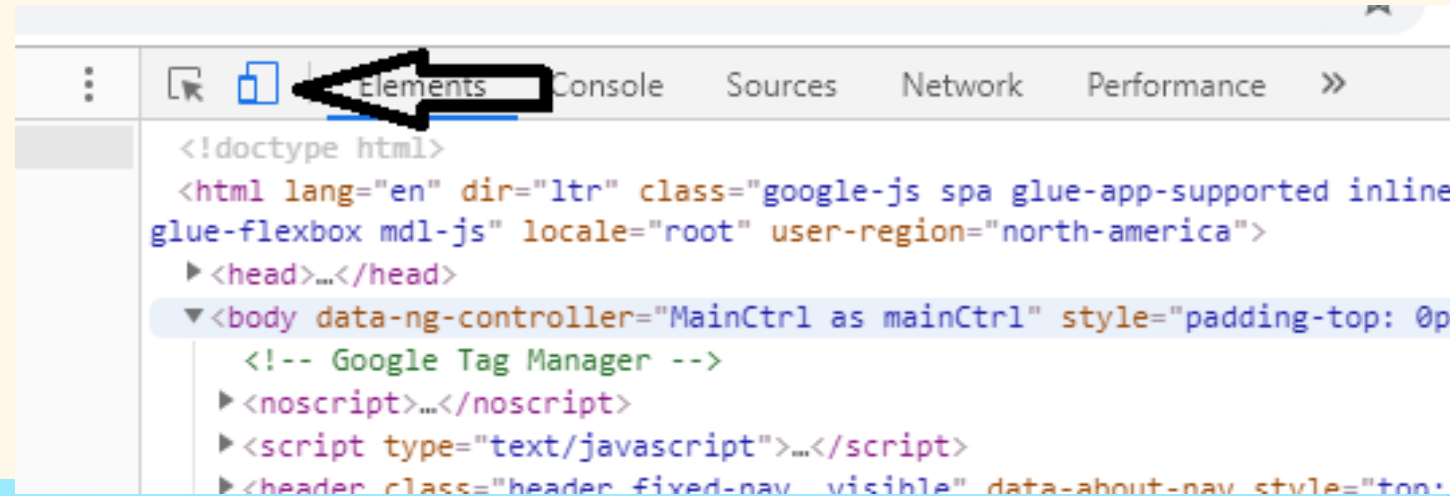


1. Поясніть необхідність розробки адаптивних сайтів.
2. Що таке медіа-запити? Чим обумовлена їх поява?
3. Яка сфера використання медіа-запиту braille?
4. Опишіть принципи адаптивного дизайну.
5. Які інструменти дозволяють розробляти адаптивні сайти?
6. Поміркуйте, чому для адаптивної верстки краще використовувати векторну графіку.

Завдання для самотійного виконання

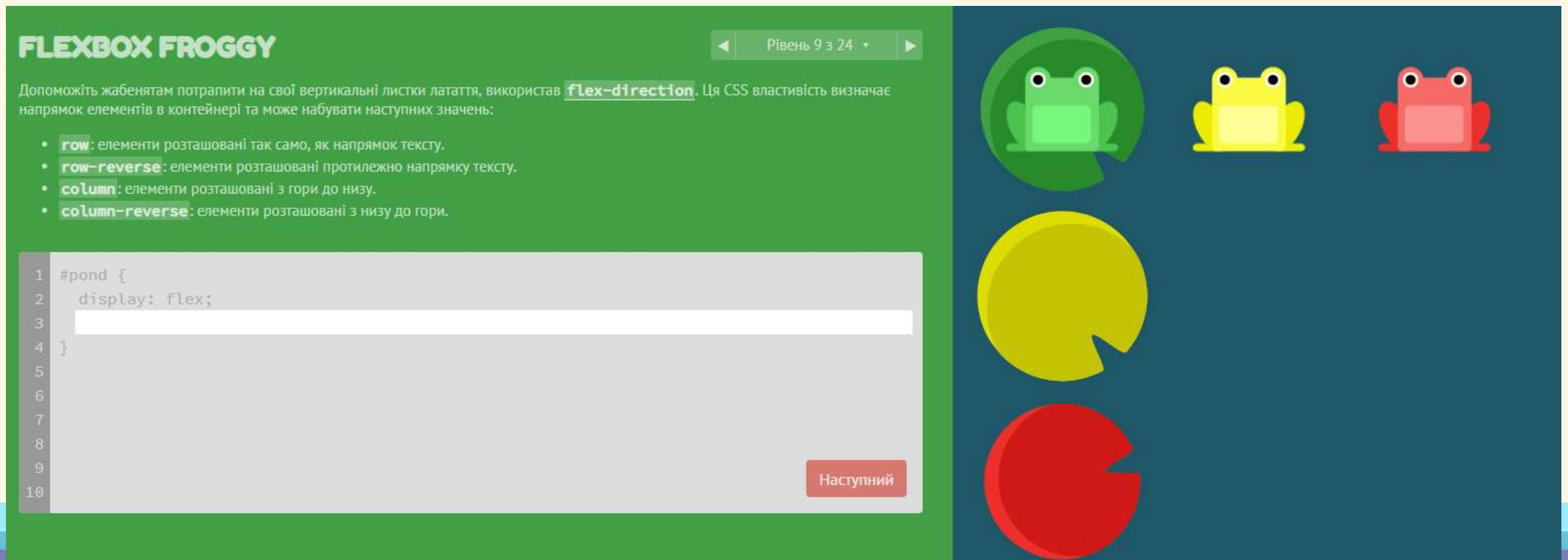
1. Для розуміння роботи адаптивного шаблону відкрийте сторінку Google.
2. Зменшуйте розмір вікна браузера і слідкуйте за тим, як змінюються положення блоків, зникають зображення, як перетворюється меню в результаті зменшення вікна браузера.
3. Опишіть зміни у відображенні сторінки під час застосування режиму інспекції.
4. Перегляньте код. З'ясуйте, який механізм використовують розробники сайту для досягнення адаптивності.

Підказка: для перегляду коду натисніть на кнопку, вказану на рисунку



Домашнє завдання

- Спробуйте свої сили в грі Flexbox Froggy, де потрібно допомагати жабеняті Фрогі та його друзям написанням CSS-коду (<https://flexboxfroggy.com/#uk>). Пройдіть перші шість рівнів. Скріншот результату надішліть на поштову скриньку вчителю.





**Автор презентації:
Речич Наталія
Гімназія №178 м. Києва**