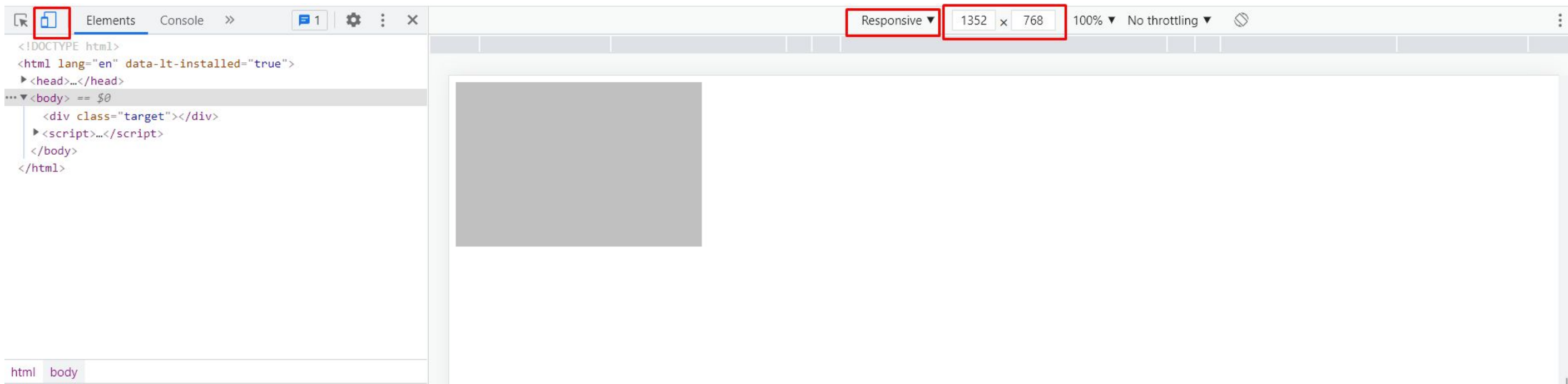




@media screen

Правило `@media screen` дозволяє визначити, які стилі будуть застосовуватись при тому чи іншому розширенні екрану. Для конкретизації правила можна задати характеристики `min-width` та `max-width`. Звісно можливо задати інші характеристики, але саме наведені вище являються базовими.

owwu

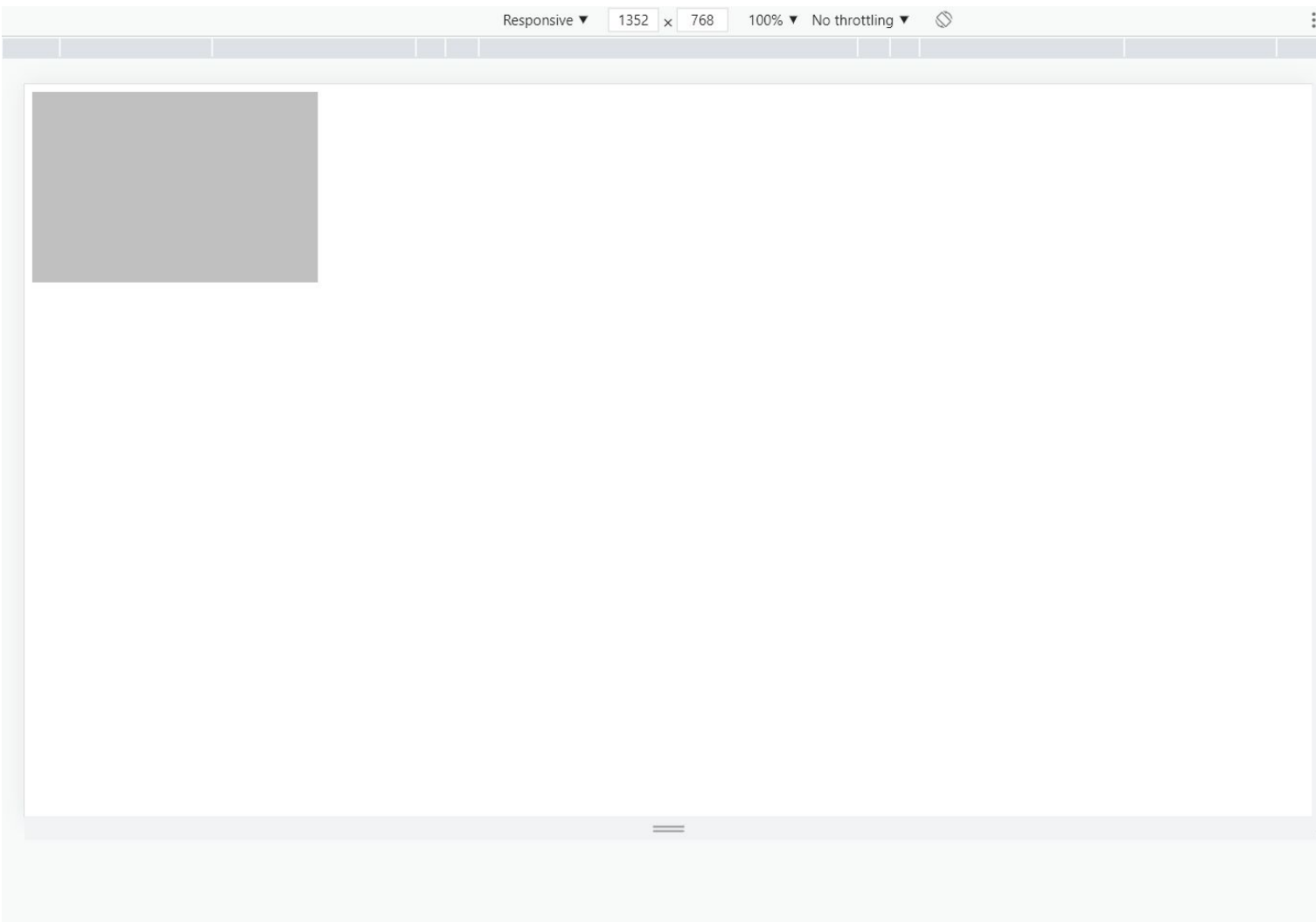


В браузері , в режимі розробника можна обрати потрібне вам розширення для того, щоб тестувати поведінку блоків.

В поточному прикладі width=1352px, height=768px. Блок має наступні стилі :

```
.target {  
  width: 300px;  
  height: 200px;  
  background: silver;  
}
```

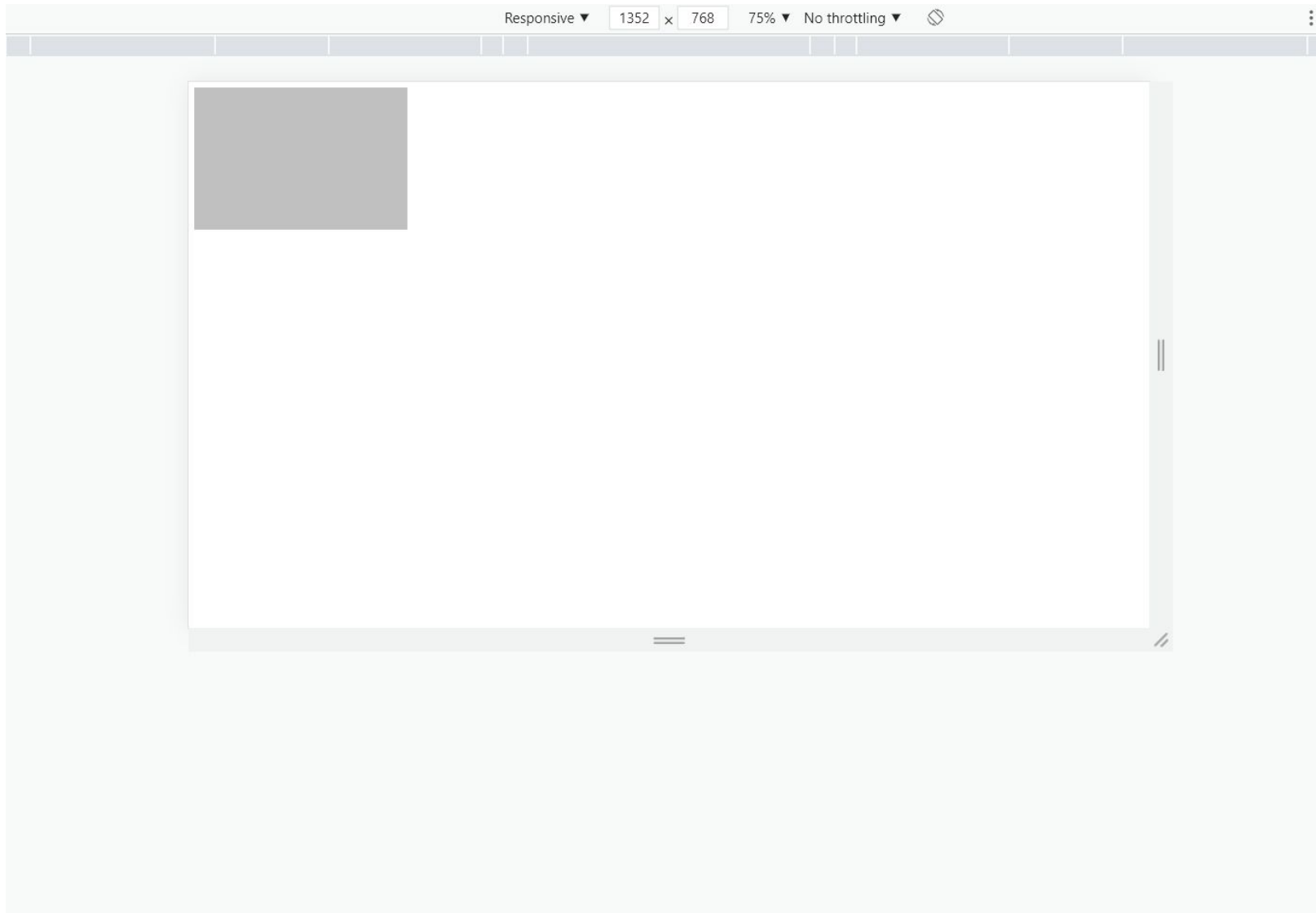
Задамо зміну коліру фону при розширенні менше ніж 1300px



```
.target {  
  width: 300px;  
  height: 200px;  
  background: silver;  
}  
  
@media screen and (max-width: 1300px) {  
  .target {  
    background: cornflowerblue;  
  }  
}
```

В прикладі видно, що всередині правила `@media screen` є звернення до елемента `.target`, та перевизначення тільки його характеристики `background`. Всі інші характеристики (висота та ширина) залишились без змін. Отже якщо ми не перевизначаємо характеристику примусово, вона залишається без змін.

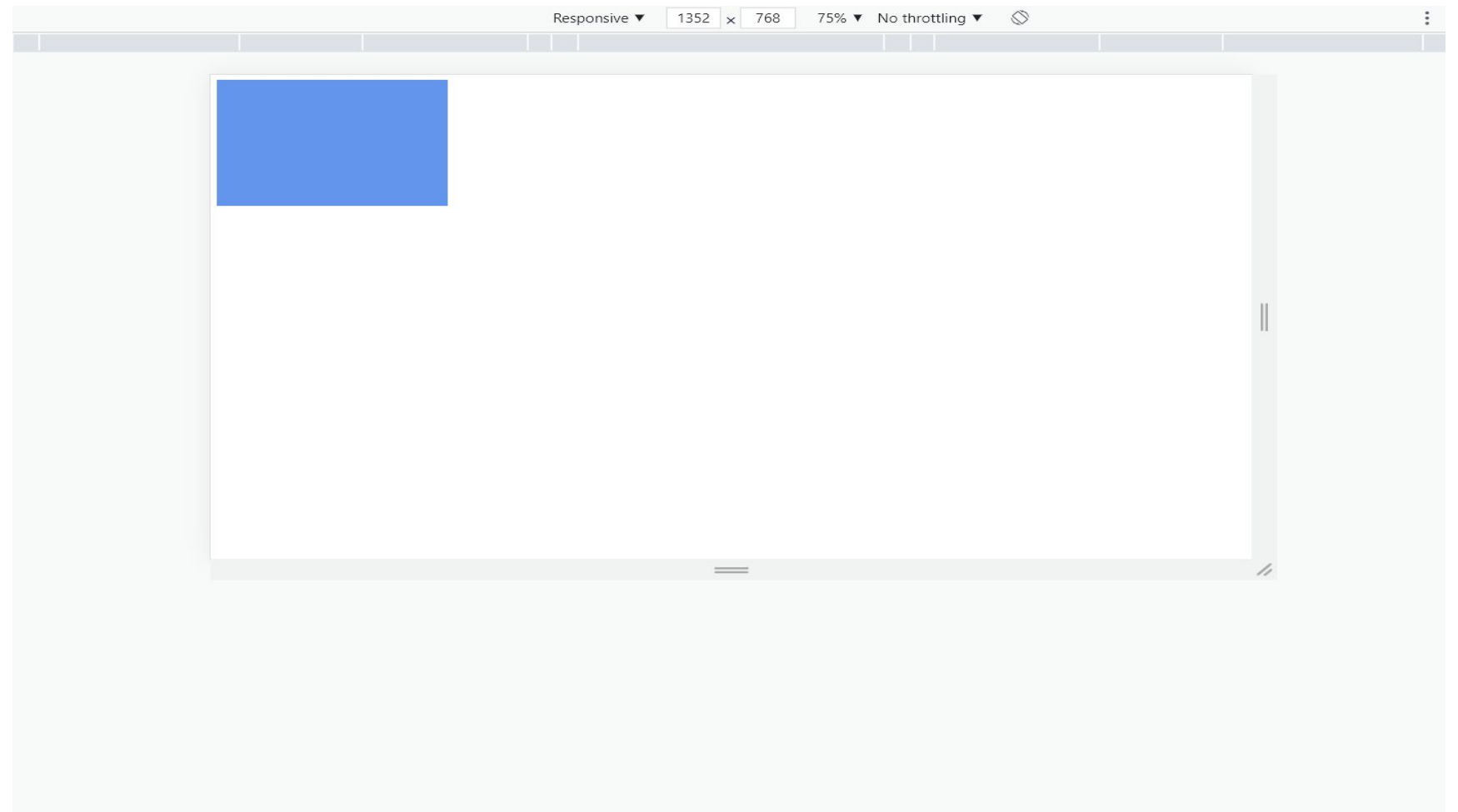
Ми також спроможні визначати зміни у випадку якщо розширення екрану збільшується, або знаходиться в певному діапазоні



Зміна коліру блоку у випадку, якщо ширина більше ніж 1400px

```
.target {  
  width: 300px;  
  height: 200px;  
  background: silver;  
}  
  
@media screen and (min-width: 1400px) {  
  .target {  
    background: cornflowerblue;  
  }  
}
```

Ми також спроможні визначати зміни у випадку якщо розширення екрану збільшується, або знаходиться в певному діапазоні



```
.target {  
  width: 300px;  
  height: 200px;  
  background: silver;  
}
```

```
@media screen and (min-width: 1300px) and (max-width: 1400px) {  
  .target {  
    background: cornflowerblue;  
  }  
}
```



Коли використовувати?

Адаптивна верстка використовується коли потрібно змінювати орієнтацію екрану, або відобразити сайт на мобільному пристрої, або планшеті. Таким чином при різних розширеннях екрану ми не мусимо пропорційно стискати елементи (адже вони ставатимуть дуже маленькими), а можемо їх переносити, видаляти, зменшувати, іншими словами змінювати так, як буде краще, щоб донести користувачу всю інформацію.

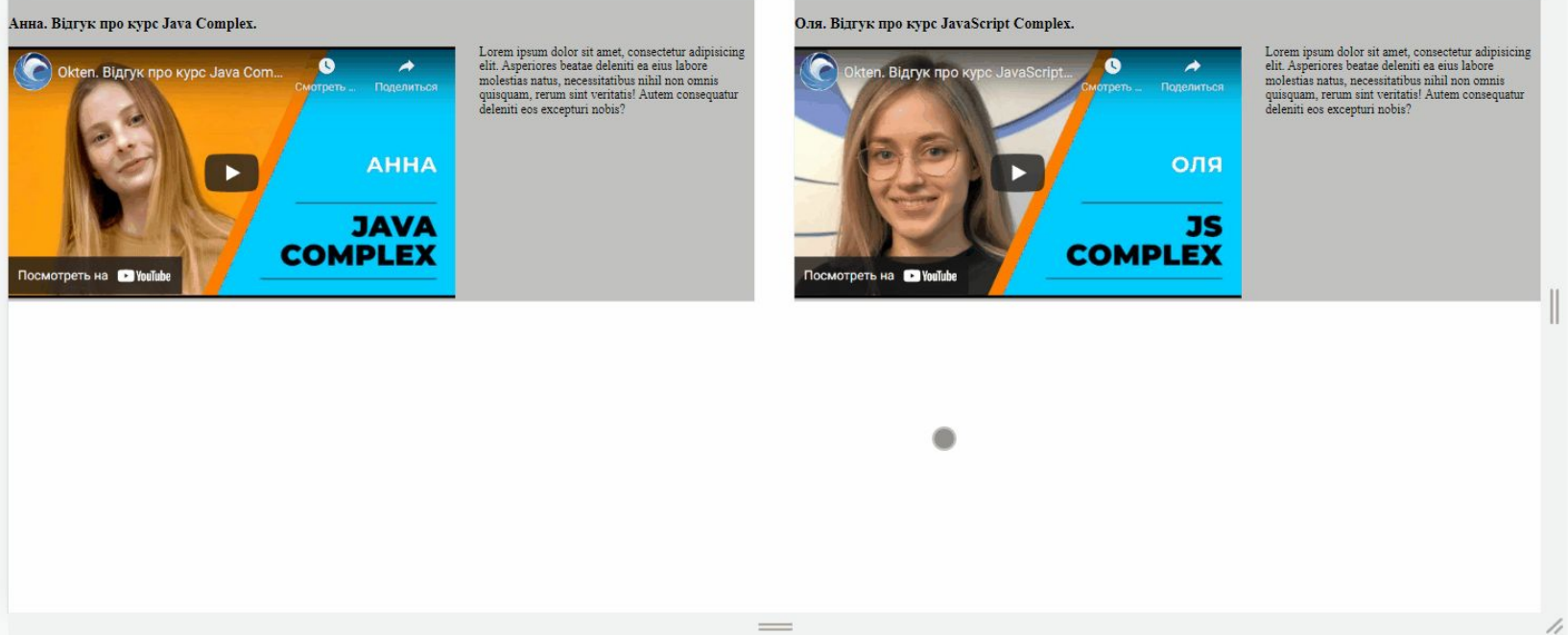
Приклад. Є сайт на якому є декілька відео блоків з описом праворуч. Подібна концепція не буде "найкращим вибором" для мобільного телефону. Тому ми мусимо перенести опис під відео.

OWWU

Ми також спроможні визначати зміни у випадку якщо розширення екрану збільшується, або знаходиться в певному діапазоні

```
body {  
  margin: 0;  
}  
  
.videos {  
  display: flex;  
  justify-content: space-evenly;  
  column-gap: 50px;  
}  
  
.video {  
  display: flex;  
  column-gap: 30px;  
  background: silver;  
}  
  
.video > p {  
  padding-top: 40px;  
}
```

```
<div class="videos">  
  <div class="video">  
    <div class="video_wrap">  
      <h3>  
        Анна. Відгук про курс Java Complex.  
      </h3>  
      <iframe width="560" height="315" src="https://www.youtube.com/embed/SUUKIvxkOIE"  
        title="YouTube video player"  
        frameborder="0"  
        allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;  
picture-in-picture"  
        allowfullscreen></iframe>  
    </div>  
    <p>  
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores beatae deleniti  
      ea eius labore  
      molestias natus, necessitatibus nihil non omnis quisquam, rerum sint veritatis!  
      Autem consequatur deleniti  
      eos excepturi nobis?  
    </p>  
  </div>  
  <div class="video">  
    <div class="video_wrap">  
      <h3>  
        Оля. Відгук про курс JavaScript Complex.  
      </h3>  
      <iframe width="560" height="315" src="https://www.youtube.com/embed/Q7MBL15vY34"  
        title="YouTube video player" frameborder="0"  
        allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;  
picture-in-picture"  
        allowfullscreen></iframe>  
    </div>  
    <p>  
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores beatae deleniti  
      ea eius labore  
      molestias natus, necessitatibus nihil non omnis quisquam, rerum sint veritatis!  
      Autem consequatur deleniti  
      eos excepturi nobis?  
    </p>  
  </div>  
</div>
```



```
@media screen and (max-width: 992px) {  
  .videos {  
    flex-direction: column;  
    row-gap: 30px;  
  }  
  
  .video {  
    flex-direction: column;  
  }  
  
  .video__wrap {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
  }  
}
```

При цьому html розмітка залишилась без змін. Всі маніпуляції відбувались тільки в середині css правил та з використанням @media screen.



VH, VW

Viewport height (vh), Viewport width (vw) - це процентна величина від загальної висоти області перегляду (viewport). $10vh$ становить 10% від поточної висоти області перегляду.

Таким чином, щоб задати окремому блоку розмір висоти, котрий дорівнює висоті області перегляду (або на весь екран) достатньо вказати `height:100vh`

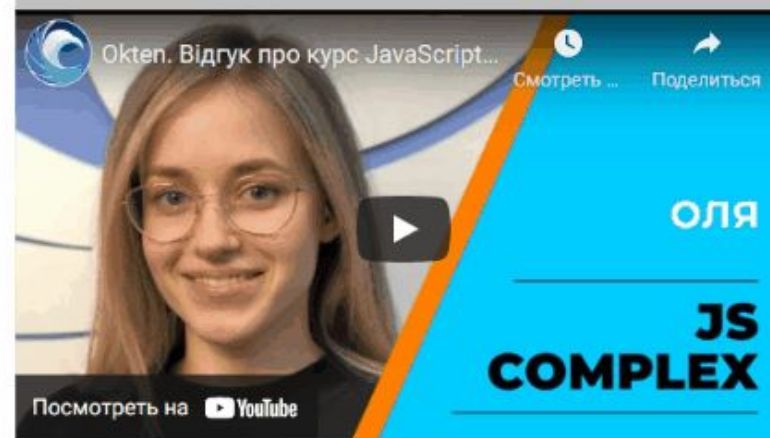
OWWU

Анна. Відгук про курс Java Complex.



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores beatae deleniti ea eius labore molestias natus, necessitatibus nihil non omnis quisquam, rerum sint veritatis! Autem consequatur deleniti eos excepturi nobis?

Оля. Відгук про курс JavaScript Complex.



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores beatae deleniti ea eius labore molestias natus, necessitatibus nihil non omnis quisquam, rerum sint veritatis! Autem consequatur deleniti eos excepturi nobis?

```
@media screen and (max-width: 992px) {  
  .videos {  
    flex-direction: column;  
    row-gap: 30px;  
  }  
  
  .video {  
    height: 100vh;  
    flex-direction: column;  
  }  
  
  .video__wrap {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
  }  
}
```



Breakpoints

Ми розуміємо, що існує багато пристроїв, кожен з яких має свої розширення. Відповідно, повинна існувати певна специфікація, яка описує, які межі розширень потрібно виставляти. Але її не існує. Отже яких розширень дотримуватись? Більшість дотримується контрольних точок визначених бібліотекою bootstrap.

Available breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px