

HTML и CSS

Практика

Козлов Антон



ТИНЬКОФФ



Вспомнить всё



О чём мы говорили на прошлой лекции?



Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей
- Что может CSS

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей
- Что может CSS
- Как подключить стили к странице

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей
- Что может CSS
- Как подключить стили к странице
- Синтаксис CSS, типы селекторов, единицы цвета и размеров

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей
- Что может CSS
- Как подключить стили к странице
- Синтаксис CSS, типы селекторов, единицы цвета и размеров
- Каскад, наследование, специфичность

Вспомнить всё

- Немного узнали о истории HTML и CSS. Как они развивались
- Для чего нужен HTML и CSS
- Узнали из каких элементов состоит html-документ и таблица стилей
- Что может CSS
- Как подключить стили к странице
- Синтаксис CSS, типы селекторов, единицы цвета и размеров
- Каскад, наследование, специфичность
- Блочная модель

Вспомнить всё



display

Определяет, как будет показан (или не показан) элемент на странице

display

Определяет, как будет показан (или не показан) элемент на странице

- **block**: элемент показывается как блочный, занимает всю доступную ширину и всегда начинается с новой строки

block

```
<p>  
<div>  
<h1> - <h6>  
<section>  
<form>  
<header>  
<footer>
```


display

Определяет, как будет показан (или не показан) элемент на странице

- **block**: элемент показывается как блочный, занимает всю доступную ширину и всегда начинается с новой строки
- **inline**: при возможности помещается в ту же строку, занимает столько места, сколько ему требуется

inline

` <a> <button> <input> <label> `

block

`<p>
<div>
<h1> - <h6>
<section>
<form>
<header>
<footer>`

display

Определяет, как будет показан (или не показан) элемент на странице

- **block**: элемент показывается как блочный, занимает всю доступную ширину и всегда начинается с новой строки
- **inline**: при возможности помещается в ту же строку, занимает столько места, сколько ему требуется
- **none**: элемент не отображается

display

Определяет, как будет показан (или не показан) элемент на странице

- **block**: элемент показывается как блочный, занимает всю доступную ширину и всегда начинается с новой строки
- **inline**: при возможности помещается в ту же строку, занимает столько места, сколько ему требуется
- **none**: элемент не отображается
- **inline-block**: ведут себя как инлайн-элементы, но им можно задать свойства блоков
- **flex, grid, table** и др.

display

Есть особенность у `display: inline-block`

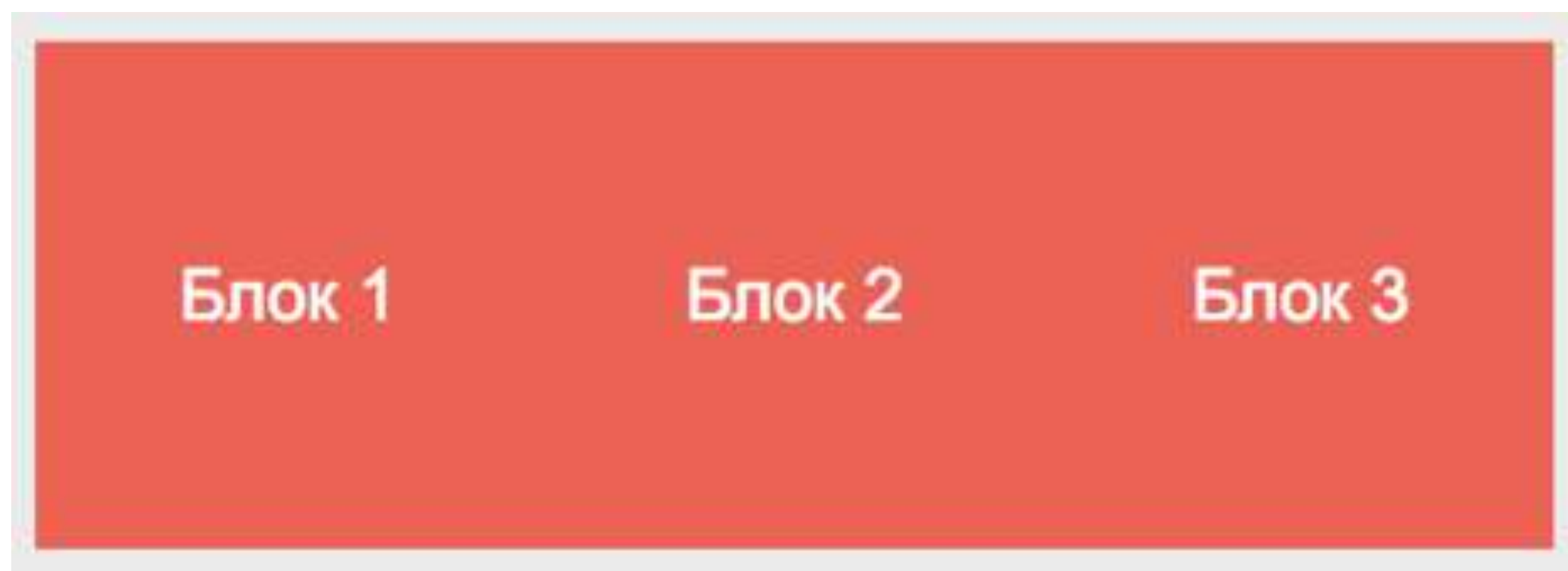


Загадочный отступ

```
<div class="box">  
  <div class="block">Блок 1</div>  
  <div class="block">Блок 2</div>  
  <div class="block">Блок 3</div>  
</div>
```

display

Есть особенность у `display: inline-block`



```
/* Способ 1 */  
.box {  
    font-size: 0;  
}  
  
<!-- Способ 2 -->  
<div>Блок 1</div><div>Блок 2</div>
```


Особенности box-model

Какой будет отступ у блоков расположенных друг под другом?

`margin-bottom: 50px;`

? px

`margin-top: 20px;`

Особенности box-model

Какой будет отступ у блоков расположенных друг под другом?

`margin-bottom: 50px;`

50px

`margin-top: 20px;`

Особенности box-model

Какой будет отступ у блоков расположенных друг под другом?

`margin-bottom: 50px;`

`margin-top: 20px;`

50px

Отступы `margin-top` и `margin-bottom` иногда объединяются в один, с размером равным наибольшему из них (или размеру одного, если они равны).

Это поведение известно как схлопывание внешних отступов (margin collapsing).



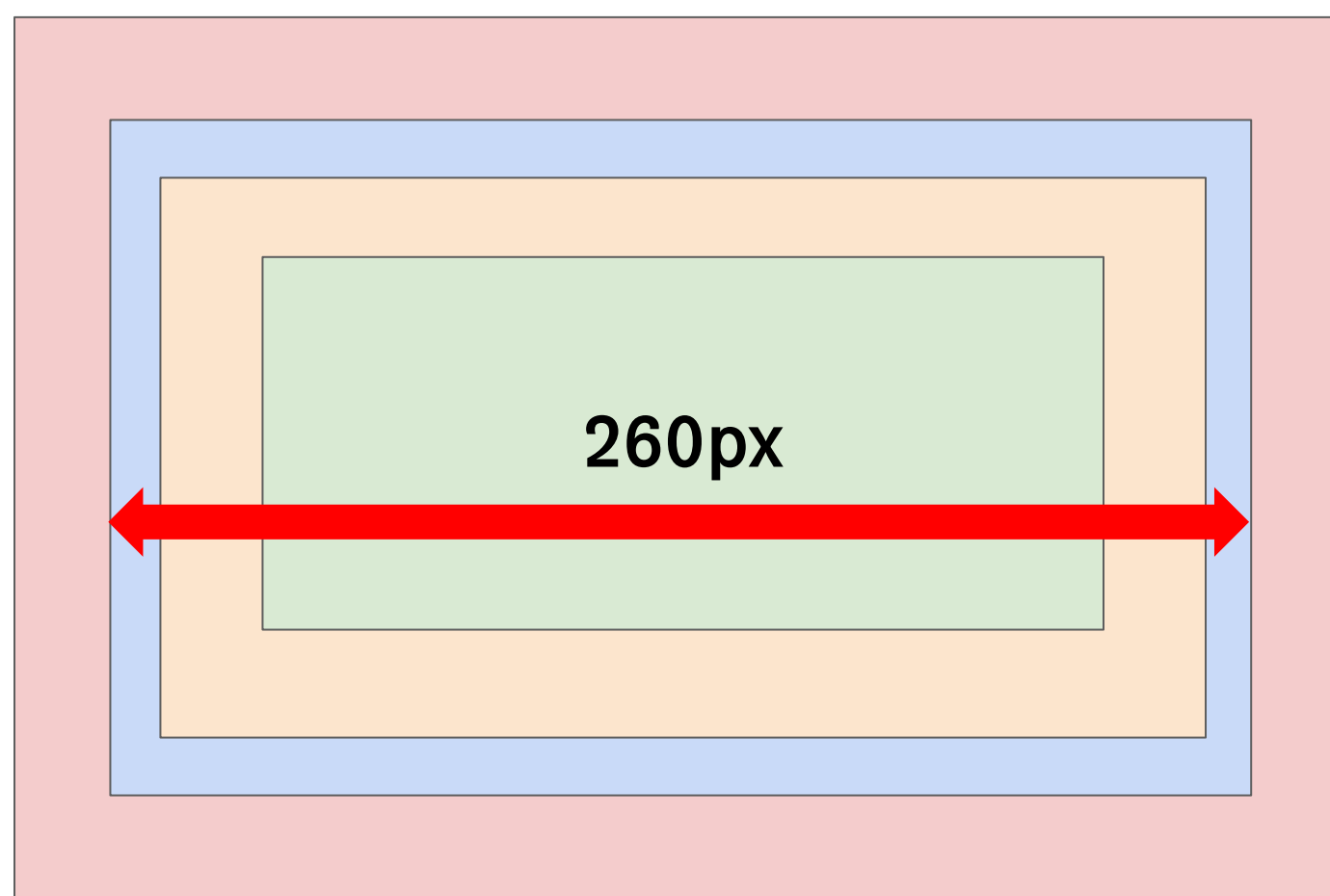
https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing



ТИНЬКОФФ

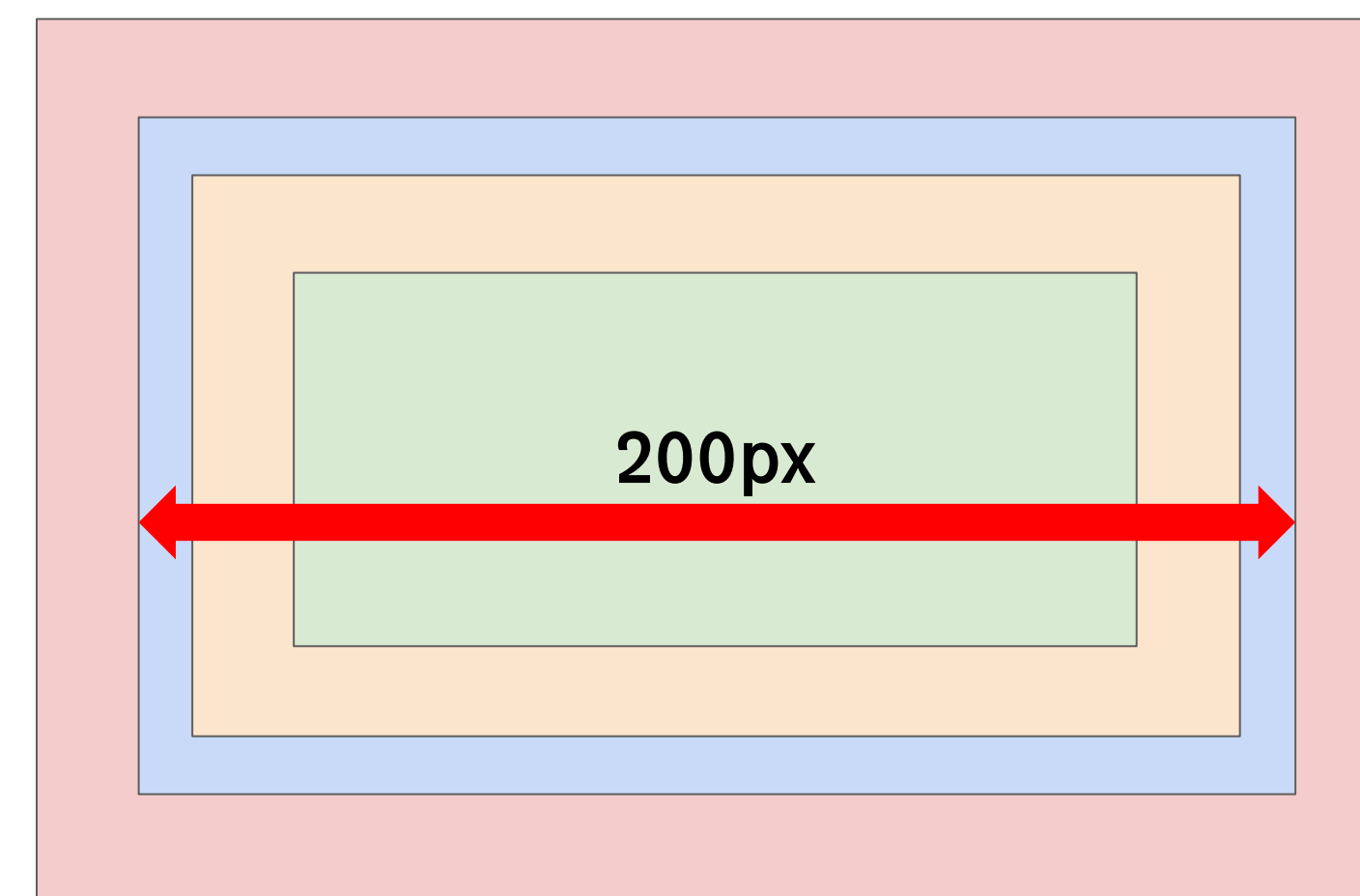
box-sizing

content-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
}
```

border-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
  box-sizing: border-box;  
}
```

max-width / max-height

А также min-width / min-height

Используются для задания
максимальных/минимальных значений

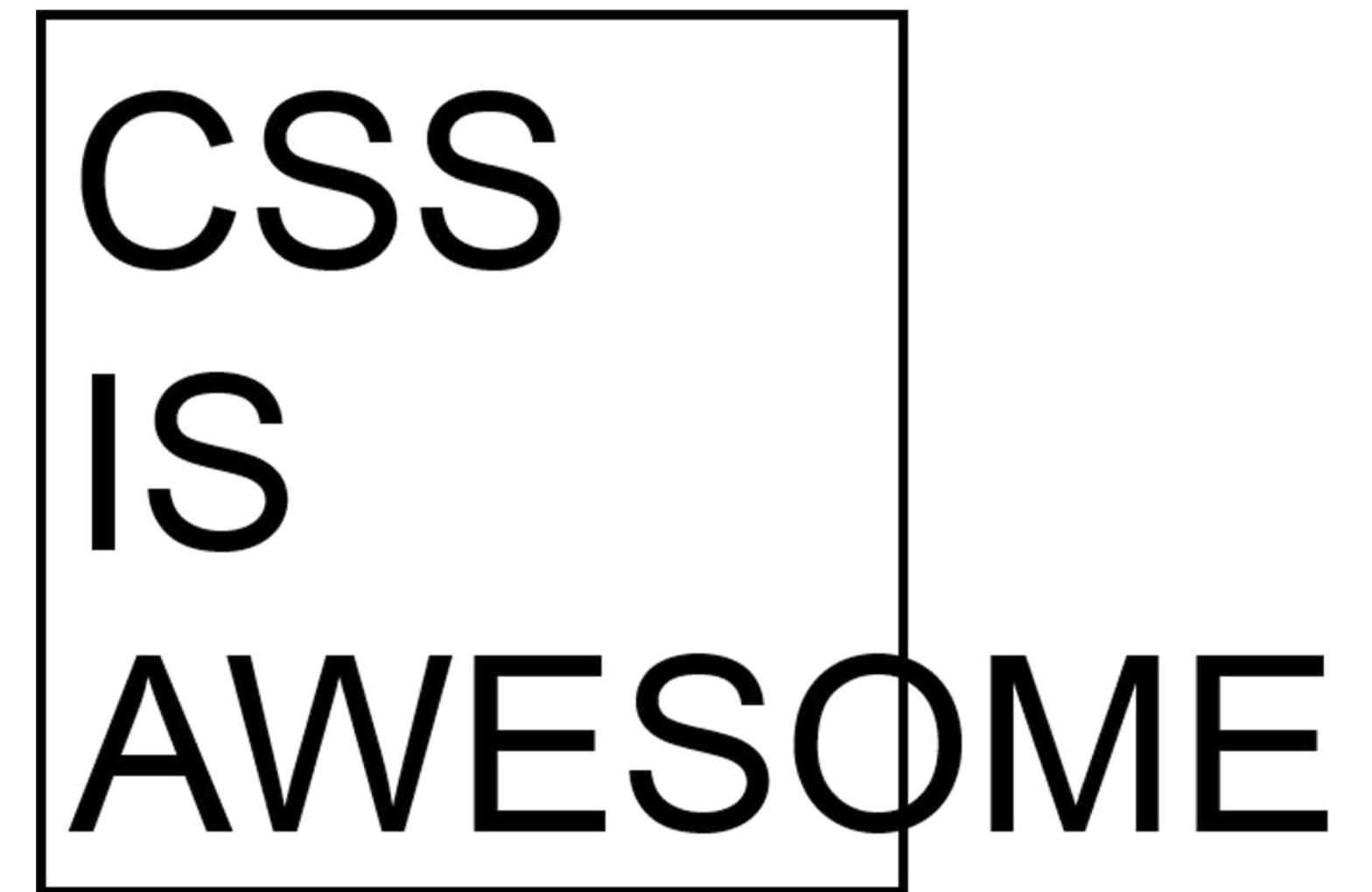
- Если элементу задано и свойство **width**,
и **max-width/min-width**, то последние
являются более приоритетными
- Разумеется, то же самое относится к свойствам
height/min-height/max-height

```
.box {  
    width: 300px;  
    max-width: 200px;  
}
```


overflow

Определяет, как отображать контент, если он не помещается в заданную область

- **visible**: контент отображается за пределами блока
- **hidden**: контент, который не помещается в заданную область, не отображается
- **scroll**: в блок добавляется скроллбар для прокрутки контента
- **auto**: если контент не помещается в заданную область, в блок добавляется скроллбар



`overflow: visible;`

position

Определяет способ позиционирования элемента

position

Определяет способ позиционирования элемента

- **static**: элемент находится в обычном потоке (свойство по умолчанию)

position

Определяет способ позиционирования элемента

- **static**: элемент находится в обычном потоке (свойство по умолчанию)
- **relative**: относительное позиционирование

position

Определяет способ позиционирования элемента

- **static**: элемент находится в обычном потоке (свойство по умолчанию)
- **relative**: относительное позиционирование
- **absolute**: позиционируется относительно родителя с заданным position (элемент выпадает из потока)

position

Определяет способ позиционирования элемента

- **static**: элемент находится в обычном потоке (свойство по умолчанию)
- **relative**: относительное позиционирование
- **absolute**: позиционируется относительно родителя с заданным position (элемент выпадает из потока)
- **fixed**: позиционируется относительно окна браузера (элемент выпадает из потока)

position

Определяет способ позиционирования элемента

- **static**: элемент находится в обычном потоке (свойство по умолчанию)
- **relative**: относительное позиционирование
- **absolute**: позиционируется относительно родителя с заданным position (элемент выпадает из потока)
- **fixed**: позиционируется относительно окна браузера (элемент выпадает из потока)
- **sticky**: «липкое» позиционирование



<https://itchief.ru/html-and-css/css-positioning-elements>

Сетки

Что такое сетка?

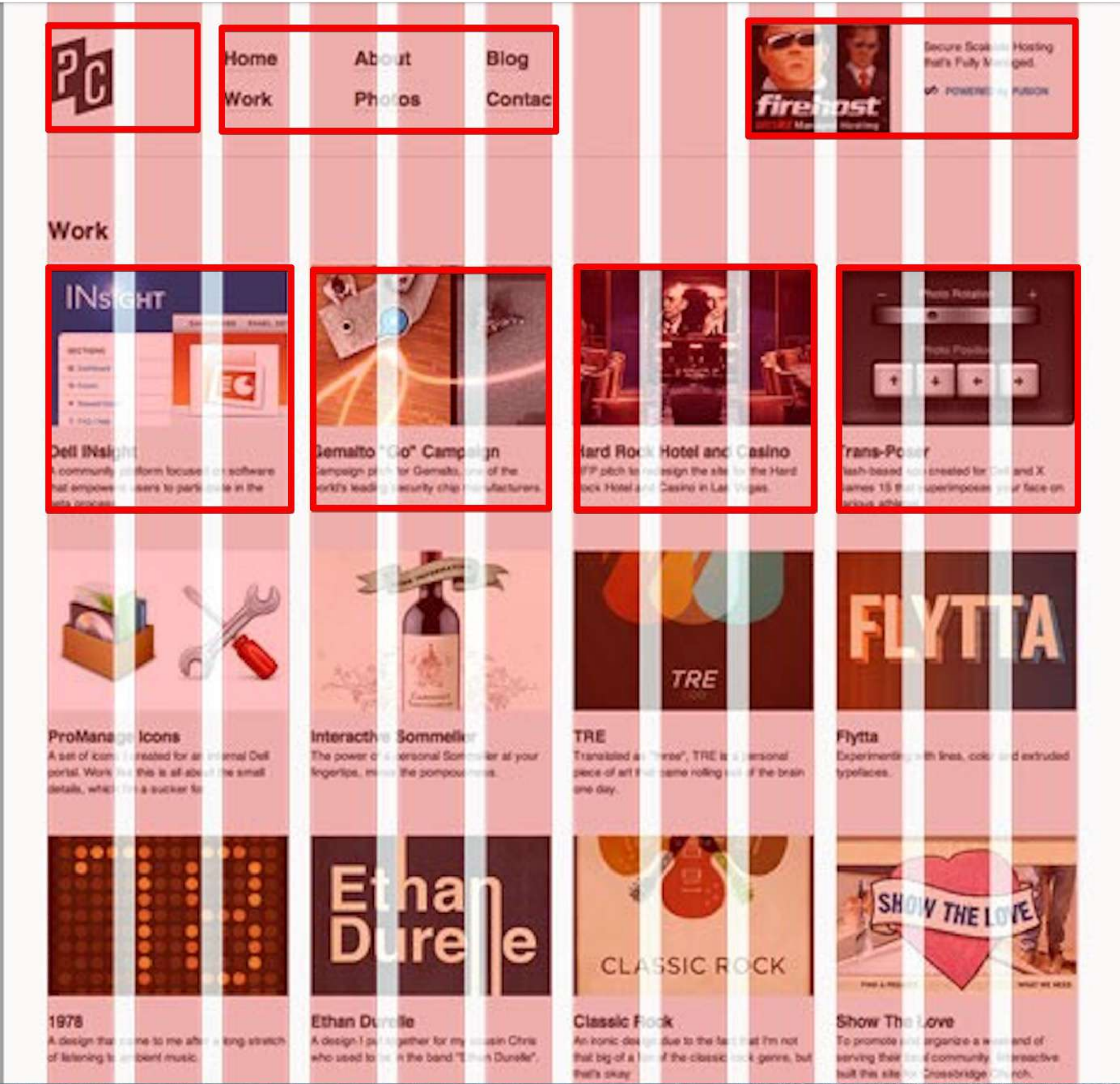
Сетки



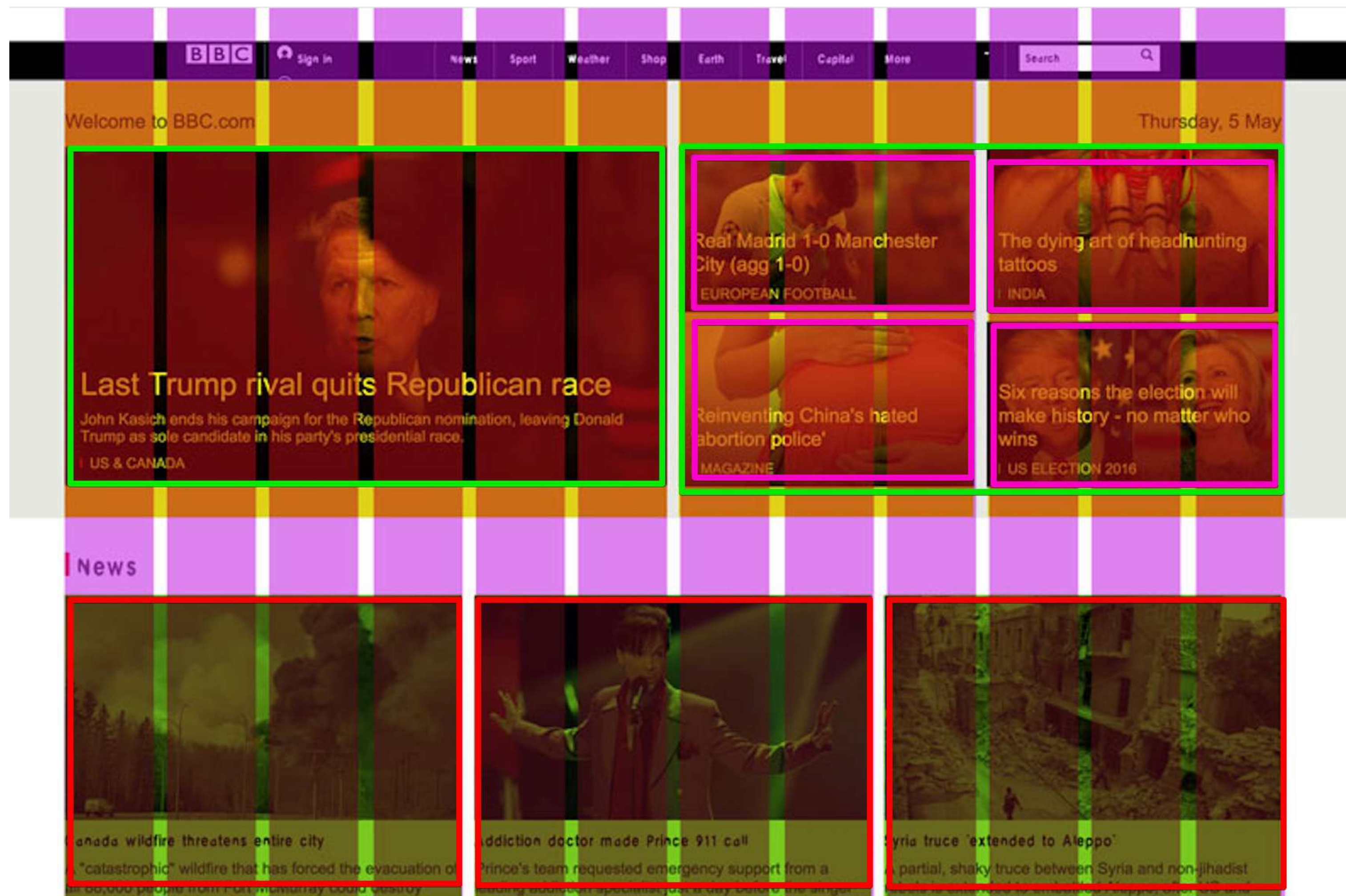
Сетки



Сетки

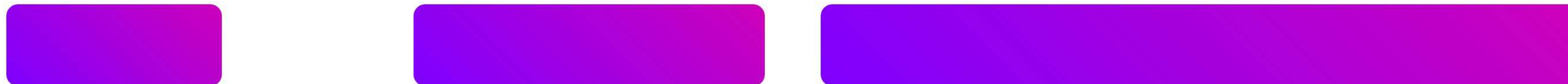


Сетки

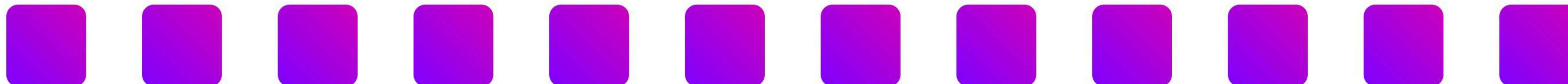


Сетки

HTML-сетки



12 колонок



Сетки

HTML

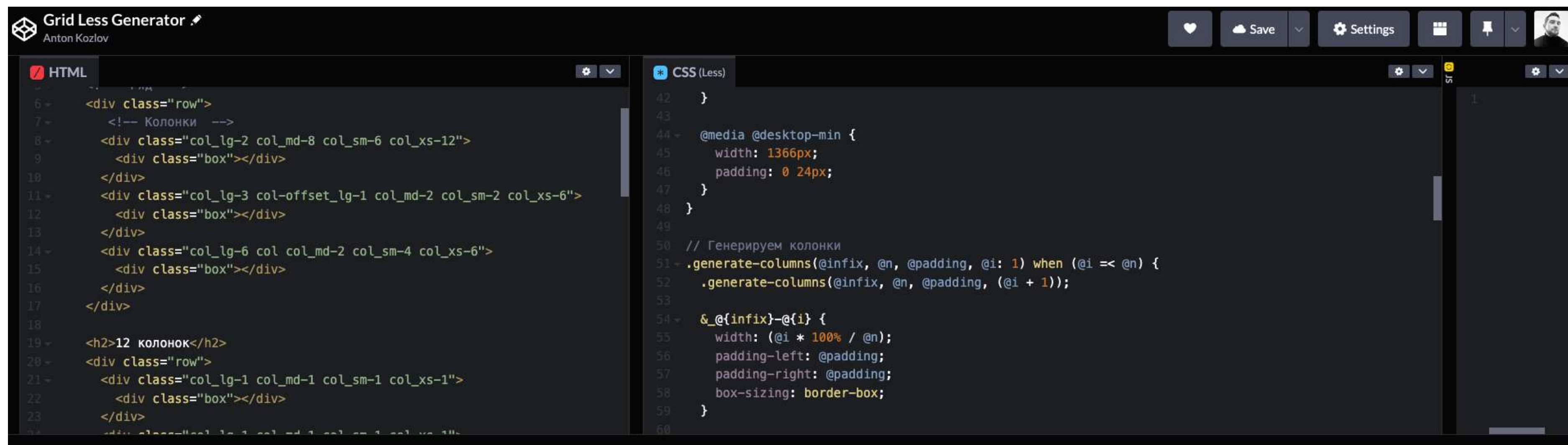
```
1 <body>
2 <!-- Контейнер -->
3 <div class="container">
4   <h1>HTML-сетки</h1>
5   <!-- Ряд -->
6   <div class="row">
7     <!-- Колонки -->
8     <div class="col_lg-2 col_md-8 col_sm-6 col_xs-12">
9       <div class="box"></div>
10    </div>
11    <div class="col_lg-3 col_offset_lg-1 col_md-2 col_sm-2 col_xs-6">
12      <div class="box"></div>
13    </div>
14    <div class="col_lg-6 col col_md-2 col_sm-4 col_xs-6">
15      <div class="box"></div>
16    </div>
17  </div>
18
19  <h2>12 колонок</h2>
20  <div class="row">
21    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
22      <div class="box"></div>
23    </div>
24    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
25      <div class="box"></div>
26    </div>
27    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
28      <div class="box"></div>
29    </div>
30    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
31      <div class="box"></div>
32    </div>
33    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
34      <div class="box"></div>
35    </div>
36    <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
```

HTML-сетки



ТИНЬКОФФ

Сетки



```
HTML
6 <div class="row">
7   <!-- Колонки -->
8   <div class="col_lg-2 col_md-8 col_sm-6 col_xs-12">
9     <div class="box"></div>
10  </div>
11  <div class="col_lg-3 col_offset_lg-1 col_md-2 col_sm-2 col_xs-6">
12    <div class="box"></div>
13  </div>
14  <div class="col_lg-6 col_md-2 col_sm-4 col_xs-6">
15    <div class="box"></div>
16  </div>
17 </div>
18
19 <h2>12 колонок</h2>
20 <div class="row">
21   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
22     <div class="box"></div>
23   </div>
24   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
25     <div class="box"></div>
26   </div>
27   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
28     <div class="box"></div>
29   </div>
30   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
31     <div class="box"></div>
32   </div>
33   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
34     <div class="box"></div>
35   </div>
36   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
37     <div class="box"></div>
38   </div>
39   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
40     <div class="box"></div>
41   </div>
42   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
43     <div class="box"></div>
44   </div>
45   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
46     <div class="box"></div>
47   </div>
48   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
49     <div class="box"></div>
50   </div>
51   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
52     <div class="box"></div>
53   </div>
54   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
55     <div class="box"></div>
56   </div>
57   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
58     <div class="box"></div>
59   </div>
60   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
61     <div class="box"></div>
62   </div>
63   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
64     <div class="box"></div>
65   </div>
66   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
67     <div class="box"></div>
68   </div>
69   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
70     <div class="box"></div>
71   </div>
72   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
73     <div class="box"></div>
74   </div>
75   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
76     <div class="box"></div>
77   </div>
78   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
79     <div class="box"></div>
80   </div>
81   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
82     <div class="box"></div>
83   </div>
84   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
85     <div class="box"></div>
86   </div>
87   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
88     <div class="box"></div>
89   </div>
90   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
91     <div class="box"></div>
92   </div>
93   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
94     <div class="box"></div>
95   </div>
96   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
97     <div class="box"></div>
98   </div>
99   <div class="col_lg-1 col_md-1 col_sm-1 col_xs-1">
100    <div class="box"></div>
101  </div>
102 </div>
```

```
CSS (Less)
42 }
43
44 @media @desktop-min {
45   width: 1366px;
46   padding: 0 24px;
47 }
48
49 // Генерируем колонки
50 .generate-columns(@infix, @n, @padding, @i: 1) when (@i <= @n) {
51   .generate-columns(@infix, @n, @padding, (@i + 1));
52 }
53
54 &_{infix}-@{i} {
55   width: (@i * 100% / @n);
56   padding-left: @padding;
57   padding-right: @padding;
58   box-sizing: border-box;
59 }
60
```

HTML-сетки



12 колонок



ТИНЬКОФФ



<https://codepen.io/ante/pen/wvEMMON>

Сетки

Представьте, что это макет вашей страницы.

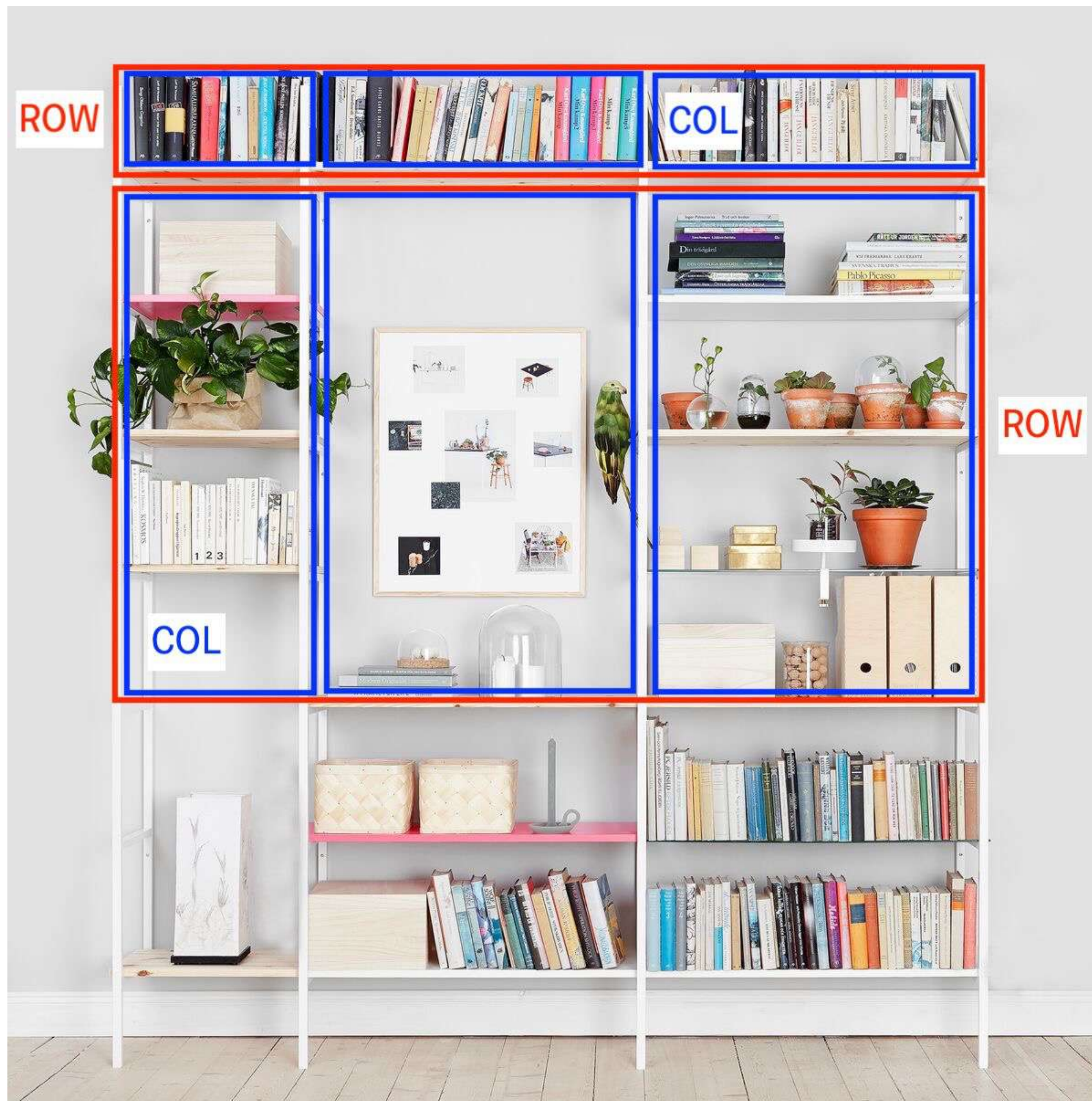
Для организации пространства использован стеллаж (grid) у которого есть ряды (row) и колонки (col):



ТИНЬКОФФ



Сетки



ТИНЬКОФФ

Преимущества сеток

- Позволяет быстро организовать макет

Преимущества сеток

- Позволяет быстро организовать макет
- Можно управлять поведением содержимого на разных разрешениях экрана

Преимущества сеток

- Позволяет быстро организовать макет
- Можно управлять поведением содержимого на разных разрешениях экрана
- Существенно сокращает количество CSS кода для реализации адаптивной версии

Преимущества сеток

- Позволяет быстро организовать макет
- Можно управлять поведением содержимого на разных разрешениях экрана
- Существенно сокращает количество CSS кода для реализации адаптивной версии
- Позволяет задать поведение и отступы у элементов на странице

Преимущества сеток

- Позволяет быстро организовать макет
- Можно управлять поведением содержимого на разных разрешениях экрана
- Существенно сокращает количество CSS кода для реализации адаптивной версии
- Позволяет задать поведение и отступы у элементов на странице
- И т. д.

Сетки

Особенности сеток

Особенности сеток

- Максимальное количество колонок заранее определено

Hello, world!

This is an example to show the potential of an offcanvas layout pattern in Bootstrap. Try some responsive-range viewport sizes to see it in action.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Link

Link

Link

Link

Link

Link

Link

Link

Link

Особенности сеток

- Максимальное количество колонок заранее определено
- У колонок одинаковая ширина

Hello, world!

This is an example to show the potential of an offcanvas layout pattern in Bootstrap. Try some responsive-range viewport sizes to see it in action.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Link

Link

Link

Link

Link

Link

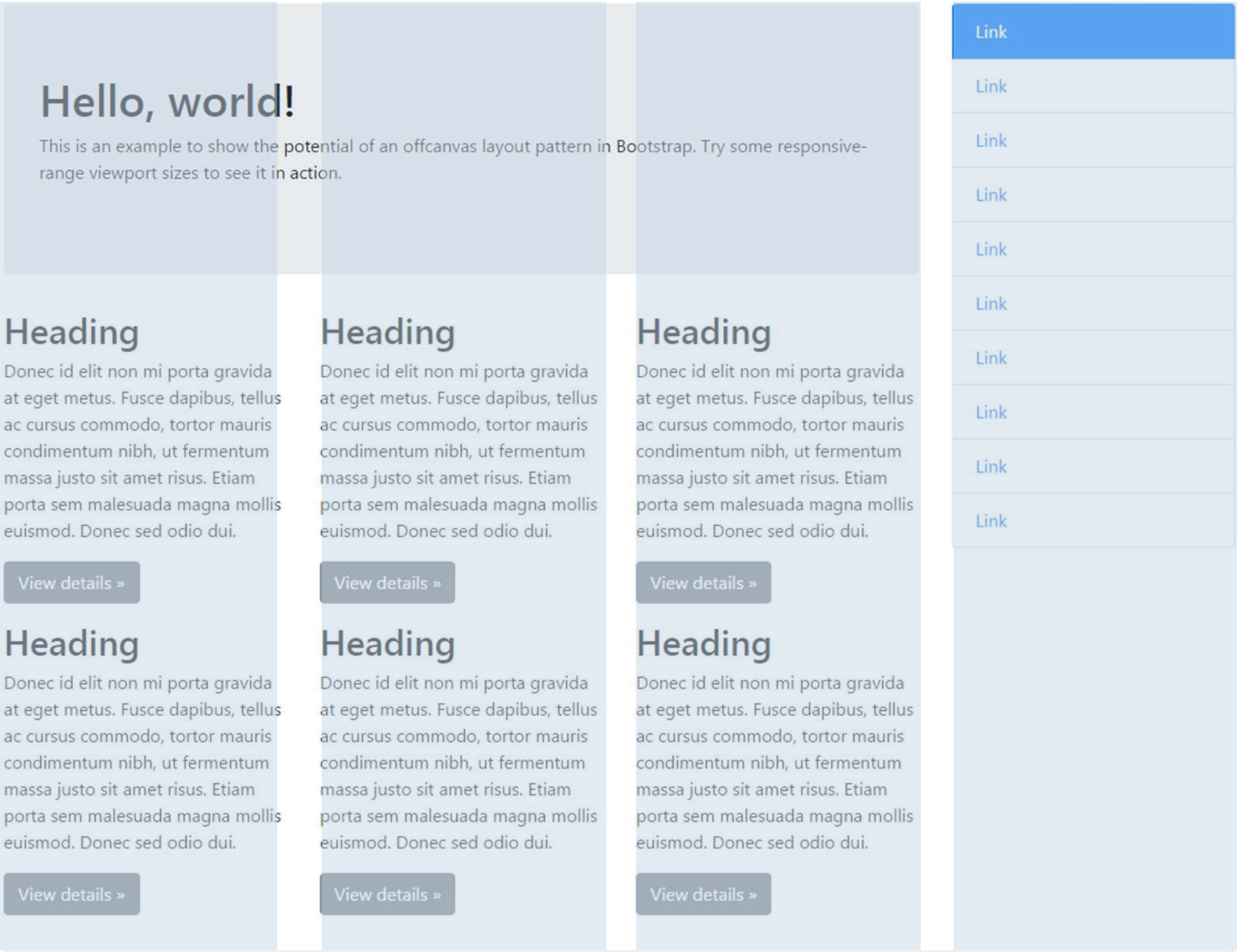
Link

Link

Link

Особенности сеток

- Максимальное количество колонок заранее определено
- У колонок одинаковая ширина
- Между колонками одинаковые отступы



Полезные ссылки:

[Понятие модульной сетки и ее главные особенности](#)

[Полное руководство по сеткам в веб дизайне](#)

[Модульная сетка сайта](#)

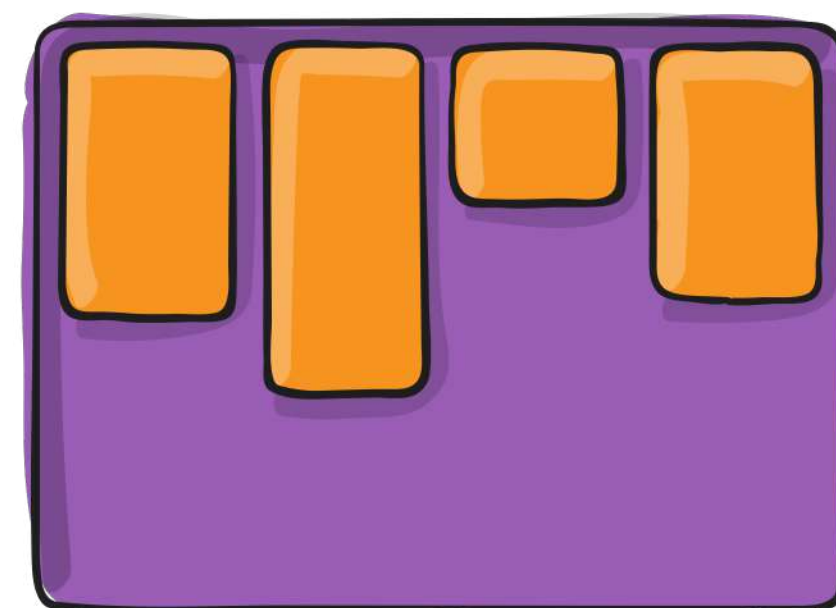
Flexbox



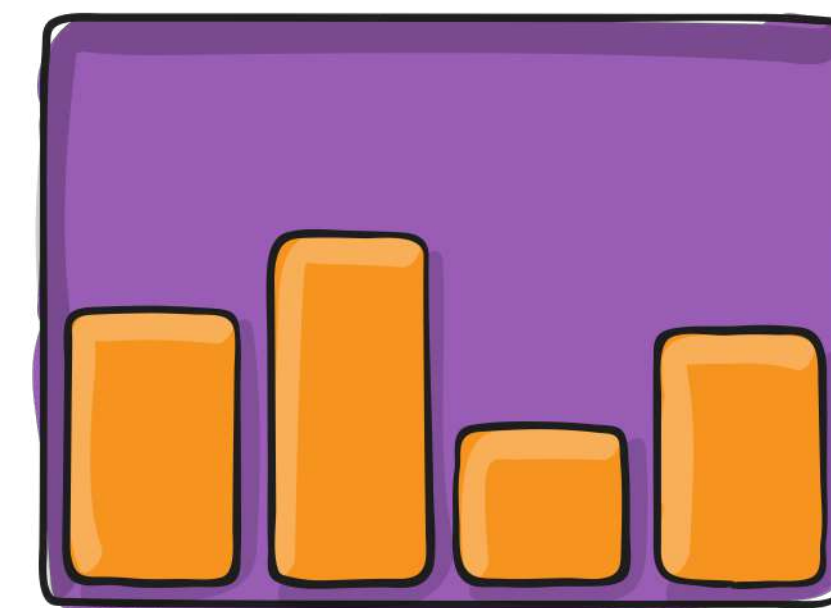
Flexbox

CSS Flexbox (Flexible Box Layout Module) — способ расположения блоков на странице. Эта технология, созданная именно для раскладки элементов.

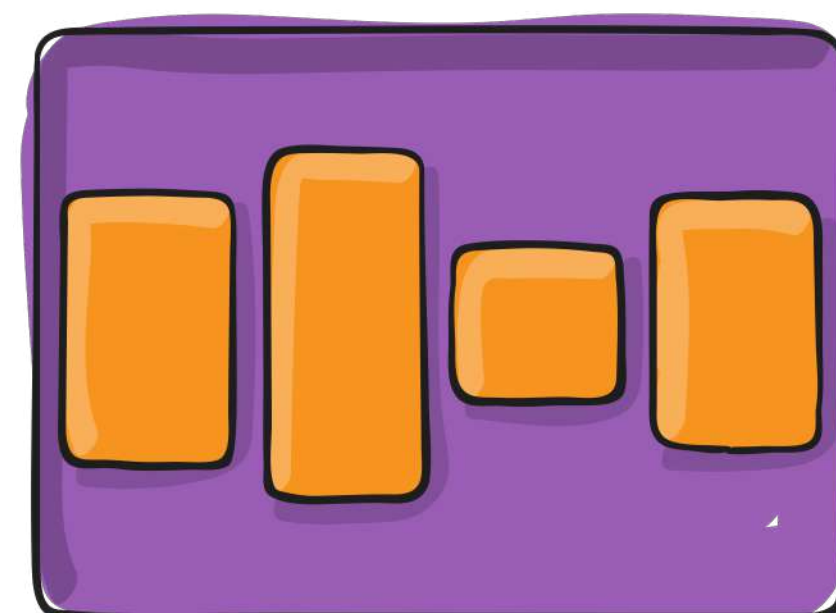
flex-start



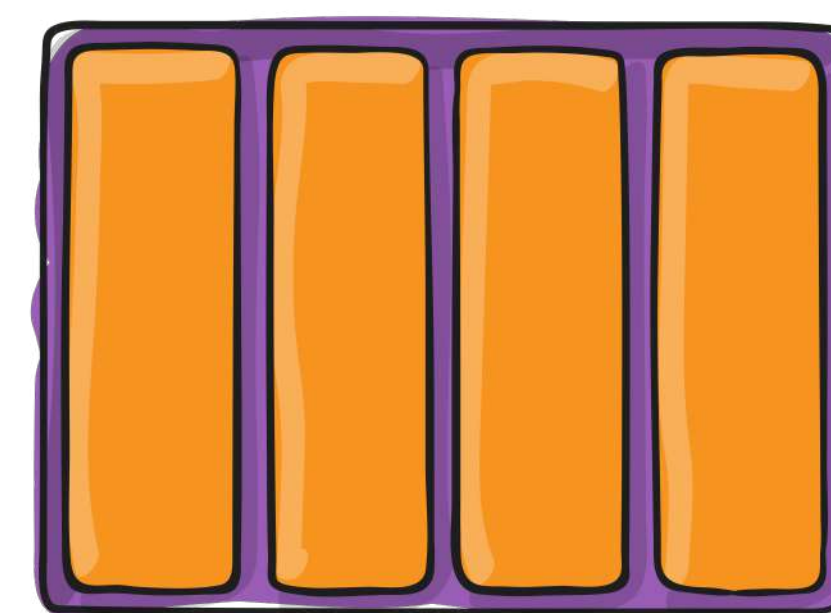
flex-end



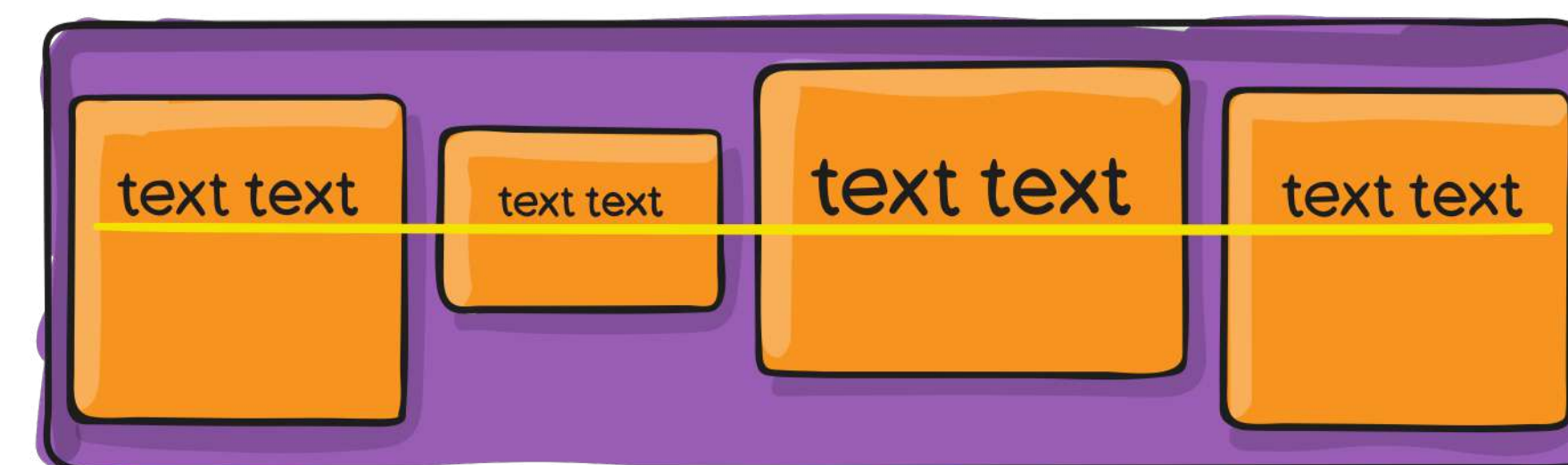
center



stretch



baseline



Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.

Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Очень просто выравнивать элементы по вертикали, горизонтали или по базовой линии текста.

Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Очень просто выравнивать элементы по вертикали, горизонтали или по базовой линии текста.
- Можно менять порядок элементов несмотря на их порядок в html.

Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Очень просто выравнивать элементы по вертикали, горизонтали или по базовой линии текста.
- Можно менять порядок элементов несмотря на их порядок в html.
- Элементы могут выстраиваться в несколько строк/столбцов, занимая все свободное место.

Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Очень просто выравнивать элементы по вертикали, горизонтали или по базовой линии текста.
- Можно менять порядок элементов несмотря на их порядок в html.
- Элементы могут выстраиваться в несколько строк/столбцов, занимая все свободное место.
- Удобно создавать колонки одинаковой высоты.

Flexbox

Основные преимущества Flexbox:

- Все блоки делаются резиновыми. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Очень просто выравнивать элементы по вертикали, горизонтали или по базовой линии текста.
- Можно менять порядок элементов несмотря на их порядок в html.
- Элементы могут выстраиваться в несколько строк/столбцов, занимая все свободное место.
- Удобно создавать колонки одинаковой высоты
- Синтаксис CSS правил очень прост и понятен для освоения.

Для самостоятельного изучения:

[Полное руководство по Flexbox](#)

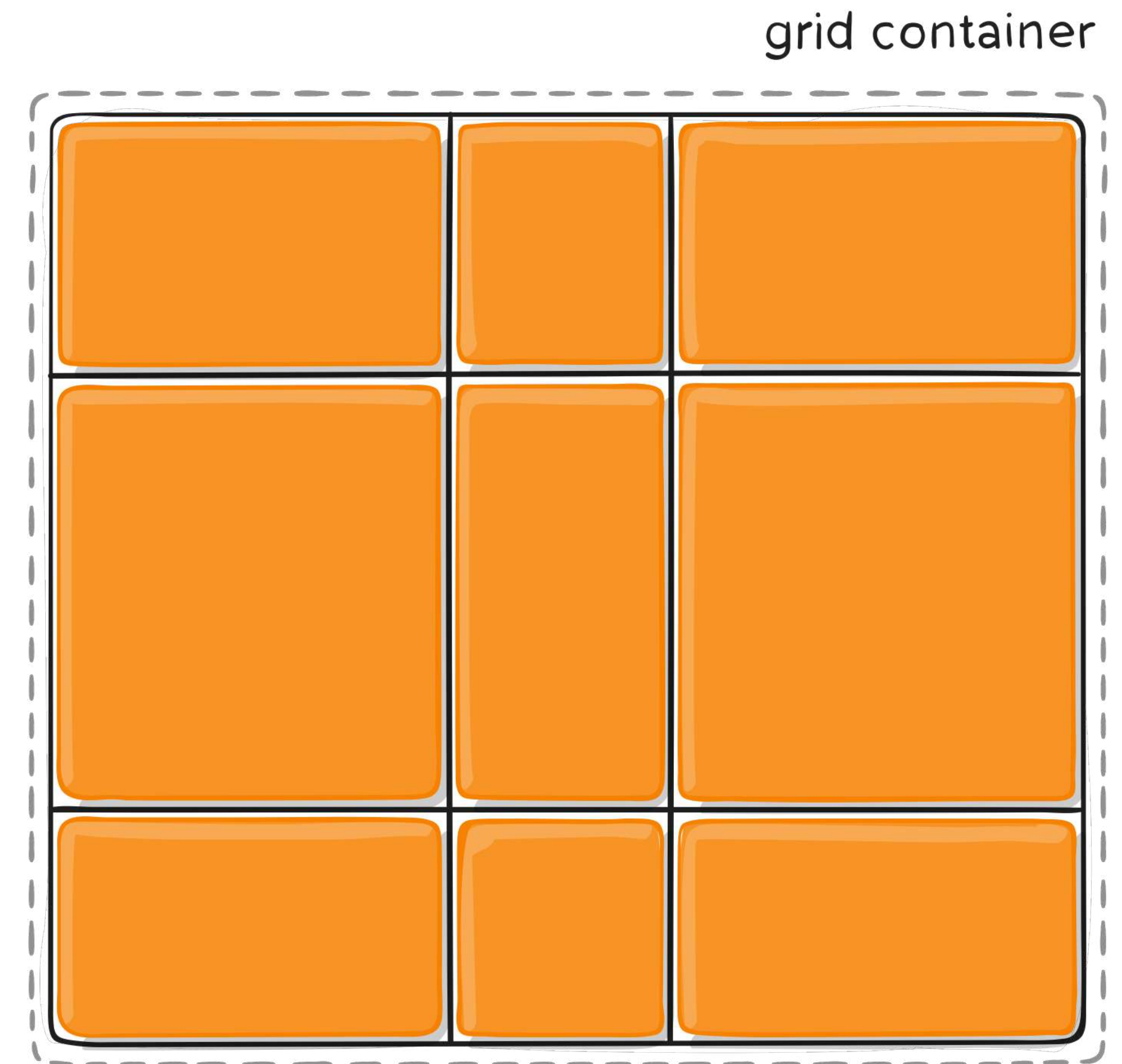
[Flexbox Fundamentals](#)

[Flexbox Froggy](#)

CSS Grid Layout

CSS Grid

CSS Grid Layout — двухмерная сетка для CSS. Можно использовать для размещения основных областей страницы или небольших элементов интерфейса.



CSS Grid

Преимущества CSS Grid:

- В отличие от Flexbox позволяет управлять элементами в двух измерениях.

CSS Grid

Преимущества CSS Grid:

- В отличие от Flexbox позволяет управлять элементами в двух измерениях.
- Упрощение разметки, меньше html и css кода.

CSS Grid

Преимущества CSS Grid:

- В отличие от Flexbox позволяет управлять элементами в двух измерениях.
- Упрощение разметки, меньше html и css кода.
- Нет ограничения по элементам построения сетки макета.

CSS Grid

Преимущества CSS Grid:

- В отличие от Flexbox позволяет управлять элементами в двух измерениях.
- Упрощение разметки, меньше html и css кода.
- Нет ограничения по элементам построения сетки макета.
- Можно менять layout не трогая разметку.

CSS Grid

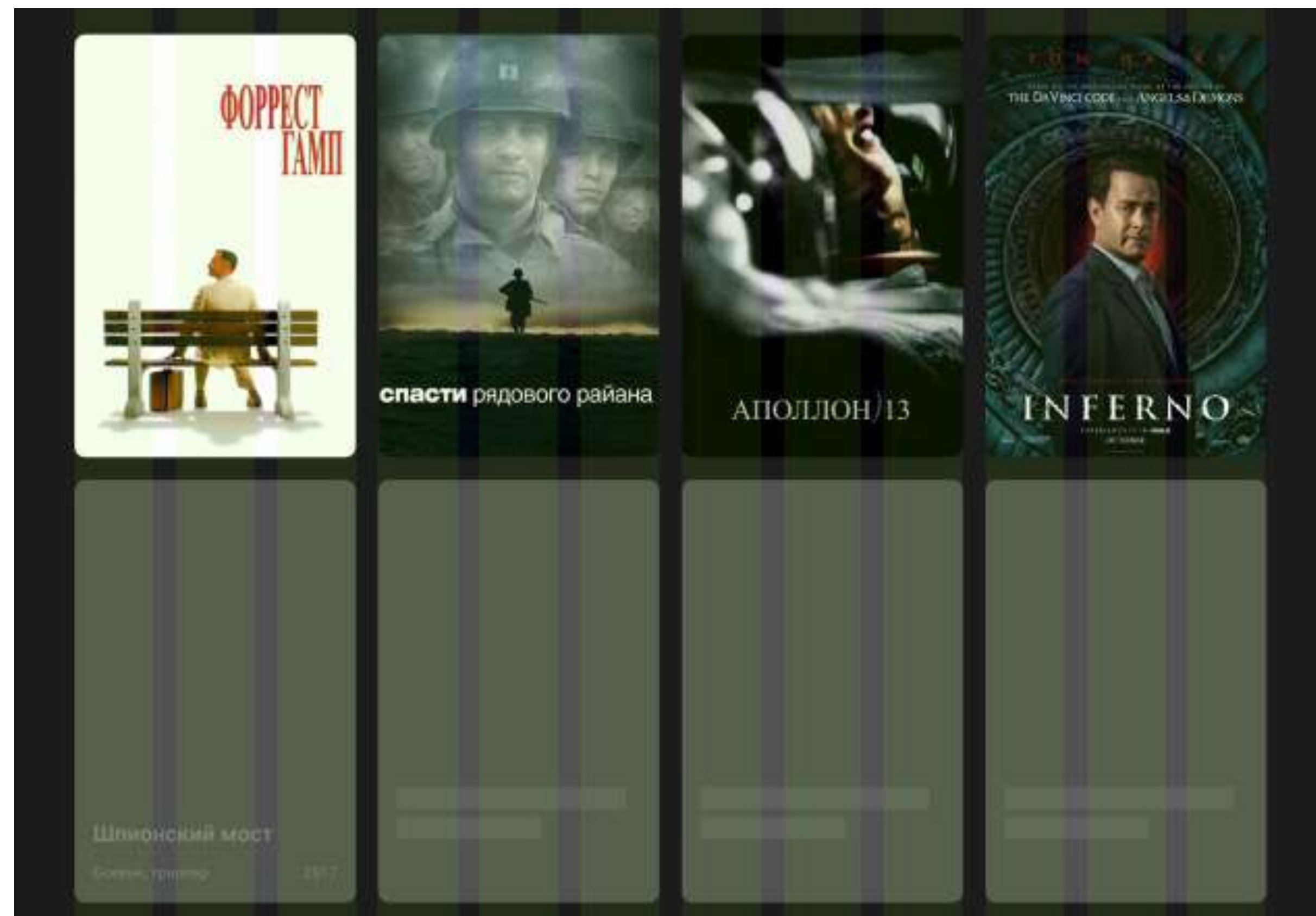
Преимущества CSS Grid:

- В отличии от Flexbox позволяет управлять элементами в двух измерениях.
- Упрощение разметки, меньше html и css кода.
- Нет ограничения по элементам построения сетки макета.
- Можно менять layout не трогая разметку.
- И т. д.

Grid vs Flexbox

Grid VS Flexbox

Давайте сверстаем простую сетку и посмотрим на разницу в коде



Grid VS Flexbox

Flexbox

```
<div class="container">
  <div class="row">
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
    <div class="col">
      <div class="box"></div>
    </div>
  </div>
</div>
```

```
.row {
  display: flex;
  flex-wrap: wrap;
  margin-left: -12px;
  margin-right: -12px;
}

.col {
  width: 25%;
  padding: 0 12px;
  margin-top: 24px;
  box-sizing: border-box;
}
```



<https://codepen.io/ante/pen/GRNrLvY>

Grid VS Flexbox

Grid

```
<div class="container">
  <div class="grid-container">
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
  </div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-gap: 24px;
}
```



<https://codepen.io/ante/pen/GRNrLvY>

Grid VS Flexbox

Grid



```
<div class="container">
  <div class="grid-container">
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
  </div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-gap: 24px;
}
```



<https://codepen.io/ante/pen/GRNrLvY>



ТИНЬКОФФ

CSS Grid

Для самостоятельного изучения:

[Вёрстка на Grid в CSS. Полное руководство и справочник](#)

[Шпаргалка по Grid CSS](#)

[CSS Grid Layout](#)

[Grid by Example](#)

[Grid Garden](#)

Контейнер

Зачем нужен контейнер?

Контейнер



Контейнер

- Ограничивает ширину контента

Контейнер

- Ограничивает ширину контента
- Управляет шириной сайта при адаптации

Контейнер

- Ограничивает ширину контента
- Управляет шириной сайта при адаптации
- Управляет боковыми отступами

Контейнер

- Ограничивает ширину контента
- Управляет шириной сайта при адаптации
- Управляет боковыми отступами
- Ваши варианты?

Переменные CSS

(CSS Custom Properties)

Переменные CSS

CSS переменные (CSS Custom Properties) — сущности, определяемые автором CSS, хранящие конкретные значения, которые можно повторно использовать в документе.

```
:root {
  --tui-base-01: #222;
  --tui-base-02: #333;
  --tui-base-03: grey;
  --tui-base-04: #959595;
  --tui-base-05: #b0b0b0;
  --tui-base-06: #d7d7d7;
  --tui-base-07: #ededed;
  --tui-base-08: #f6f6f6;
  --tui-base-09: #fff;
  --tui-focus: rgba(51,51,51,0.64);
  --tui-secondary: hsla(0,0%,100%,0.16);
  --tui-secondary-hover: hsla(0,0%,100%,0.24);
  --tui-secondary-active: hsla(0,0%,100%,0.4);
  --tui-clear: hsla(0,0%,100%,0.16);
  --tui-clear-disabled: rgba(0,0,0,0.04);
  --tui-clear-hover: hsla(0,0%,100%,0.24);
  --tui-clear-active: hsla(0,0%,100%,0.4);
  --tui-error-fill: #ff8c67;
  --tui-error-bg: rgba(244,87,37,0.32);
  --tui-error-bg-hover: rgba(244,87,37,0.4);
  --tui-success-fill: #4ac99b;
  --tui-success-bg: rgba(74,201,155,0.32);
  --tui-success-bg-hover: rgba(74,201,155,0.4);
  --tui-warning-fill: #ffc700;
  --tui-warning-bg: rgba(255,199,0,0.32);
  --tui-warning-bg-hover: rgba(255,199,0,0.4);
  --tui-info-fill: #70b6f6;
  --tui-info-bg: rgba(112,182,246,0.32);
  --tui-info-bg-hover: rgba(112,182,246,0.4);
  --tui-neutral-fill: #959ba4;
  --tui-neutral-bg: #959ba4;
  --tui-neutral-bg-hover: #959ba4;
  --tui-text-01: #fff;
  --tui-text-02: hsla(0,0%,100%,0.72);
  --tui-text-03: hsla(0,0%,100%,0.6);
  --tui-link: #6788ff;
  --tui-link-hover: #526ed3;
  --tui-positive: #44c596;
  --tui-positive-hover: #3aa981;
  --tui-negative: #ff8c67;
  --tui-negative-hover: #bb593a;
}
```

Переменные CSS

Как объявить и использовать переменную?

```
:root {  
  --main-color: #0c99c7;  
  --radius: 8px;  
}  
  
.button {  
  border-radius: var(--radius);  
  background-color: var(--main-color);  
}  
  
.link {  
  color: var(--main-color);  
}
```

Переменные CSS

А можно объявить локально

```
.alert {  
  --alert-color: #d80d0d;  
}  
  
.alert p {  
  color: var(--alert-color);  
  border: 1px solid var(--alert-color);  
}
```


Переменные CSS

Зачем?

- Уменьшение дублирования в таблице стилей

Переменные CSS

Зачем?

- Уменьшение дублирования в таблице стилей
- Легко менять значение стилей в больших и сложных веб-сайтах

Переменные CSS

Зачем?

- Уменьшение дублирования в таблице стилей
- Легко менять значение стилей в больших и сложных веб-сайтах
- Есть доступ из JS, можно менять их значение

Переменные CSS

Зачем?

- Уменьшение дублирования в таблице стилей
- Легко менять значение стилей в больших и сложных веб-сайтах
- Есть доступ из JS, можно менять их значение
- Использование в SVG

Переменные CSS

Для самостоятельного изучения:

[Использование переменных в CSS](#)

[Изучите CSS-переменные за 5 минут](#)

[Что умеют и чего не умеют CSS Custom Properties](#)

Отладка CSS



Отладка верстки

- Используем консоль в Dev Tools

```
Styles  Computed  Event Listeners  DOM Breakpoints  >>
Filter  :hov .cls +
element.style {
}
.input_size_ws-head .input__control { <style>...</style>
  border-right-width: 30px;
  height: 36px;
  border-left-width: 13px;
}
.input_ahead_yes .input__control { <style>...</style>
  position: relative;
  z-index: 9;
}
.input_theme_websearch .input__control { <style>...</style>
  display: block;
}
.input_theme_websearch .input__control { <style>...</style>
  font-family: inherit;
}
.input__control { <style>...</style>
  z-index: 3;
  width: 100%;
  margin: ▶ 0;
  padding: ▶ 0;
```

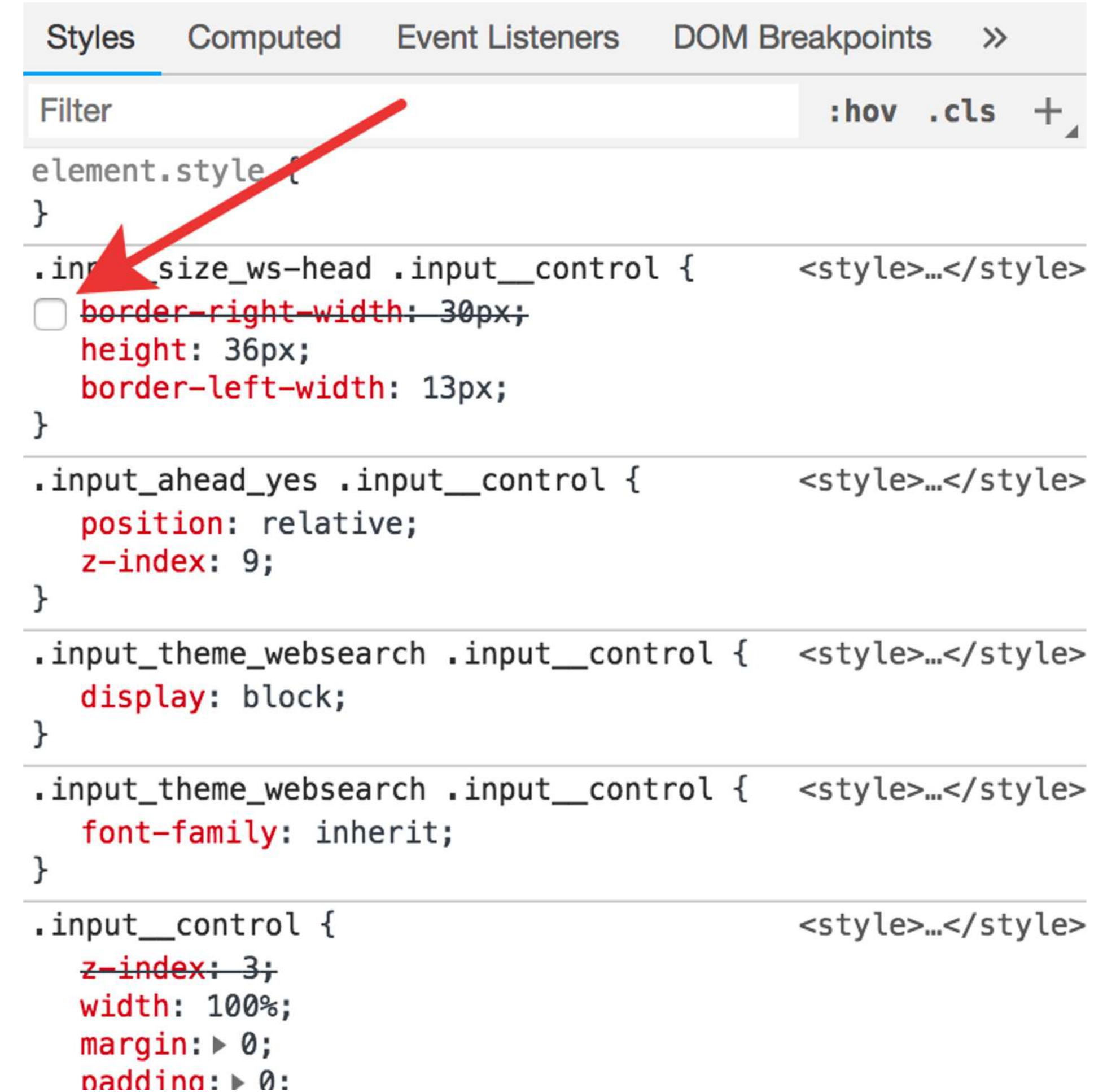
Отладка верстки

- Используем консоль в Dev Tools
- Смотрим свойства, которые не зачеркнуты

```
Styles  Computed  Event Listeners  DOM Breakpoints  >>
Filter  :hov .cls +
element.style {
}
.input_size_ws-head .input__control { <style>...</style>
  border-right-width: 30px;
  height: 36px;
  border-left-width: 13px;
}
.input_ahead_yes .input__control { <style>...</style>
  position: relative;
  z-index: 9;
}
.input_theme_websearch .input__control { <style>...</style>
  display: block;
}
.input_theme_websearch .input__control { <style>...</style>
  font-family: inherit;
}
.input__control { <style>...</style>
  z-index: 3;
  width: 100%;
  margin: ▶ 0;
  padding: ▶ 0;
```

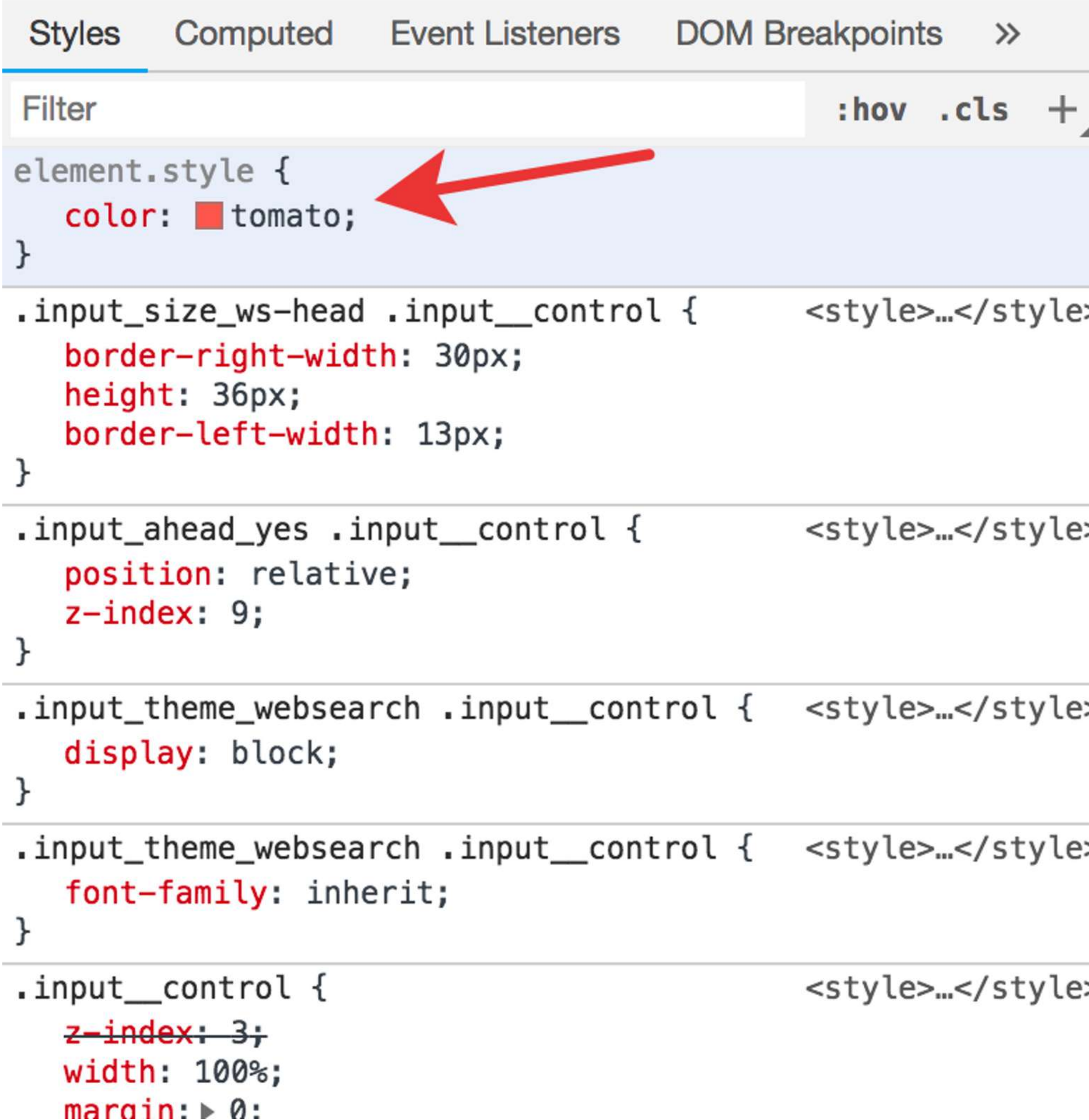

Отладка верстки

- Используем консоль в Dev Tools
- Смотрим свойства, которые не зачеркнуты
- Пробуем включать/выключать или менять значения



Отладка верстки

- Используем консоль в Dev Tools
- Смотрим свойства, которые не зачеркнуты
- Пробуем включать/выключать или менять значения
- Свои свойства пишем в `element.style` или под селекторами



```
Styles  Computed  Event Listeners  DOM Breakpoints  >>
Filter  :hov .cls +
element.style {
  color: tomato;
}
.input_size_ws-head .input__control { <style>...</style>
  border-right-width: 30px;
  height: 36px;
  border-left-width: 13px;
}
.input_ahead_yes .input__control { <style>...</style>
  position: relative;
  z-index: 9;
}
.input_theme_websearch .input__control { <style>...</style>
  display: block;
}
.input_theme_websearch .input__control { <style>...</style>
  font-family: inherit;
}
.input__control { <style>...</style>
  z-index: 3;
  width: 100%;
  margin: 0;
```


Отладка верстки

- Используем консоль в Dev Tools
- Смотрим свойства, которые не зачеркнуты
- Пробуем включать/выключать или менять значения
- Свои свойства пишем в `element.style` или под селекторами
- Вычисленные стили

Styles

Computed

Event Listeners

DOM Breakpoints

>>

position0

margin-

border-

padding-

685 x 36

013-300

0

Filter

☐ Show all

▶ background-attachment

scroll

▶ background-clip

border-box

▶ background-color

rgba(0, 0, 0, 0)

▶ background-image

none

▶ background-origin

padding-box

▶ background-position-x

0px

▶ background-position-y

0px

▶ background-repeat-x

▶ background-repeat-y

▶ background-size

auto

CSS
IS
AWESOME

CSS – это круто! Но все ли с ним хорошо?

Недостатки CSS

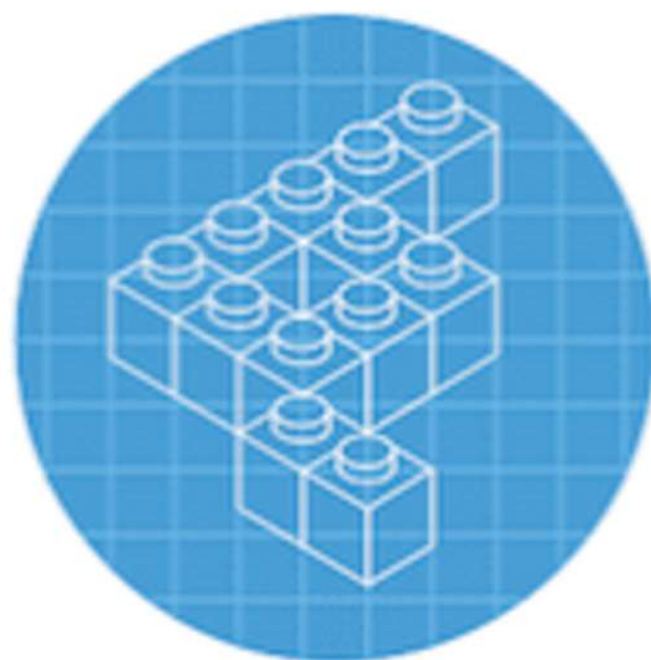
1. Отсутствие пространства имен дает возможность совпадения селекторов и их неожиданных влияний.
2. Отсутствие изоляции: часть CSS-свойств наследуются.
3. CSS сложно понять без разметки, к которой он применен.
4. В CSS поощряется каскад, использование которого связывает стилизацию и разметку, что усложняет модификацию и того, и другого.



Методология БЭМ



Методология БЭМ



Методология **БЭМ** создана в Яндексе для разработки сайтов, которые надо делать **быстро**, а **поддерживать** долгие годы.

БЭМ (Блок, Элемент, Модификатор) позволяет создавать **расширяемые и повторно используемые независимые компоненты** интерфейса.

<https://techrocks.ru/2021/01/14/bem-naming-methodology/>

<https://htmlacademy.ru/blog/boost/frontend/short-5>

<https://ru.bem.info/methodology/>

Методология БЭМ

Блок

Функционально независимый компонент страницы, который может быть повторно использован.

```
<!-- Блок header -->  
<header class="header">  
  <!-- Вложенный блок logo -->  
  <div class="logo"></div>  
  <!-- Вложенный блок search-form -->  
  <form class="search-form"></form>  
</header>
```


Методология БЭМ

Элемент

Составная часть блока, которая не может использоваться в отрыве от него.

```
<!-- Блок search-form -->  
<form class="search-form">  
  <!-- Элемент input блока search-form -->  
  <input class="search-form__input">  
  <!-- Элемент button блока search-form -->  
  <button class="search-form__button">Найти</button>  
</form>
```

Методология БЭМ

Модификатор

Сущность, определяющая внешний вид, состояние или поведение блока либо элемента.

```
<!-- Блок search-form имеет булевый модификатор `focused` -->
<form class="search-form search-form_focused">
  <input class="search-form__input">
    <!-- Элемент button имеет модификатор `size` со значением `m` -->
    <button class="search-form__button search-form__button_size_m">Найти</button>
  </input>
</form>
```

Методология БЭМ

Упрощение кода и облегчение рефакторинга

```
<ul class="nav">
  <li class="item active">
    <a class="link">One</a>
  </li>
  <li class="item">
    <a class="link">Two</a>
  </li>
  <li class="item">
    <a class="link">Three</a>
  </li>
</ul>
```

```
.item {
  padding: 4px 10px;
  color: black;
}

.active {
  font-weight: bold;
  background: #ffc7c7;
}
```

Методология БЭМ

Упрощение кода и облегчение рефакторинга

```
<ul class="nav">
  <li class="nav__item nav__item_active">
    <a class="nav__link">One</a>
  </li>
  <li class="nav__item">
    <a class="nav__link">Two</a>
  </li>
  <li class="nav__item">
    <a class="nav__link">Three</a>
  </li>
</ul>
```

```
.nav__item {
  padding: 4px 10px;
  color: black;
}

.nav__item_active {
  font-weight: bold;
  background: #ffc7c7;
}
```


Методология БЭМ

Самодокументируемый код

```
form { }  
  
input {  
    background: red;  
}  
  
input[type=submit] {  
    background: #2196f3;  
}
```

```
.form {}  
  
.form_search {}  
  
.form__field {}  
  
.form__submit-button {}
```

Методология БЭМ

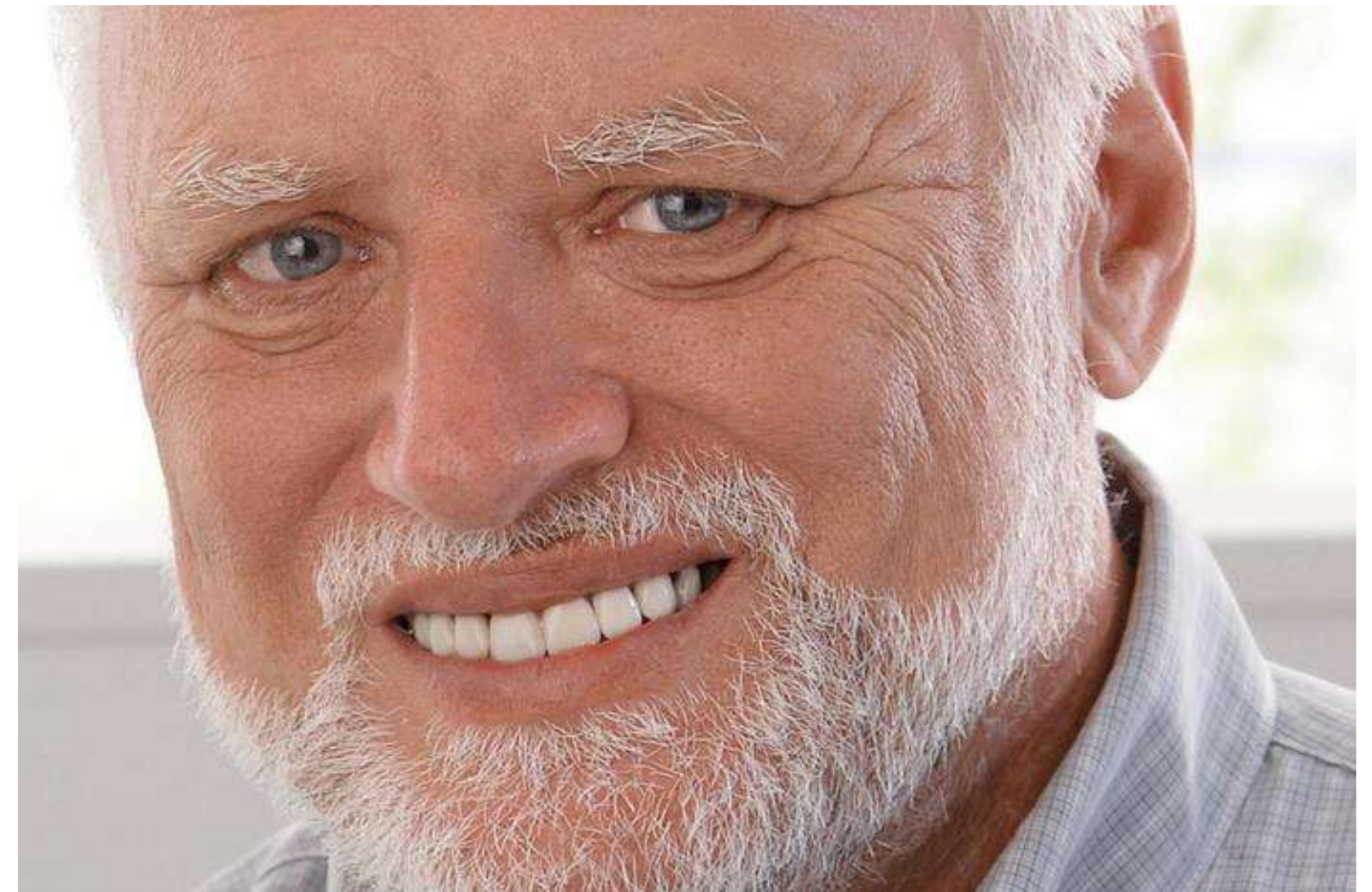
- Новые разработчики могут быстро понять связь между компонентами в разметке и CSS
- Методология способствует повышению производительности в команде
Преимущества особенно заметны в крупных проектах
- Система именования снижает риски коллизий с классами и утечку стилей
- CSS не сильно привязан к разметке в определенном месте на странице
- CSS становится повторно используемым

Методология БЭМ

```
.block {...}  
  
.block__element1 {...}  
  
.block__element1_mod1 {...}  
.block__element1_mod2 {...}  
  
.block__element2 {...}  
.block__element2_mod {...}  
  
.block__element3 {...}
```

Методология БЭМ

```
.block {...}  
  
.block__element1 {...}  
  
.block__element1_mod1 {...}  
.block__element1_mod2 {...}  
  
.block__element2 {...}  
.block__element2_mod {...}  
  
.block__element3 {...}
```



Препроцессоры



Препроцессоры

```
.block {...}

.block__element1 {...}

.block__element1_mod1 {...}

.block__element1_mod2 {...}

.block__element2 {...}

.block__element2_mod {...}

.block__element3 {...}
```

```
.block {
    &__element1 {
        &_mod1 {...}
        &_mod2 {...}
    }

    &__element2 {
        &_mod {...}
    }

    &__element3 {...}
}
```

Препроцессоры

Препроцессор — это программа, которая берёт один тип данных и преобразует его в другой тип данных.

The logo for Sass, a CSS preprocessor. It features the word "Sass" in a stylized, cursive script. The "S" is a large, flowing loop, and the rest of the letters are connected in a fluid, handwritten style. The color is a vibrant magenta or pink.The logo for Stylus, a CSS preprocessor. It features the word "stylus" in a lowercase, serif font. The "st" is in a dark grey color, and the "ylus" is in a light green color. The letters are connected in a fluid, cursive style.

Препроцессоры

- Переменные
- Вложенность
- Операции
- Импорт стилей
- Примеси (mix-in)



[Less. Путеводитель для новичков](#)

lesscss.org

Препроцессоры

Переменные

```
.title {  
    color: #333333;  
}  
  
.box {  
    box-shadow: 0 0 5px #fefefe;  
}
```

```
@mainColor: #1d365d;  
@textColor: #333333;  
@shadowColor: #fefefe;  
  
.title {  
    color: @textColor;  
}  
  
.box {  
    box-shadow: 0 0 5px @shadowColor;  
}
```


Препроцессоры

Вложенность

```
.header h1 {  
    font-size: 26px;  
    font-weight: bold;  
}  
.header p {  
    font-size: 12px;  
}  
.header p a {  
    text-decoration: none;  
}  
.header p a:hover {  
    border-width: 1px;  
}
```

```
.header {  
    h1 {  
        font-size: 26px;  
        font-weight: bold;  
    }  
    p {  
        font-size: 12px;  
        a {  
            text-decoration: none;  
            &:hover {  
                border-width: 1px;  
            }  
        }  
    }  
}
```

Препроцессоры

Вложенность

```
@media screen {  
  .screen-color {  
    color: green;  
  }  
}  
@media screen and (min-width: 768px) {  
  .screen-color {  
    color: red;  
  }  
}  
@media tv {  
  .screen-color {  
    color: black;  
  }  
}
```

```
.screen-color {  
  @media screen {  
    color: green;  
    @media (min-width: 768px) {  
      color: red;  
    }  
  }  
  @media tv {  
    color: black;  
  }  
}
```

Препроцессоры

Вложенность

```
.block {...}

.block__element1 {...}

.block__element1_mod1 {...}

.block__element1_mod2 {...}

.block__element2 {...}

.block__element2_mod {...}

.block__element3 {...}
```

```
.block {
    &__element1 {
        &_mod1 {...}
        &_mod2 {...}
    }

    &__element2 {
        &_mod {...}
    }

    &__element3 {...}
}
```

Препроцессоры

Операции

```
padding-top: 10px + 20; // = 30px  
padding-bottom: 100px - 50; // = 50px  
font-size: 2em * 2; // = 4em  
left: 50% / 2; // 25%
```

Препроцессоры

Примеси (миксины)

```
.text-big() {  
    font-weight: @font-light;  
    color: @color-gray-dark;  
    font-size: 15px;  
    line-height: 20px;  
}  
  
.some-element {  
    .text-big();  
}
```

```
.some-element {  
    font-weight: @font-light;  
    color: @color-gray-dark;  
    font-size: 15px;  
    line-height: 20px;  
}
```


Препроцессоры

Параметрические миксины

```
.font (@size, @line-height, @weight, @color) {  
    font-weight: @weight;  
    font-size: @size;  
    line-height: @line-height;  
    color: @color;  
}  
  
.some-element {  
    .font(15px, 1, 500, black);  
}
```

```
.some-element {  
    font-weight: 500;  
    color: black;  
    font-size: 15px;  
    line-height: 1;  
}
```

Самостоятельное изучение

- Flexbox

Самостоятельное изучение

- Flexbox
- Медиавыражения

Самостоятельное изучение

- Flexbox
- Медиавыражения
- Адаптивная верстка

Самостоятельное изучение

- Flexbox
- Медиавыражения
- Адаптивная верстка
- CSS Трансформации

Самостоятельное изучение

- Flexbox
- Медиавыражения
- Адаптивная верстка
- CSS Трансформации
- CSS Анимации

Самостоятельное изучение

- Flexbox
- Медиавыражения
- Адаптивная верстка
- CSS Трансформации
- CSS Анимации
- Float

Самостоятельное изучение

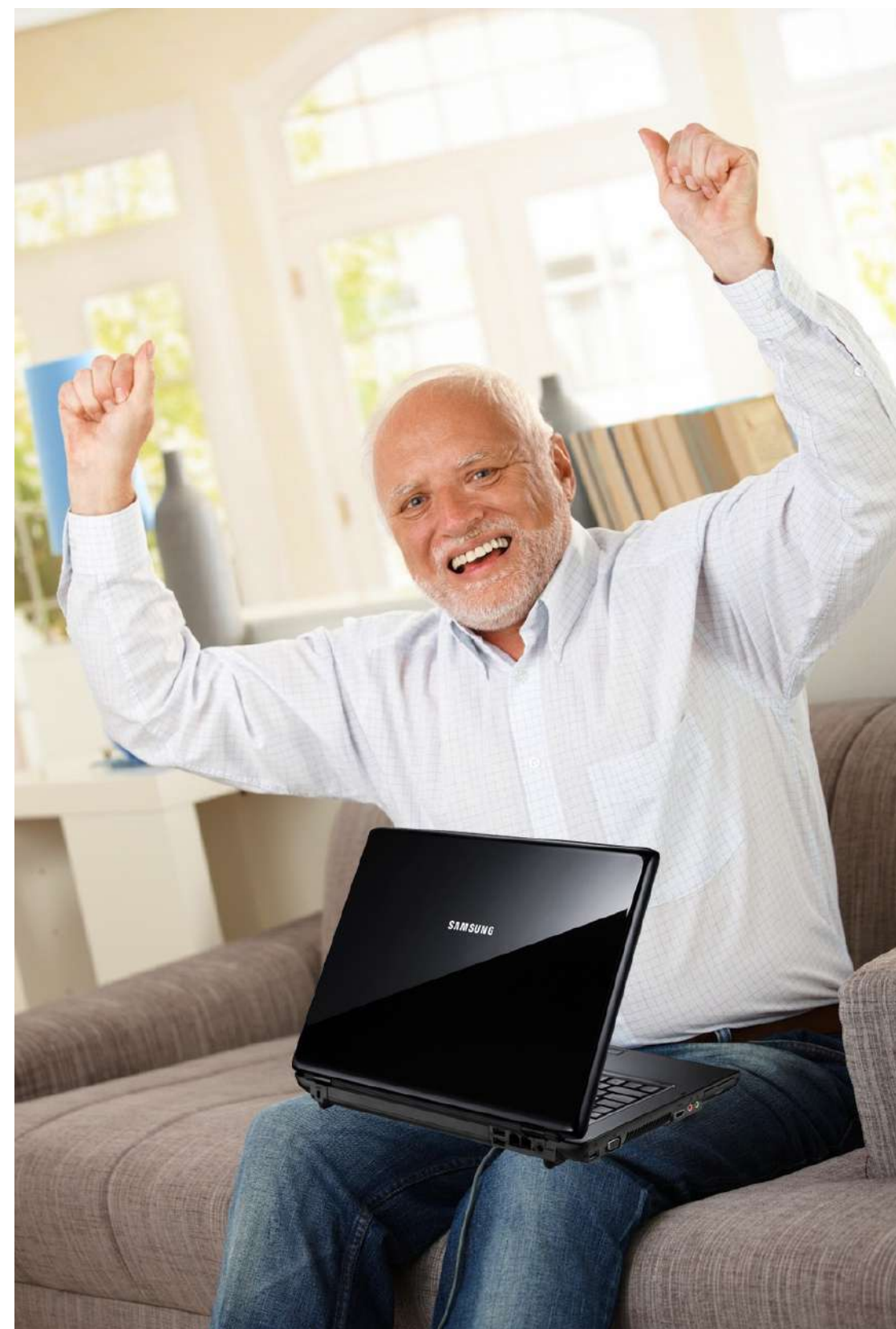
- Flexbox
- Медиавыражения
- Адаптивная верстка
- CSS Трансформации
- CSS Анимации
- Float
- CSS Grid

Примеры для д/з

- [Поле поиска](#)
- [Поле поиска при скролле](#)
- [Список карточек](#)
- [Состояния карточек](#)
- [Центрирование содержимого](#)

Обратите внимание, что в макетах справа (6-element) есть состояние карточки при наведении курсора мыши, которое необходимо доработать самостоятельно, в примере выше оно не реализовано.





Спасибо!



ТИНЬКОФФ