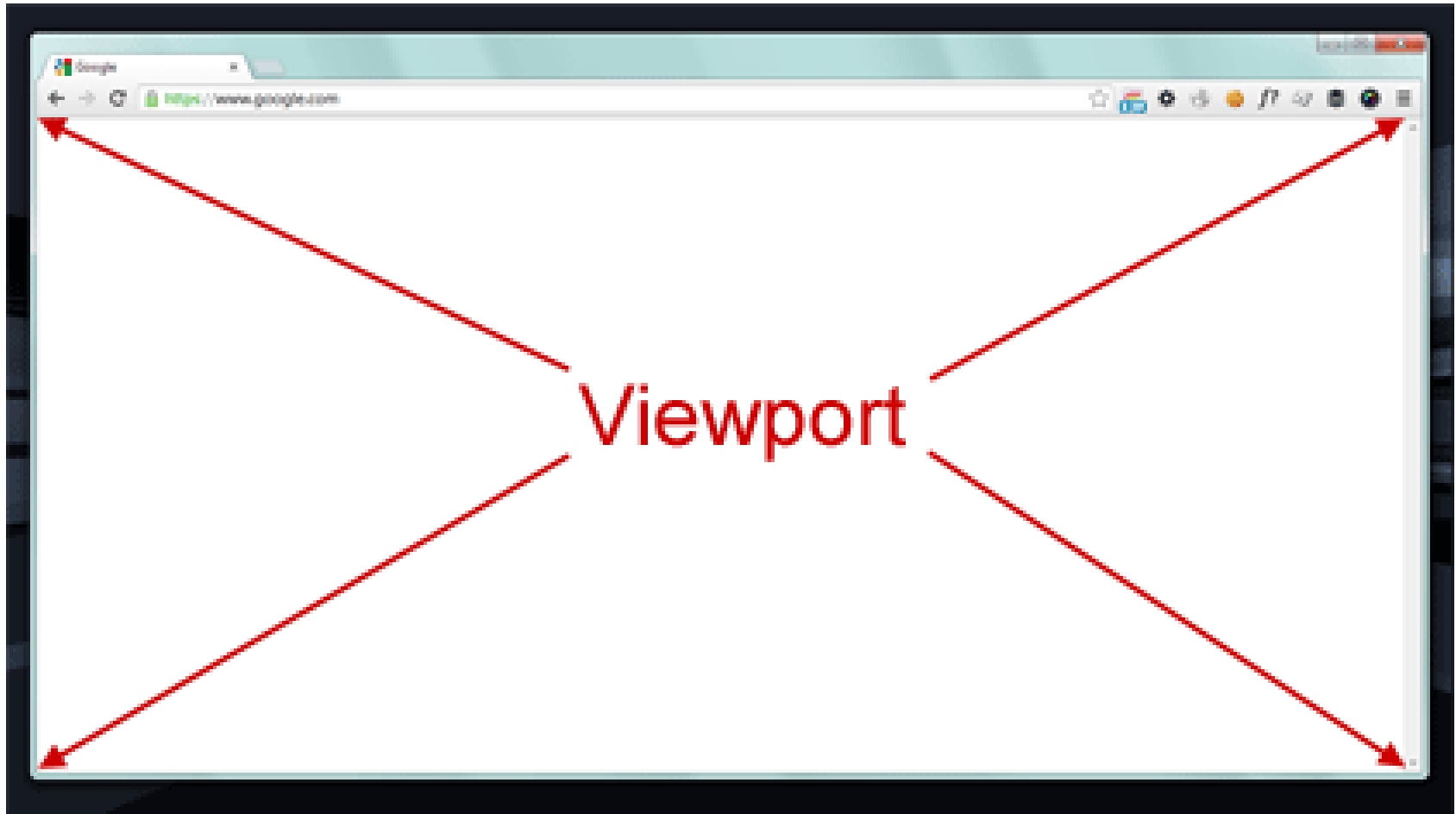


CSS3. viewport. media queries

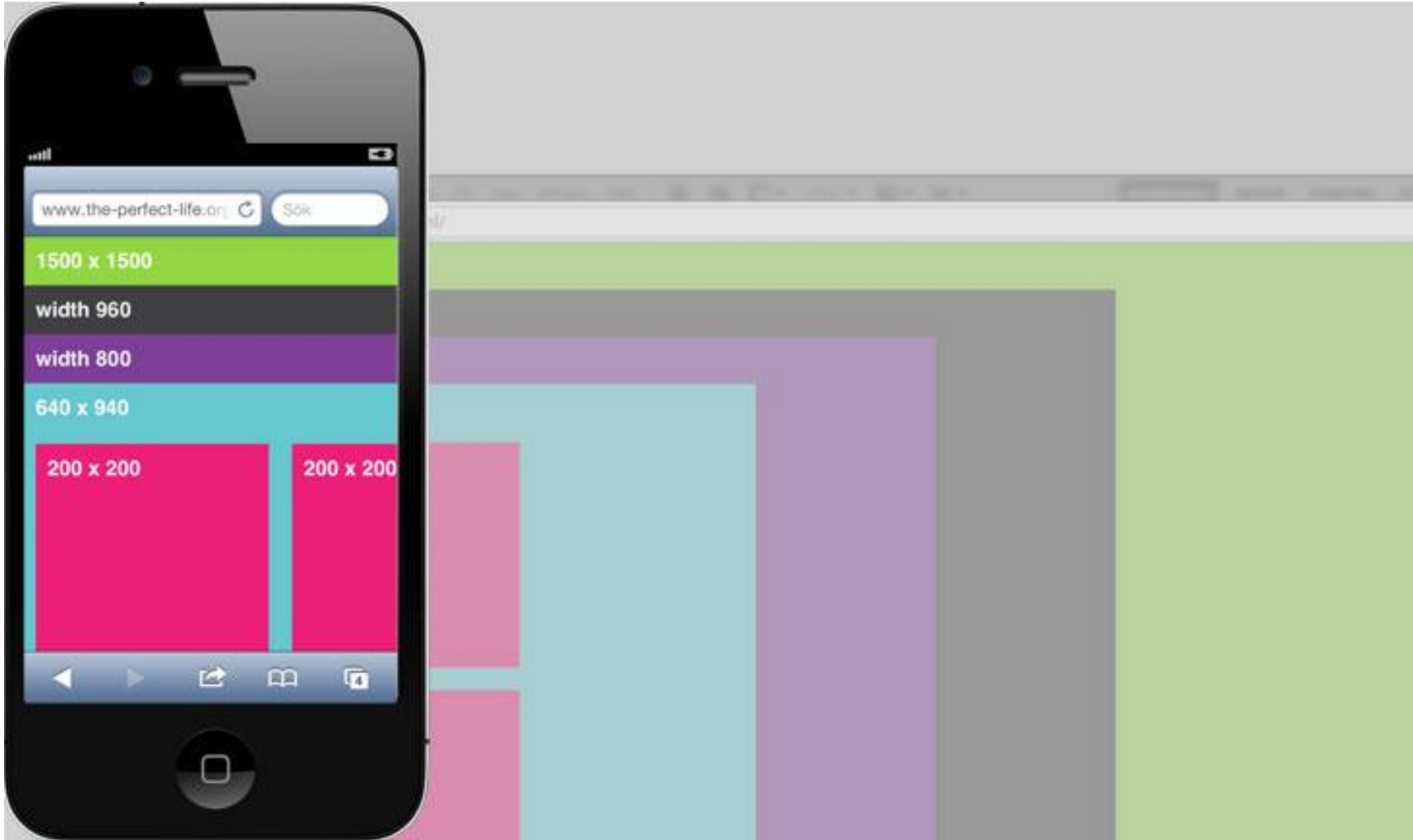
# CSS3. viewport

**Viewport** - це видима користувачу область веб-сторінки. Тобто це те, що може побачити користувач, не вдаючись до прокручування.

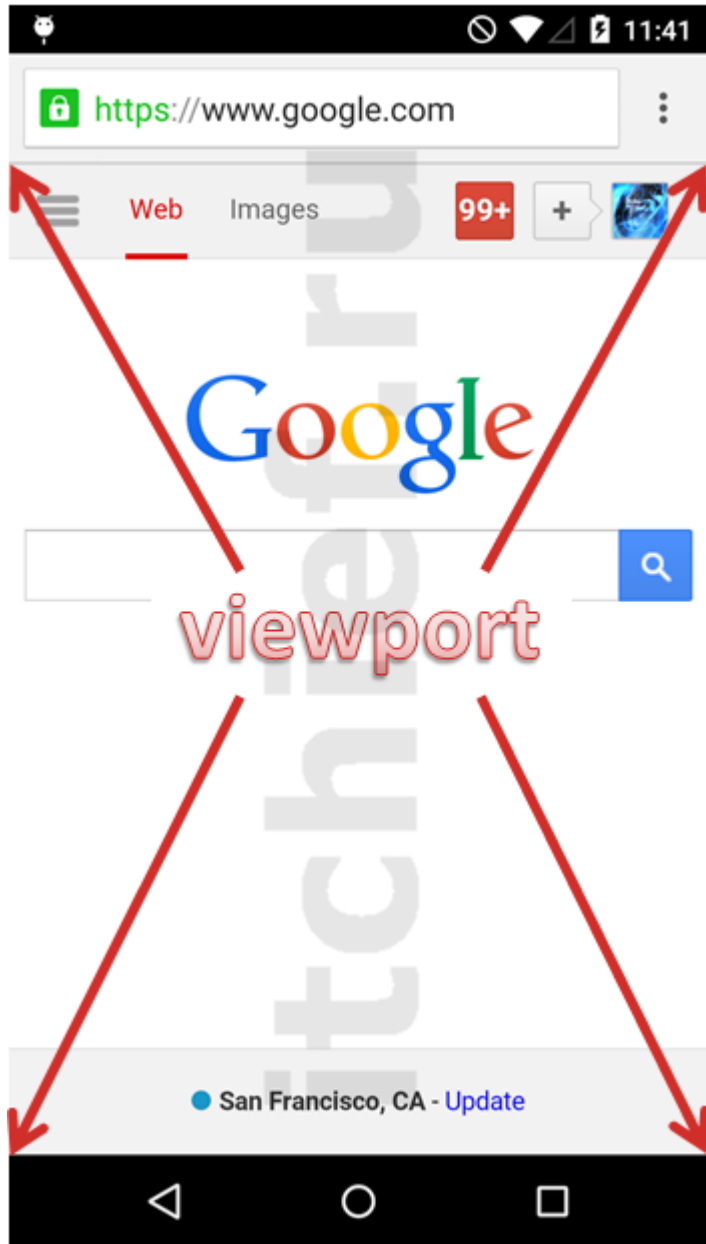


# CSS3. viewport

**Viewport** - це видима користувачу область веб-сторінки. Тобто це те, що може побачити користувач, не вдаючись до прокручування.



# CSS3. viewport



Розміри цієї області визначаються розміром екрану пристрою. Найменшу область перегляду (**viewport**) мають смартфони, розміри екранів яких коливаються від 4 "до 6". А найбільшу - монітори комп'ютерів, розміри діагоналей яких можуть перевищувати 24 ".

До появи смартфонів і планшетів, веб-сторінки в основному переглядались на екранах комп'ютерах. **Viewport** цих екранів хоч і відрізнявся, але не настільки сильно. Для створення сайтів до появи мобільних пристроїв в основному використовувалася фіксована або гумова (гнучка) розмітка.

# CSS3. viewport

Після того, як з'явилися смартфони та планшети, **viewport** одних пристроїв став сильно відрізнятися від **viewport** інших пристроїв. Це призвело до того, що сайти, створені для комп'ютерів, стало неможливо або важко переглядати на смартфонах.

Виходом з цієї ситуації стала поява **адаптивної розмітки**.

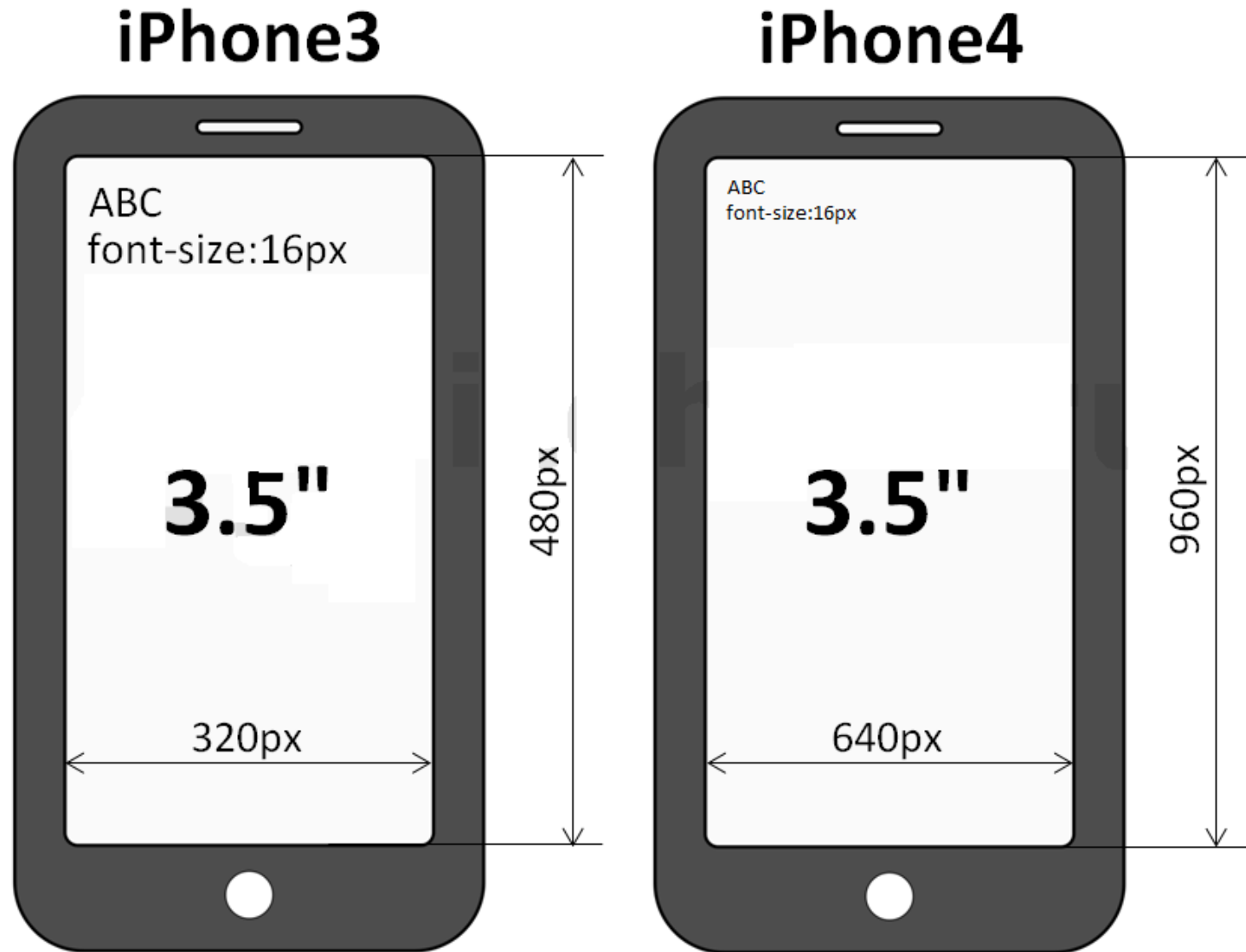
**Адаптивна** - це така **розмітка**, яку можна налаштувати під різні розміри екранів. Здійснюється створення адаптивної розмітки за допомогою **медіа запитів**.

# CSS3. viewport



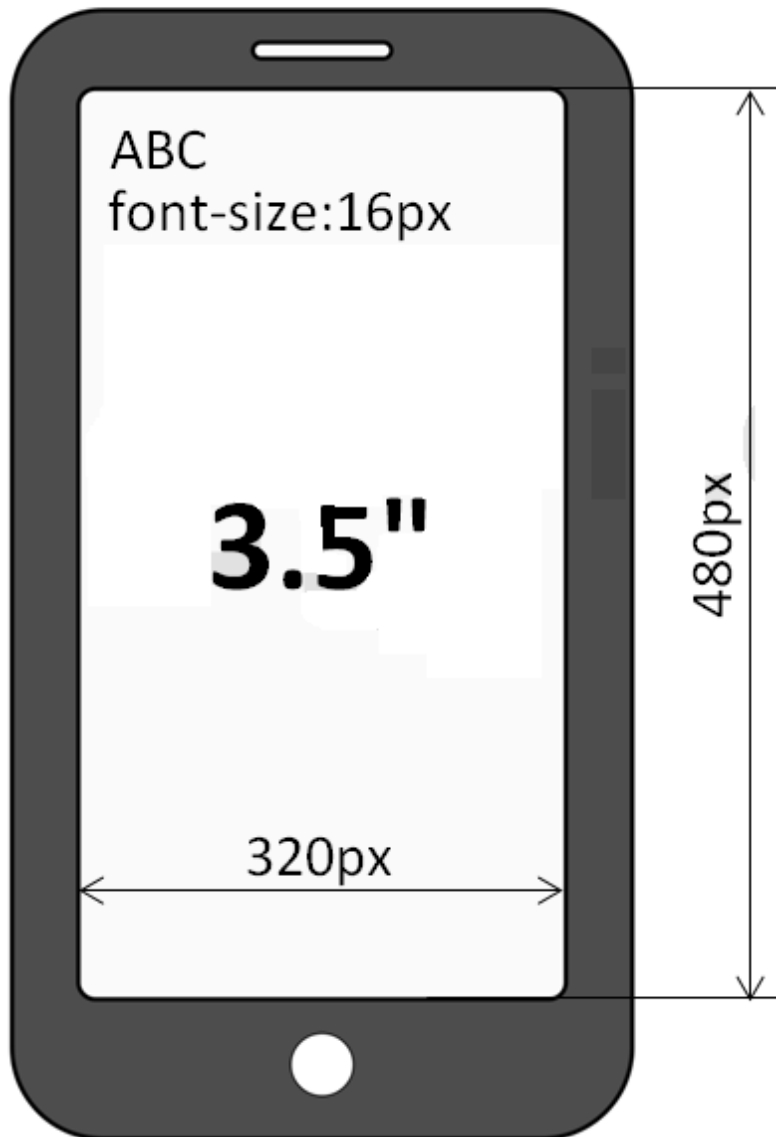
# CSS3. viewport

Але і у адаптивної розмітки з'явилися проблеми після того як з'явилися смартфони з високою щільністю пікселів і, отже з високою роздільною здатністю.



## CSS3. viewport

### iPhone3



Перший пристрій - це смартфон Apple iPhone 3 (діагональ 3.5 ").

Даний телефон не має високу щільність пікселів. У даній моделі вона становить **163ppi** (менше 200ppi).

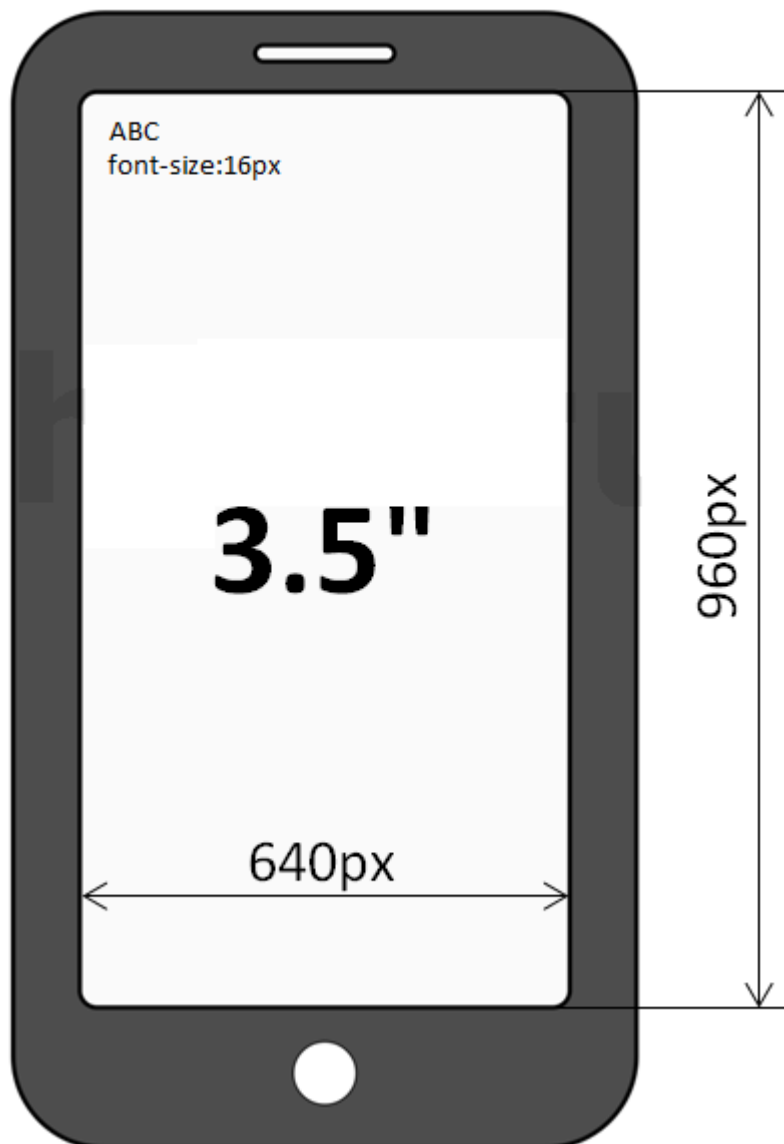
Фізична роздільна здатність даного смартфона становить **320x480**. Така роздільна здатність відповідає діагоналі, якщо порівняти з роздільною здатністю моніторів комп'ютерів.

Тобто на веб-сторінці цього смартфона, текст, виконаний розміром 16px, буде також добре читатись як на моніторі комп'ютера.



## CSS3. viewport

### iPhone4



Другий пристрій - це смартфон Apple iPhone 4. Він має діагональ таку ж як у смартфона Apple iPhone 3, тобто 3.5 ". Але відрізняється від нього тим, що має високу щільність пікселів (326ppi). Отже, більш високу роздільна здатність - 640x960 при тих же розмірах екрану.

Це призведе до того, що той же самий текст і інші об'єкти веб-сторінки будуть виглядати в ньому при тих же умовах в 2 рази меншими. Таким чином, текст буде реально виглядати на 8px.

Щоб зробити цю сторінку придатною для читання, її представлення необхідно збільшити в горизонтальному і вертикальному напрямку в 2 рази (масштабувати).

# CSS3. viewport

## Призначення метатега viewport

Метатег **viewport** був розроблений компанією Apple для того, щоб вказувати браузерам на те, в якому масштабі необхідно відображати користувачеві видиму область веб-сторінки.

Іншими словами **meta viewport** призначений для того, щоб веб-сторінки відображалися (виглядали) правильно (коректно) на смартфонах, планшетах та інших пристроях з високою щільністю пікселів (**> 200ppi**).

Даний метатег призначений в великій мірі для адаптивних сайтів, але за допомогою нього можна поліпшити представлення веб-сторінок, що мають фіксовану або гнучку розмітку.

# CSS3. viewport

Налаштування **meta viewport** для адаптивних веб-сторінок

```
<head>  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
</head>
```

Атрибут **name** призначений для того щоб вказати браузеру, яку саме інформацію про сторінку хочемо йому повідомити. В даному випадку ця інформація стосується **viewport**.

Контент (вміст) цих відомостей вказується в якості значення атрибута **content** за допомогою пар ключ-значення, розділених між собою комами.

Для адаптивного дизайну значення атрибута **content viewport** має визначатися 2 параметрами:

**width = device-width**

**initial-scale = 1**

# CSS3. viewport

## Налаштування **meta viewport** для адаптивних веб-сторінок

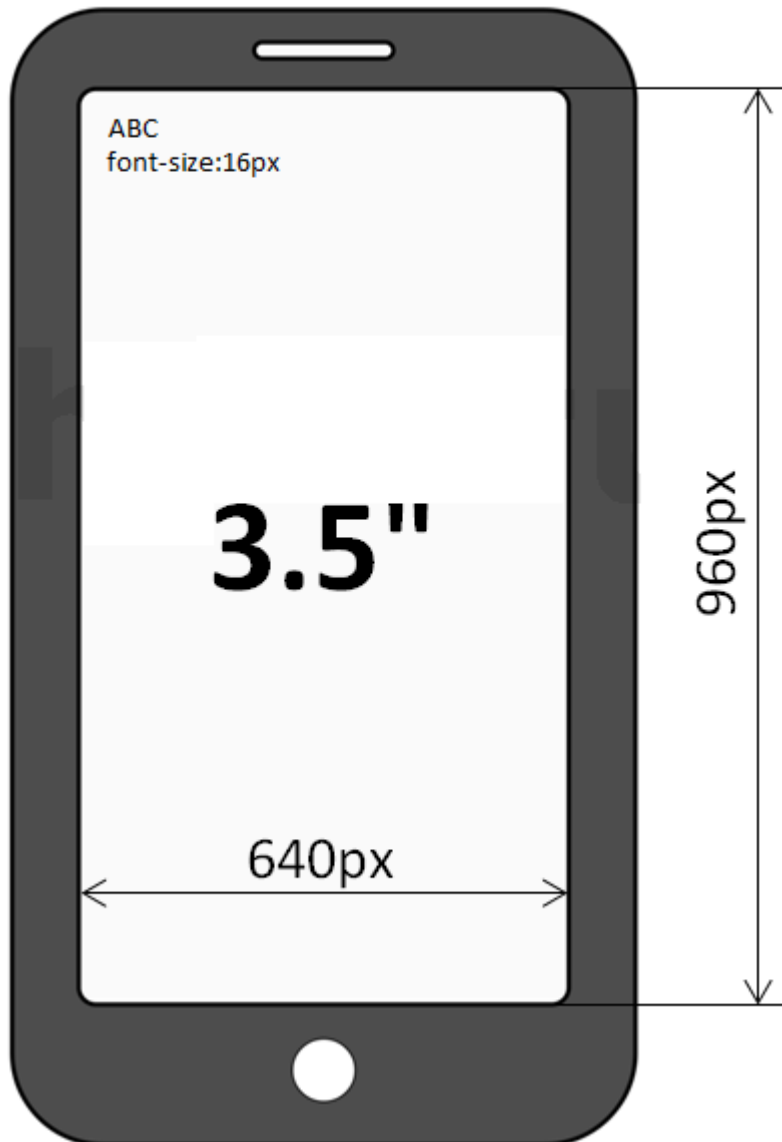
Перший параметр (**width = device-width**) відповідає за те, щоб ширина видимої області веб-сторінки дорівнювала ширині пристрою (**device-width**).

Дана ширина - це не фізична роздільна здатність екрану. Це деяка величина незалежна від роздільної здатності екрану.

Вона призначена для того, щоб мобільний адаптивний дизайн сайту відображався на всіх пристроях однаково, незалежно від щільності пікселів екрану.

## CSS3. viewport

### iPhone4



Наприклад, смартфон iPhone4 з фізичною роздільною здатністю **640x960** має CSS роздільну здатність **320x480**.

Це означає те, що сайт з метатегом **viewport (width = device-width)** на цьому пристрої буде виглядати так ніби це пристрій має розширення **320x480** (в даному випадку замість **device-width** буде підставлятися значення 320px).

Тобто на один CSS піксель буде припадати 4 фізичних пікселя (2 по горизонталі і 2 по вертикалі).

## CSS3. viewport

Другий параметр **initial-scale** - встановлює початковий масштаб веб-сторінки.

Значення 1 означає те, що масштаб дорівнює 100%.

# CSS3. viewport

## meta viewport і не адаптивні сторінки

Якщо сайт не має адаптивний дизайн, то його подання на екрані смартфона теж можна поліпшити.

Наприклад, можна зробити так щоб ширину сторінки масштабувати під ширину пристрою (якщо ширина макета більше CSS ширини). Але врахуйте, що масштабування працює тільки в розумних межах.

Здійснюється це теж за допомогою встановлення параметру width значення device-width.



**<meta name="viewport" content="width=device-width">**

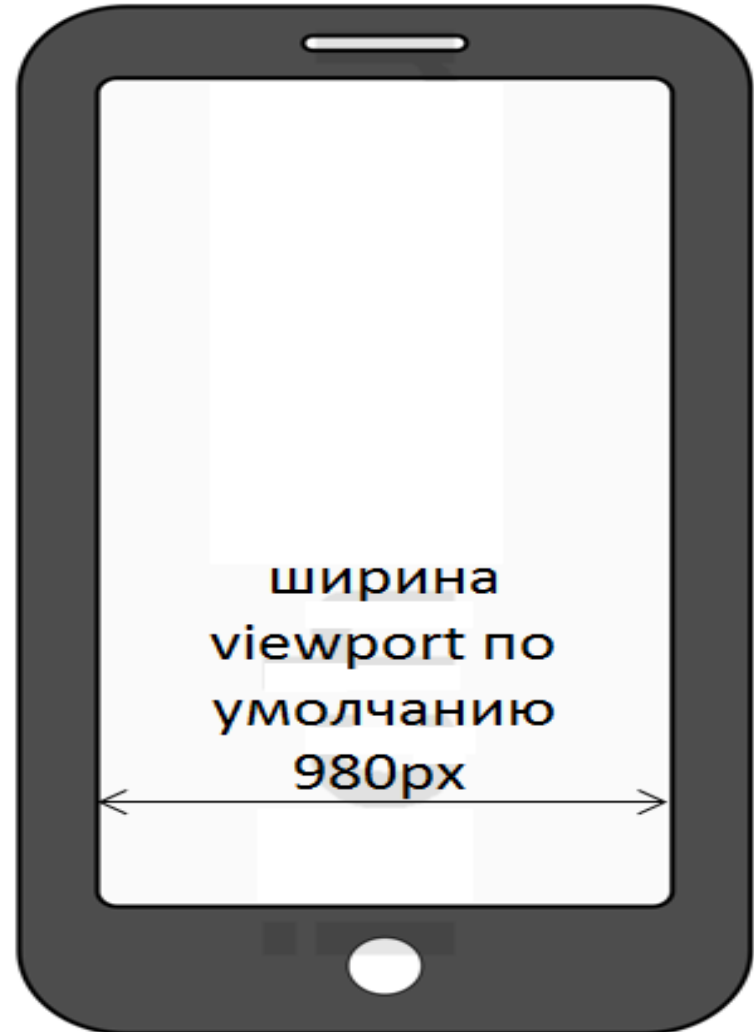
## CSS3. viewport

Крім того, розробники браузерів подбали навіть про тих, кому важко додати цей рядок. У цьому випадку екран пристрою буде за замовчуванням мати CSS ширину, рівну 980px. Це дозволить відобразити без прокрутки (по ширині) більшість десктопних макетів сайтів.

Якщо ж необхідно фіксований сайт відобразити в браузері мобільного пристрою в звичайному масштабі (не зменшення), то необхідно використовувати такий вигляд метатега viewport:

```
<meta name="viewport"  
content="initial-scale=1">
```

без viewport





# CSS3. viewport

## Розширені можливості **meta viewport**

Крім основних параметрів, тег meta viewport містить багато інших.

Ось деякі з них:

**minimal-scale** - задає мінімальний масштаб;

**maximal-scale** - встановлює максимальний масштаб;

**user-scalable** - вказує, чи може користувач управляти масштабом чи ні.

# CSS3. viewport

Приклади **viewport** з використанням додаткових параметрів:

1. `<!-- viewport, без можливості його збільшення користувачем -->`
2. `<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">`
- 3.
4. `<!-- viewport, з можливістю його збільшення користувачем -->`
5. `<meta name="viewport" content="width=device-width, user-scalable=yes">`
- 6.
7. `<!-- viewport, що має ширину 1024 пікселів -->`
8. `<!-- Веб-сторінка, з фіксованим макетом (наприклад, з шириною 1024px або менше) буде відображатися на екрані мобільного перегляду без прокручування -->`
9. `<meta name="viewport" content="width=1024">`

# CSS3. media queries

## CSS3. media queries

З допомогою медіа-запитів можна визначати зовнішній вигляд веб-сторінок, спираючись на ширину і висоту вікна браузера.

Завдяки медіа-запитами ви можете паралельно створювати стилі для невеликих мобільних телефонів, планшетів, лептопів і моніторів з великою діагоналлю.

Ідея адаптивного дизайну - надати відвідувачам сайт в найбільш зручній і привабливій формі.

Дизайн створюється під як мінімум три цільових екрани: настільний ПК, планшет і смартфон.

Але з огляду на велику кількість варіацій цих девайсів, недостатньо зупинитися на якійсь одній ширині для кожного типу пристрою. Тому важливо використовувати гнучку сітку і проводити тестування в процесі розробки, відстежуючи те, як підлаштовується дизайн сайту під різну ширину браузерного вікна.

# CSS3. media queries

Коли використовувати медіа-запити?

Як правило, для адаптації дизайну веб-сторінки необхідно прописати ряд CSS-стилів для певних елементів.

Найбільш часті сценарії, коли вам знадобиться звернутися до медіа-запитами:

**Визначення кількості колонок.** Трьохколоночної макет сайту, який шикарно виглядає на настільних ПК, буде зовсім недоречний для мобільних телефонів і планшетів. За допомогою медіа-запитів можна перевизначити кількість колонок в макеті з урахуванням ширини екрану пристрою, згрупувавши для смартфонів весь контент в одну колонку.

**Відносна ширина.** Коли ви встановлюєте фіксовану ширину макета, скажімо, в 960 пікселів, це підійде для настільних комп'ютерів, лептопів і деяких особливо великих планшетів. Однак для мобільного телефону такий розмір контейнера занадто великий. Використовуючи медіа-запит, можна спеціально для смартфонів створити «гумовий» макет, ширина якого буде не фіксованою, а відносною. Таким чином, контейнер шириною 100% буде легко підлаштовуватися під будь-який смартфон в будь-якій орієнтації.

# CSS3. media queries

Коли використовувати медіа-запити?

**Зменшення відступів.** Великі відстані між блоками і елементами створюють відчуття повітряного, легкого дизайну. Однак відступи, які добре виглядають на моніторі з діагоналлю 21-27 дюймів, будуть погано виглядати на невеликому екрані смартфона, до того ж примушуючи користувачів більше прокручувати сторінку. Медіа-запити дозволяють задати альтернативні розміри відступів для вузьких дисплеїв.

**Регулювання розміру шрифту.** Заголовки висотою в 60 пікселів можуть красиво виглядати на робочому столі, але для мобільних пристроїв такий розмір шрифту рідко коли доречний, тому для вузьких екранів буде доцільним встановити менше значення font-size.

# CSS3. media queries

Коли використовувати медіа-запити?

**Адаптивна навігація.** Нерідко буває так, що навігаційне меню, яке було продумано для десктопної версії сайту, абсолютно не підходить для мобільних пристроїв через свою об'ємність або манери розташування пунктів меню. Існує кілька підходів до реалізації зручної навігації для мобільної версії сайту, і медіа-запити неодмінно приймають в цьому участь.

**Приховування елементів.** Деякі частини веб-сторінки можуть бути марними або малозначущими, якщо сайт проглядається з мобільного телефону. За допомогою властивості **display: none** і медіа-запитів ви можете приховати певні елементи від користувача, якщо він зайшов на сайт зі смартфона. Але майте на увазі, що подібний прийом не економить трафік: приховані елементи як і раніше будуть завантажуватися браузером.

# CSS3. media queries

## Breakpoints (контрольні точки)

За допомогою медіа-запитів ви можете створювати так звані контрольні точки (**Breakpoints**) і прив'язувати до них CSS-стилі. Наприклад, можна визначити для браузера наступні правила:

*«Якщо ширина екрану складає більше ніж 767 пікселів, до веб-сторінки застосовуються ці стилі, а якщо ширина екрана становить більше ніж 991 піксель, застосовуються інші стилі».*

Ось ці числа, що позначають ширину екрану, і називаються контрольними точками.

Яку ширину необхідно вказувати в контрольних точках? Найбільш проста відповідь: ту, на якій верстка веб-сторінки починає ламатися.

Уявімо, що у вас є сайт з контейнером шириною 1180 пікселів. Тоді у вікні шириною 1200 пікселів (враховуємо смугу прокрутки) або менше він, швидше за все, буде виглядати не дуже добре. Ось і ваша перша контрольна точка: необхідно внести правки в дизайн, якщо сайт проглядається в вікні завширшки менше ніж 1200 пікселів.



# CSS3. media queries

## Breakpoints (контрольні точки)

При створенні гнучких сіток часто застосовується певний набір медіа-запитів, передбачений для трьох різних контрольних точок: смартфон, планшет, монітор ПК. Контрольних точок може бути і більше.

Наприклад, в Bootstrap 4 використовується чотири контрольні точки в одному напрямку і ще чотири - в зворотному напрямку:

Від маленьких екранів до великих:

```
@media (min-width: 576px) { ... }  
@media (min-width: 768px) { ... }  
@media (min-width: 992px) { ... }  
@media (min-width: 1200px) { ... }
```

Від великих екранів до маленьких

```
@media (max-width: 1199px) { ... }  
@media (max-width: 991px) { ... }  
@media (max-width: 767px) { ... }  
@media (max-width: 575px) { ... }
```

# CSS3. media queries

## Mobile First vs. Desktop First

Коли ви пишете стилі для контрольних точок, ви не створюєте весь дизайн з нуля, а лише коригуєте існуючу верстку. Виникає питання, під що верстати спочатку: під настільні комп'ютери з подальшим коректуванням під мобільні пристрої або ж навпаки - спочатку створити версію для невеликих екранів, а потім розширити її?

**Desktop First.** При цьому підході ви спочатку повністю верстаєте дизайн для великих екранів, без використання медіа-запитів. Після цього, використовуючи контрольні точки в медіа-запитах, коригуєте дизайн під середні та маленькі екрани (зменшуєте шрифт, перебудовуєте макет, приховуєте другорядні елементи і т.д.). Перевага такої верстки полягає в тому, що старі браузері (наприклад, Internet Explorer 8), які не знають, що таке медіа-запити, зможуть відобразити ваш сайт, оскільки це вихідний дизайн, який ви писали без використання директиви **@media**.

**Mobile First.** Якщо ви вибираєте даний підхід, то спочатку верстаєте дизайн для найменших екранів, не використовуючи медіа-запити. Після цього, створюючи контрольні точки, ви коригуєте верстку під все більш і більш широкі екрани.

# CSS3. media queries

## Mobile First vs. Desktop First

Бажано розміщувати всі медіа-запити під звичайними стилями, а не над ними.

Якщо ви використовуєте контрольні точки, відштовхуючись від мінімальної ширини екрану (**min-width**), розміщуйте їх в порядку **зростання** ширини екрану.

Якщо ви застосовуєте **max-width**, тоді розташуйте медіа-запити в порядку **зменшення** ширини.

В іншому випадку одні медіа-запити можуть «перебити» попередні.

# CSS3. media queries

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, який повертає true або false.

Медіа-запити можуть бути додані наступними способами:

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

@import: **@import url(color.css) screen and (color);**

CSS:

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

# CSS3. media queries

## Логічні оператори

За допомогою логічних операторів можна створювати комбіновані медіазапити, в яких буде перевірятися відповідність декільком умовам.

### оператор **and**

Оператор **and** пов'язує різні умови:

```
@media screen and (max-width: 600px) {  
  /* CSS-стили */;  
}
```

Стилі цього запиту будуть застосовуватися тільки для екранних пристроїв з шириною області перегляду не більше 600px.

Стилі цього запиту будуть застосовуватися для всіх пристроїв при ширині області перегляду від 600px до 800px включно.

```
@media (min-width: 600px) and (max-width: 800px) {  
  /* CSS-стили */;  
}
```

Правило **@media all and (max-width: 600px) {...}** рівнозначно правилу **@media (max-width: 600px) {...}**.

# CSS3. media queries

## Логічні оператори

### оператор **кома**

Оператор **кома** працює за аналогією з логічним оператором **or**.

```
@media screen, projection {  
  /* CSS-стили */;  
}
```

В даному випадку CSS-стили, вкладені у фігурні дужки, спрацюють тільки для екранних або проекційних пристроїв.

# CSS3. media queries

## Логічні оператори

### оператор **not**

Оператор **not** дозволяє спрацювати медіазапиту в протилежному випадку. Ключове слово **not** додається в початок медіазапиту і застосовується до всього запитом цілком, тобто запит

**@media not all and (monochrome) {...}**

буде еквівалентний запиту:

**@media not (all and (monochrome)) {...}**

Якщо медіазапит складений з використанням оператора **кома**, то заперечення буде поширюватися тільки на ту частину, яка йде до коми, тобто запит

**@media not screen and (color), print and (color)**

буде еквівалентний запиту

**@media (not (screen and (color))), print and (color)**

# CSS3. media queries

## Логічні оператори

### оператор **only**

Оператор **only** використовується, щоб приховати стилі від старих браузерів (синтаксис медіа-запитів CSS2).

**media = "only screen and (min-width: 401px) and (max-width: 600px)"**

Ці браузери чекають список медіа-типів, розділених комами. І, таким чином, щоб вони повинні відсікати кожне значення безпосередньо перед першим неалфавітних-цифровим символом, який не є дефісом. Таким чином, старий браузер повинен інтерпретувати попередній приклад як media = "only".

Оскільки даного типу медіа-типу не існує, то і таблиці стилів будуть ігноруватися.



# CSS3. media queries

## Технічні характеристики носіїв

До характеристик медіаносія відносяться параметри пристрою, що перевіряються. Значення, які використовуються при заданні характеристик, є контрольними точками.

**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 більша значення, зазначеного в запиті, **max-width** - ширина області перегляду менша 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 більша значення, зазначеного в запиті, **max-height** - висота області перегляду якого менша значення, зазначеного в запиті.

# CSS3. media queries

## Технічні характеристики носіїв

**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 розширення екрану (кількість пікселів). Значення також можуть перевіряти кількість точок на дюйм (**dpi**) або кількість точок на сантиметр (**dpcm**), наприклад, (**resolution: 300dpi**).

**min-resolution** перевіряє мінімальне розширення екрана, **max-resolution** - максимальне.

# CSS3. media queries

## Технічні характеристики носіїв

**color** Перевіряє кількість біт на кожен з колірних 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** - максимальну кількість біт.

**color-index** Перевіряє кількість записів в таблиці підстановки кольорів. Як значення вказується позитивне число, наприклад, (**color-index: 256**).

**min-color-index** перевіряє мінімальну кількість записів, **max-color-index** - максимальну кількість записів.

**monochrome** Перевіряє кількість біт на піксель монохромного пристрою. Значення задається цілим позитивним числом, наприклад, (**min-monochrome: 8**). **min-monochrome** перевіряє мінімальну кількість бітів, **max-monochrome** - максимальну кількість біт.

**-webkit-device-pixel-ratio** Задає кількість фізичних пікселів пристрою на кожен CSS-піксель.

## CSS3. media queries

### Технічні характеристики носіїв

**device-width, device-height, device-aspect-ratio** є застарілими API, вони видалені з Media Queries Level 4.

## CSS3. media queries

При складанні медіазапитів потрібно орієнтуватися на так звані переломні (контрольні) точки дизайну, тобто такі значення ширини області перегляду, в яких дизайн сайту істотно змінюється, наприклад, з'являється горизонтальна смуга прокрутки. Щоб визначити ці точки, потрібно відкрити сайт в браузері і поступово зменшувати область перегляду.

```
/* Smartphones (вертикальна і горизонтальна орієнтація) ----- */
@media only screen and (min-width : 320px) and (max-width : 480px) {
    /* стилі */
}
```

```
/* Smartphones (горизонтальна) ----- */
@media only screen and (min-width: 321px) {
    /* стилі */
}
```

```
/* Smartphones (вертикальна) ----- */
@media only screen and (max-width: 320px) { /* стилі */ }
```

```
/* iPads (вертикальна і горизонтальна ) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) { /* стилі */ }
```

## CSS3. media queries

```
/* iPads (горизонтальна) ----- */
```

```
@media only screen and (min-width: 768px) and (max-width: 1024px) and  
(orientation: landscape) { /* стилі */ }
```

```
/* iPads (вертикальна) ----- */
```

```
@media only screen and (min-width: 768px) and (max-width: 1024px) and  
(orientation: portrait) { /* стилі */ }
```

```
/* iPad 3*/
```

```
@media only screen and (min-width: 768px) and (max-width: 1024px) and  
(orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) { /* стилі */ }
```

```
@media only screen and (min-width: 768px) and (max-width: 1024px) and  
(orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) { /* стилі */ }
```

## CSS3. media queries

```
/* ПК і ноутбуки ----- */
```

```
@media only screen and (min-width: 1224px) { /* стилі */ }
```

```
/* Великі екрани----- */
```

```
@media only screen and (min-width: 1824px) { /* стилі */ }
```

```
/* iPhone 4 ----- */
```

```
@media only screen and (min-width: 320px) and (max-width: 480px) and  
(orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) { /* стилі */ }
```

```
@media only screen and (min-width: 320px) and (max-width: 480px) and  
(orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) { /* стилі */ }
```

```
/* iPhone 5 ----- */
```

```
@media only screen and (min-width: 320px) and (max-height: 568px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
@media only screen and (min-width: 320px) and (max-height: 568px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

## CSS3. media queries

```
/* iPhone 6 ----- */
```

```
@media only screen and (min-width: 375px) and (max-height: 667px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
@media only screen and (min-width: 375px) and (max-height: 667px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
/* iPhone 6+ ----- */
```

```
@media only screen and (min-width: 414px) and (max-height: 736px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
@media only screen and (min-width: 414px) and (max-height: 736px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```



## CSS3. media queries

```
/* Samsung Galaxy S3 ----- */
```

```
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){ /* стилі */ }
```

```
/* Samsung Galaxy S4 ----- */
```

```
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 3){ /* стилі */ }
```

```
@media only screen and (min-width: 320px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 3){ /* стилі */ }
```

```
/* Samsung Galaxy S5 ----- */
```

```
@media only screen and (min-width: 360px) and (max-height: 640px) and  
(orientation: landscape) and (-webkit-device-pixel-ratio: 3){ /* стилі */ }
```

```
@media only screen and (min-width: 360px) and (max-height: 640px) and  
(orientation: portrait) and (-webkit-device-pixel-ratio: 3){ /* стилі */ }
```

## CSS3. media queries

Щоб перевірити, як виглядає ваш сайт на різних пристроях, можна скористатися сервісом

[Responsive design testing for the masses.](#)

# CSS3. media queries

[https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=ru#\\_1](https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=ru#_1)

<https://webformyself.com/media-zaprosy-obzor-razlichnyx-vozmozhnostej/>

<http://frontender.com.ua/media-queries/>