



КОМПЛЕКС
СОЦИАЛЬНОГО
РАЗВИТИЯ
МОСКВЫ



Моя
профессия

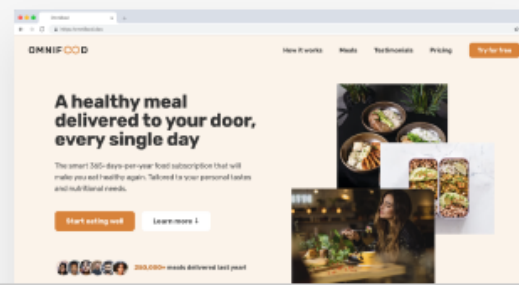
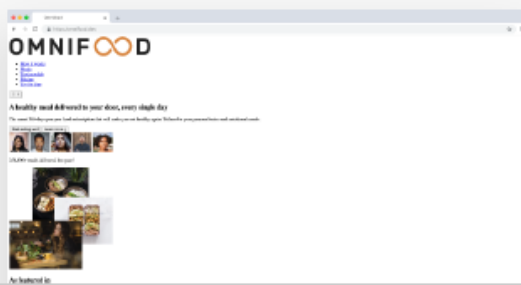
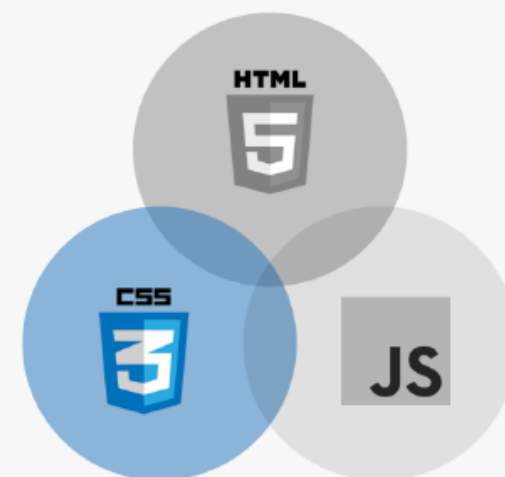
Введение в CSS

Синтаксис, селекторы, свойства, анимация, Flexbox, CSS GRID

Что такое CSS ?

CSS

- 👉 Cascading **S**tyle **S**heets
- 👉 **CSS** задает визуальное и стилевое оформление контенту, написанному на HTML
- 👉 CSS состоит из бесчисленного множества свойств, которые разработчики используют для форматирования контента



Как выбрать и стилизовать элемент в CSS

A CSS RULE

Селектор

Блок свойств

```
h1 {  
  color: blue;  
  text-align: center;  
  font-size: 20px;  
}
```

Свойство

Название
свойства

Значение

Внедрение стилей в документ

```
<!DOCTYPE html>
<html>
  <head> <!-- голова документа -->
    <meta charset="utf-8">
    <title>Document</title>
    <!-- блок для задания стилей -->
    <style>
      h1{
        color: ■ orangered;
      }
    </style>
  </head>
  <body>
    <h1>Hello, world!!!</h1>
  </body>
</html>
```

Стили можно задавать не только
в атрибуте элемента

В голове документа в теге `<style>`
можно задавать стили
элементам текущей страницы

Подключение стилей отдельным файлом

```
<!DOCTYPE html>
<html>
  <head> <!-- голова документа -->
    <meta charset="utf-8">
    <title>Document</title>

    <!-- подключаем файл со стилями -->
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
  </body>
</html>
```

Файл index.html

```
/* задаем стили заголовку 1 уровня */
h1{
  color: ■ aqua;
  background-color: ■ black;
  font-size: 25px;
  text-align: center;
  border: 3px solid ■ brown;
}
```

Файл style.css

Селекторы в CSS

Для того чтобы задать стиль HTML элементу, его нужно сначала выбрать

Выбрать элемент можно по названию тега, по имени класса или идентификатора

Перед названием класса в CSS добавляется точка, перед идентификатором - решетка

```
<!DOCTYPE html>
<html>
  <head> <!-- голова документа -->
    <meta charset="utf-8">
    <title>Document</title>

    <!-- подключаем файл со стилями -->
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1 class="hello" id="world">Hello, world!!!</h1>
  </body>
</html>
```

```
/* задаем стили заголовку 1 уровня */
h1{
  color: aqua;
}

.hello{
  background-color: black;
  font-size: 25px;
}

#world{
  text-align: center;
  border: 3px solid brown;
}
```

Цвет текста в CSS

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>
```

Файл index.html

```
h1{
  color: aqua;
}

h2{
  color: rebeccapurple;
}

p{
  color: firebrick;
}
```

Файл style.css

Цвет фона в CSS

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>
```

Файл index.html

```
h1{
  background-color: saddlebrown;
}

h2{
  background-color: palegoldenrod;
}

p{
  background-color: darkkhaki;
}
```

Файл style.css

Границы элемента в CSS

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>

```

Файл index.html

```

h1{
  border: 3px solid #rosybrown;
}

h2{
  border: 2px dotted #gainsboro;
}

p{
  border: 4px dashed #yellowgreen;
}

```

Файл style.css

margin - внешние отступы в CSS

Для того чтобы добавить отступы между элементами в CSS, используется свойство **margin**. Отступы можно указать одним значением для всех сторон, а можно каждой стороне по отдельности, начиная сверху и по часовой стрелке.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>
```

```
h1{
  margin: 30px;
}

h2{
  margin: 20px 30px 25px 15px;
}

p{
  margin-top: 20px;
  margin-right: 30px;
  margin-bottom: 25px;
  margin-left: 15px;
}
```

padding - внутренние отступы в CSS

Для того, чтобы добавить внутренние отступы между контентом и границей в CSS используется свойство **padding**. Отступы можно указать одним значением для всех сторон, а можно каждой стороне по отдельности, начиная сверху и по часовой

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>
```

```
h1{
  padding: 14px;
}

h2{
  padding: 15px 20px 17px 10px;
}

p{
  padding-top: 15px;
  padding-right: 20px;
  padding-bottom: 17px;
  padding-left: 10px;
}
```

Ширина и высота элементов в CSS

Для того чтобы задать ширину элементу, используется свойство width.

Для того чтобы задать высоту элементу, используется свойство height.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello, world!!!</h1>
    <h2>Temporibus dolores aperiam quos.</h2>
    <p>Lorem ipsum dolor sit, amet consectetur
    adipiscing elit. Amet veritatis incidunt
    dolor aspernatur nobis! Veniam quod.
    </p>
  </body>
</html>
```

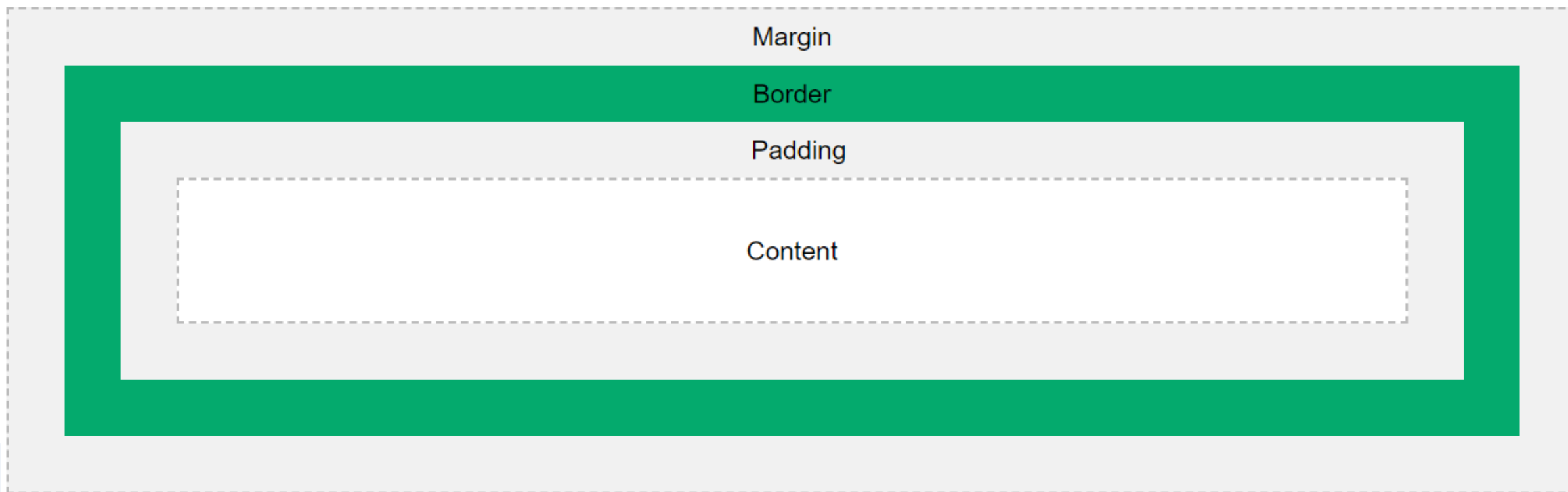
```
h1{
  width: 300px;
  height: 400px;
}

h2{
  width: 500px;
  height: 200px;
}

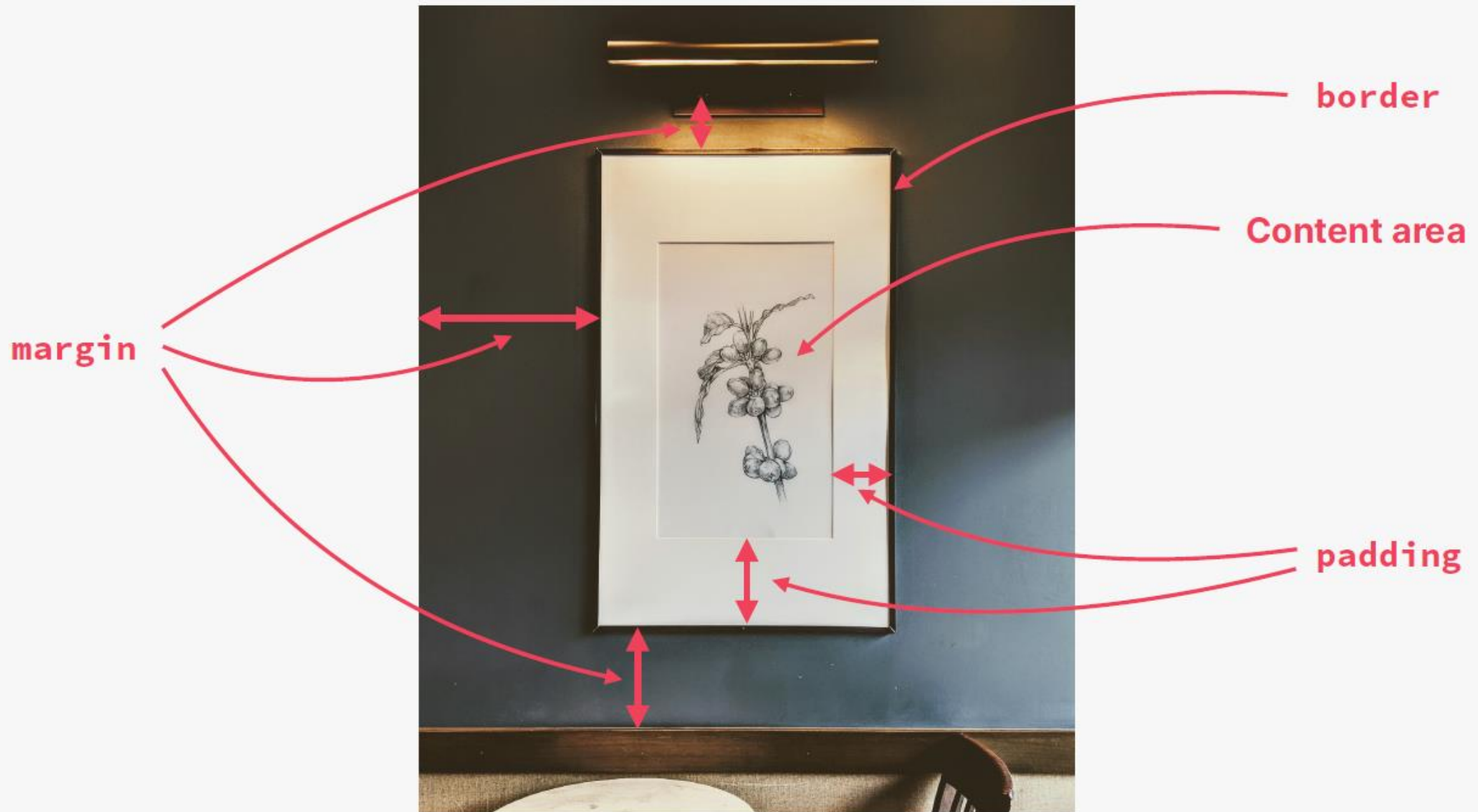
p{
  width: 400px;
  height: 600px;
}
```

Блочная модель в CSS

Свойства **width**, **height**, **padding**, **border**, **margin** формируют отображение элемента



Аналогия блочной модели в CSS



Блочные элементы

- 👉 Элементы отображаются как блоки
- 👉 Элемент занимает 100% от ширины родительского элемента
- 👉 Элементы отображаются друг за другом
- 👉 Элементам можно задать ширину и высоту

Default elements: body, main, header, footer, section, nav, aside, div, h1-h6, p, ul, ol, li, etc.

With CSS: display: block

The Basic Language of the Web: HTML



Posted by **Laura Jones** on Monday, June 21st 2027

```
    <!--HEADER BOXED FONT WHITE TRANSPARENT-->
    <div class="header-black-bg"></div>
    <!--NEED FOR TRANSPARENT HEADER ON MOBILE-->
    <header id="nav" class="header header-1 header-boxed header-transparent header-transparent-header">
    <!--FEATURES 7 HALF IMG-->
    <div class="page-section bg-gray-light clearfix">
      ::before
      <div class="fes7-img-cont col-md-5">
        <div class="fes7-img" style="background-image: url('img/fes7-img.jpg');">
        </div>
      <div class="container">
      ::after
    </div>
```

All modern websites and web applications are built using three *fundamental* technologies: HTML, CSS and JavaScript. These are the languages of the web.

In this post, let's focus on HTML. We will learn what HTML is all about, and why should learn it.

What is HTML?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam recusandae reprehenderit vitae ratione veritatis corrupti sit ut vero, dolores nulla exercitationem eos quod iusto incidunt, perferendis alias tenetur. Est, vel!

In HTML, each element is made up of 3 parts:

1. **The opening tag**
2. **The closing tag**
3. **The actual element**

You can learn more at the [MDN Web Docs](#).

Строчные элементы

👉 Занимают место необходимое для контента

👉 Нет свойств ширины и высоты

👉 Не переносятся на новую строку, идут друг за другом

👉 padding и margin применяются только в горизонтальной оси

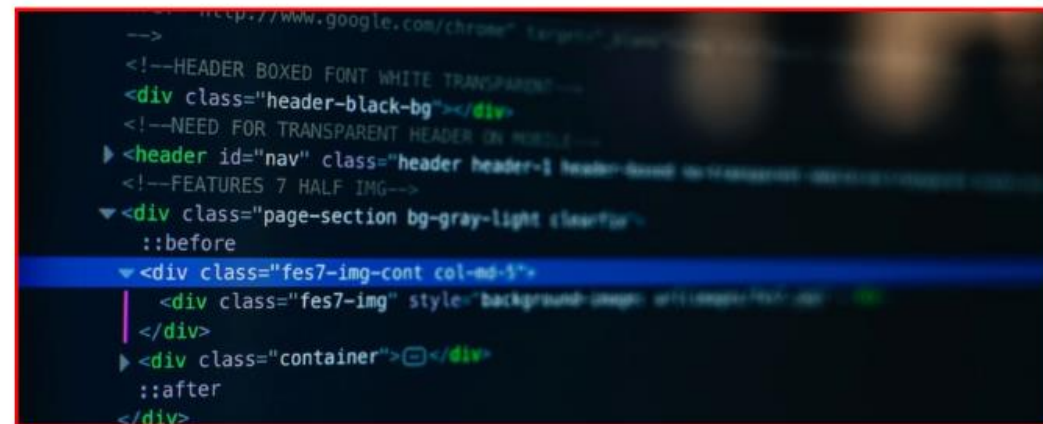
Default elements: a, img, strong, em, button, etc.

With CSS: display: inline

The Basic Language of the Web: HTML



Posted by [Laura Jones](#) on Monday, June 21st 2027



All modern websites and web applications are built using three [fundamental](#) technologies: HTML, CSS and JavaScript. These are the languages of the web.

In this post, let's focus on HTML. We will learn what HTML is all about, and why should learn it.

What is HTML?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam recusandae reprehenderit vitae ratione veritatis corrupti sit ut vero, dolores nulla exercitationem eos quod iusto incidunt, perferendis alias tenetur. Est, vel!

In HTML, each element is made up of 3 parts:

1. The opening tag
2. The closing tag
3. The actual element

You can learn more at the [MDN Web Docs](#).

Стили для форматирования текста

```
h1{  
  text-align: center; /* center / left / right / justify */  
  text-decoration: overline; /* overline / line-through / underline */  
  text-transform: uppercase; /* uppercase / lowercase / capitalize */  
  letter-spacing: 5px;  
}
```

Text-align: расположение текста по горизонтали

Text-decoration: линия вдоль текста

Text-transform: меняет регистр текста





Letter-spacing: расстояние между символами

Шрифты в CSS

```
h1{
  font-family: "Times New Roman", Times, serif;
  font-style: normal;
  font-weight: normal;
  font-size: 40px;
}

h2{
  font-family: Arial, Helvetica, sans-serif;
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
}

p{
  font-family: "Lucida Console", "Courier New", monospace;
  font-style: italic;
  font-weight: normal;
  font-size: 20px;
}
```

```
a:link{  
    color:  orange;  
}/* непосещенная ссылка */  
  
a:visited{  
    color:  black;  
}/* посещенная ссылка */  
  
a:hover{  
    color:  orangered;  
}/* стили при наведении мыши */  
  
a:active{  
    color:  yellow;  
}/* стили в момент клика по ссылке */
```

```
ul.a {  
  list-style-type: circle;  
}  
  
ul.b {  
  list-style-type: square;  
}  
  
ol.c {  
  list-style-type: upper-roman;  
}  
  
ol.d {  
  list-style-type: lower-alpha;  
}
```

List-style-type задает маркер
около каждого элемента
списка

Свойство display

```
h1{  
  display: block;  
  /*  
    block / inline / inline-block / flex  
    grid / none  
  */  
}
```

Inline-block имеет ширину, высоту, отступы и поля, а также их можно расположить рядом друг с другом

Flex и grid изучим дальше по курсу

Абсолютное позиционирование



Переходы в CSS

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Transition</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div></div>
</body>
</html>

```

```

div {
  width: 100px;
  height: 100px;
  transition: width 2s, height 4s;
  /*
   *   при наведении мыши
   *   меняется высота и ширина элемента
   */
}

div:hover {
  width: 300px;
  height: 300px;
}

```

Анимация в CSS

Анимация позволяет менять стили элемента одни на другие.
Вы можете изменить столько CSS свойств, сколько захотите.

```
• • •
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Animation</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div></div>
</body>
</html>
```

```
• • •
div {
  width: 100px;
  height: 100px;
  background-color: ■ red;
  animation-name: example; /*имя*/
  animation-duration: 4s; /*длительность*/
}

/* анимация */
@keyframes example {
  /*меняет фон с красного*/
  from {background-color: ■ red;}
  /*на желтый*/
  to {background-color: ■ yellow;}
}
```


Анимация по кадрам в CSS

Анимацию можно расписать по кадрам: на каждом кадре можно задать нужные свойства

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Animation</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div></div>
</body>
</html>
```

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example; /*имя*/
  animation-duration: 4s; /*длительность*/
}

@keyframes example {
  /*4 кадра анимации, в каждом
  меняется цвет фона*/
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  75%  {background-color: green;}
  100% {background-color: red;}
}
```

Анимация по кадрам в CSS

Анимировать можно расположение элемента, то есть двигать его по странице

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Animation</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div></div>
</body>
</html>
```

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red; left:0px; top:0px;}
  25%  {background-color: yellow; left:200px; top:0px;}
  50%  {background-color: blue; left:200px; top:200px;}
  75%  {background-color: green; left:0px; top:200px;}
  100% {background-color: red; left:0px; top:0px;}
}
```

Анимация по кадрам в CSS

Анимацию можно повторять столько раз, сколько нужно, даже бесконечно

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Animation</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div></div>
</body>
</html>
```

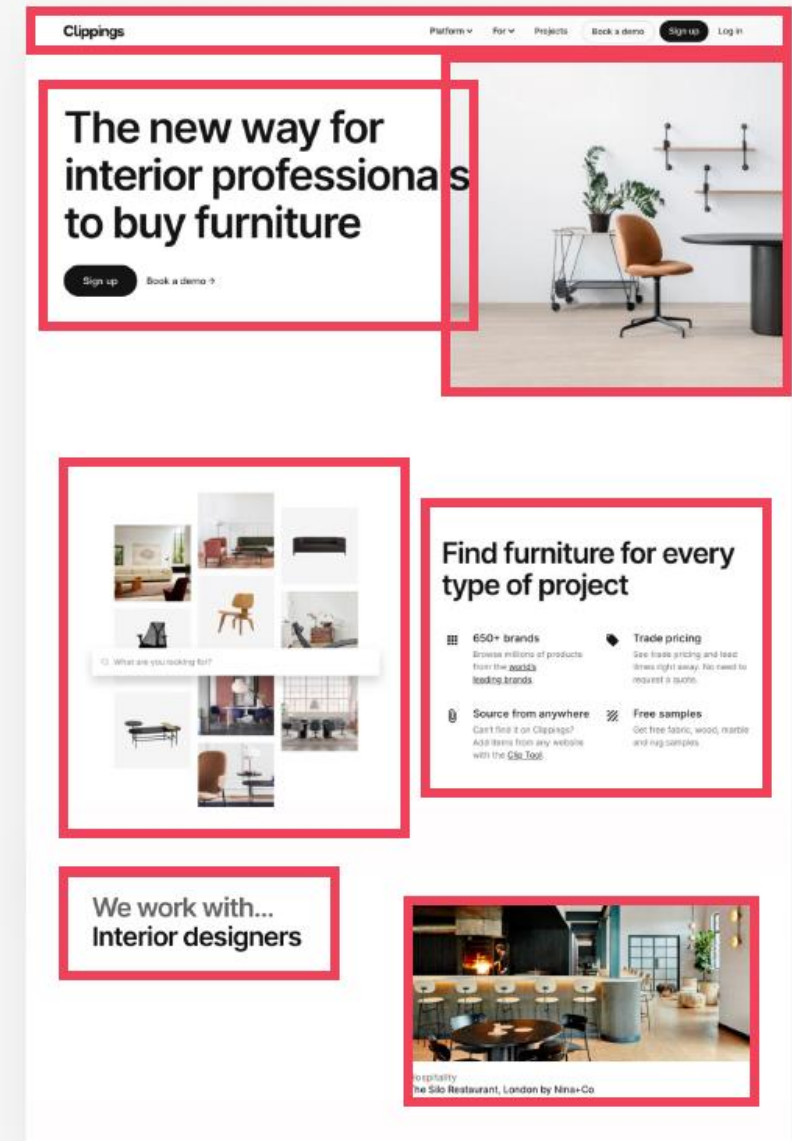
```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3; /*infinite-бесконечно*/
}

@keyframes example {
  0%   {background-color: red; left:0px; top:0px;}
  25%  {background-color: yellow; left:200px; top:0px;}
  50%  {background-color: blue; left:200px; top:200px;}
  75%  {background-color: green; left:0px; top:200px;}
  100% {background-color: red; left:0px; top:0px;}
}
```

Макет сайта

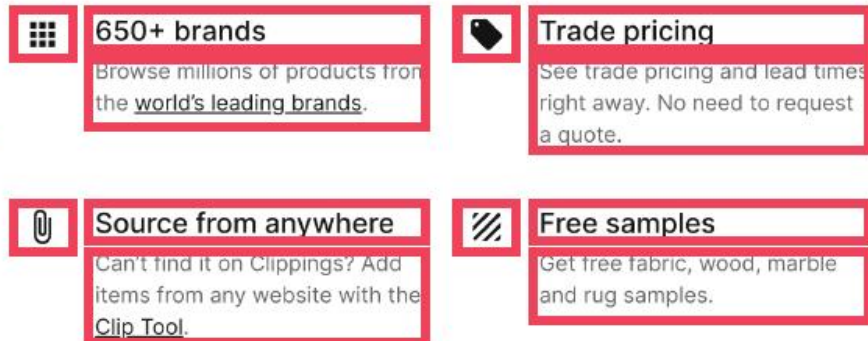
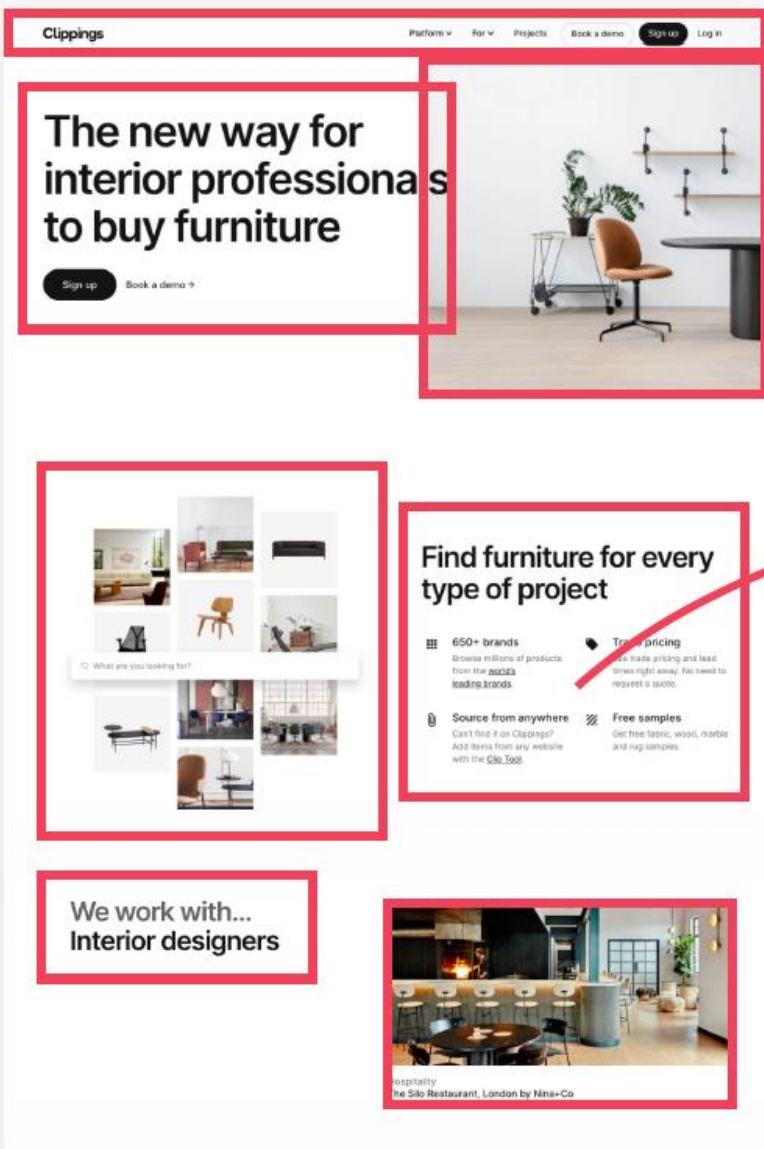
LAYOUT

- 👉 **Макет** - это то, каким образом текст, картинки и другой контент расположены на странице
- 👉 **Макет** дает странице визуальную структуру, в которой мы располагаем контент
- 👉 **Построение макета** - компоновка элементов на странице, вместо расположения в столбик один за другим



Макет сайта VS Макет компонента

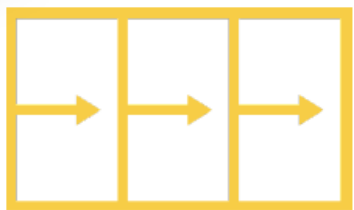
PAGE LAYOUT



COMPONENT LAYOUT

Три способа построить макет в CSS

1



FLOAT LAYOUTS

Старый способ создания макетов, использует CSS свойство **float**. Еще встречается, но быстро устаревает.

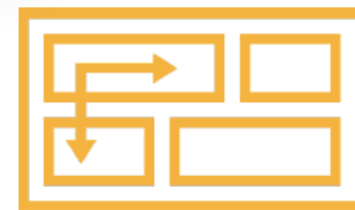
2



FLEXBOX

Современный способ выравнивания элементов в один ряд. Идеален для **макетов компонентов**.

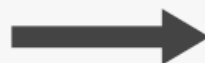
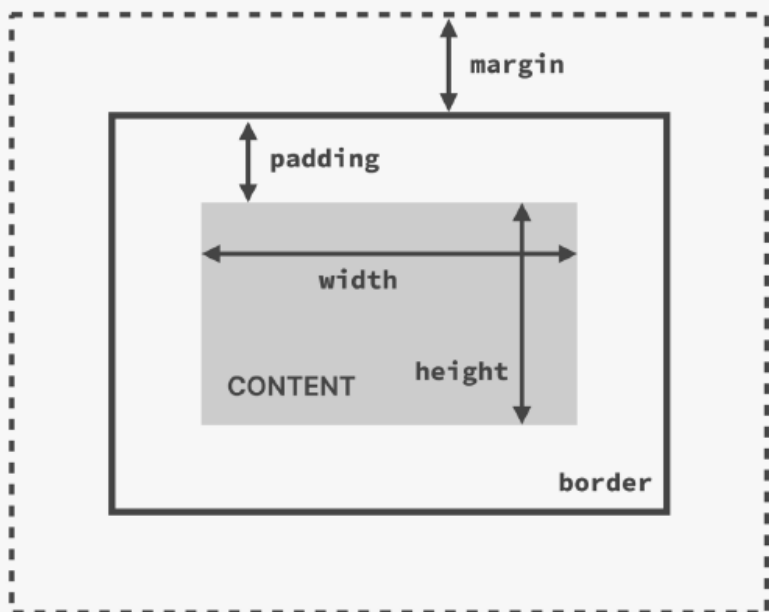
3



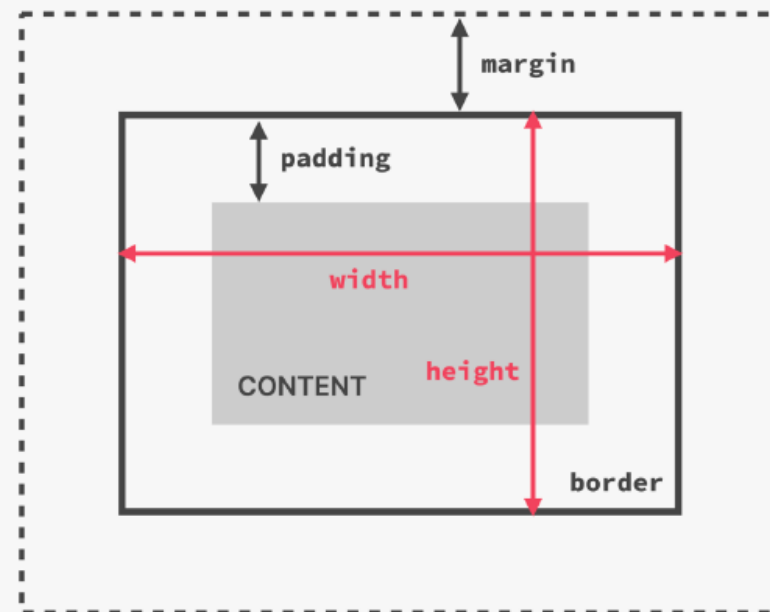
CSS GRID

Используется для постройки сетки более одного ряда. Идеален для **макетов сайтов** и **сложных компонентов**.

Блочная модель с: **box-sizing: border-box**



box-sizing: border-box



Final element width = ~~right border~~ + ~~right padding~~ + **width** + ~~left padding~~ + ~~left border~~

Final element height = ~~top border~~ + ~~top padding~~ + **height** + ~~bottom padding~~ + ~~bottom border~~

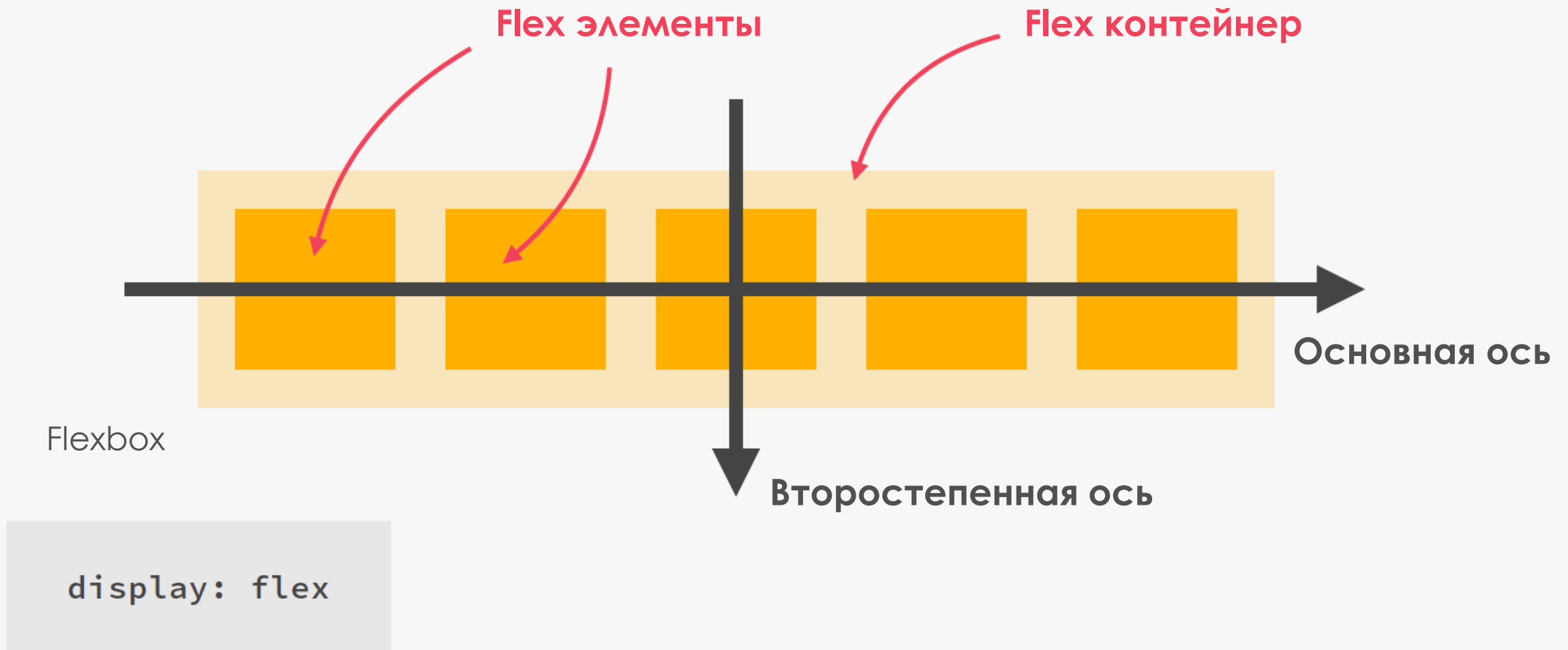
Что такое flexbox?

FLEXBOX

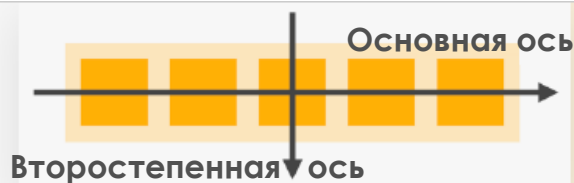


- 👉 Flexbox это набор связанных **CSS свойств**
- 👉 Основная идея flexbox в том, чтобы **распределить** место в родительском элементе между дочерними
- 👉 Flexbox **упрощает выравнивание** дочерних элементов в родительском как горизонтально, так и вертикально
- 👉 Flexbox решает проблему **вертикального выравнивания** элементов и создания блоков **одинаковой высоты**
- 👉 Flexbox идеален для **замены float**, позволяет нам писать более чистый и компактный код

Терминология **flexbox**



Flex контейнер



Flex элементы

- 1 `gap: 0 | <length>`
👉 Создает **промежутки** между элементами
- 2 `justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly`
👉 Выравнивает элементов по **ОСНОВНОЙ** оси
- 3 `align-items: stretch | flex-start | flex-end | center | baseline`
👉 Выравнивает по **ВТОРОСТЕПЕННОЙ** оси
- 4 `flex-direction: row | row-reverse | column | column-reverse`
👉 **Выбирает** главную ось
- 5 `flex-wrap: nowrap | wrap | wrap-reverse`
👉 Регулирует перенос элементов на **новую** строку
- 6 `align-content: stretch | flex-start | flex-end | center | space-between | space-around`
👉 Выравнивает ряды, только если **более одного** ряда

- 1 `align-self: auto | stretch | flex-start | flex-end | center | baseline`
👉 Перезаписывает **align-items** элемента
- 2 `flex-grow: 0 | <integer>`
👉 Позволяет элементу **увеличиваться** (0 или 1)
- 3 `flex-shrink: 1 | <integer>`
👉 Позволяет элементу **уменьшаться** (0 или 1)
- 4 `flex-basis: auto | <length>`
👉 Задаёт **ширину** элемента (вместо width)
- 5 `flex: 0 1 auto | <int> <int> <len>`
👉 Короткая запись **flex-grow, flex-shrink, flex-basis**
- 6 `order: 0 | <integer>`
👉 Контролирует **порядок** элементов

Что такое CSS GRID?

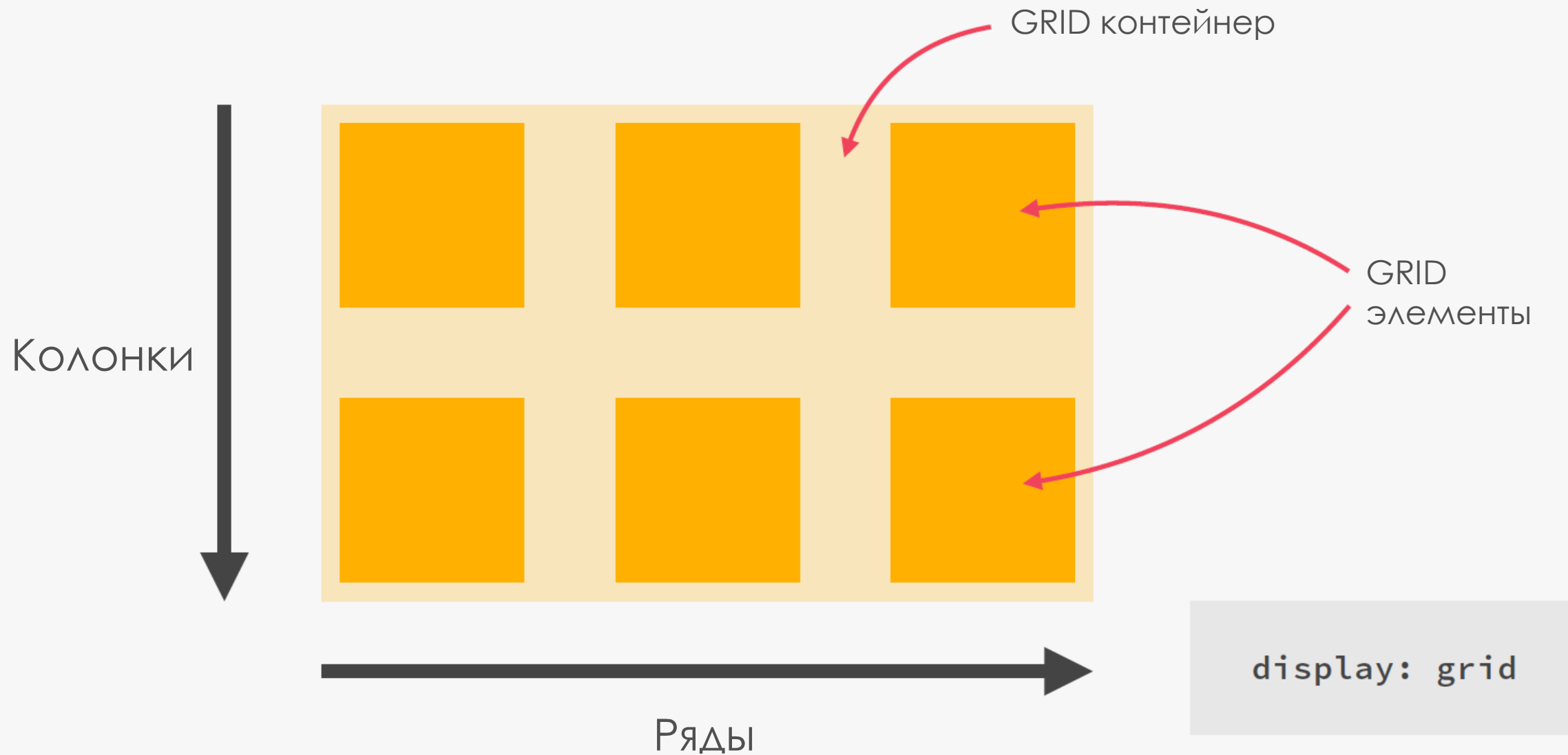
CSS GRID



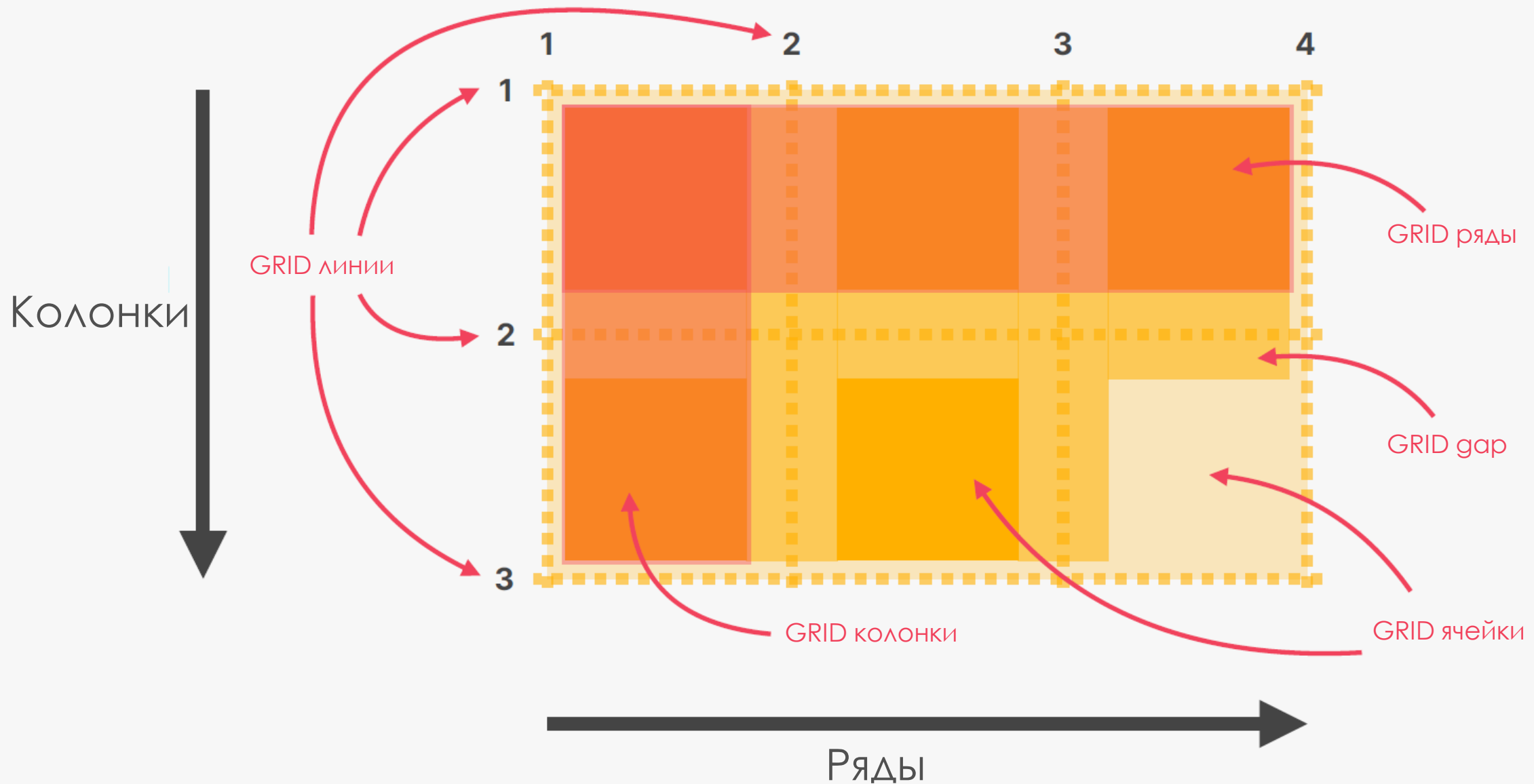
- 👉 CSS GRID - это набор связанных **CSS свойств** для постройки сложных макетов
- 👉 Основная идея CSS GRID в том, чтобы **разделить контейнер на ряды и колонки** и заполнить дочерними элементами
- 👉 CSS GRID позволяет писать **меньше вложенного HTML** кода и легче читать CSS код
- 👉 CSS GRID **не заменяет** Flexbox, они работают вместе. Нужен один ряд элементов используй Flexbox, несколько - CSS GRID



Базовая терминология CSS GRID



Базовая терминология CSS GRID



GRID контейнер

1 `grid-template-rows: <track size>*`
`grid-template-columns: <track size>*`

👉 Задаёт ряды и колонки. Каждому юниту задаётся свое значение

2 `row-gap: 0 | <length>`
`column-gap: 0 | <length>`] `gap: 0 | <length>`

👉 Создает пустое пространство

3 `justify-items: stretch | start | center | end`
`align-items: stretch | start | center | end`

👉 Выравнивает элементы в рядах/колонках

4 `justify-content: start | start | center | end | ...`
`align-content: start | start | center | end | ...`

👉 Выравнивает сетку внутри контейнера, если контейнер больше сетки

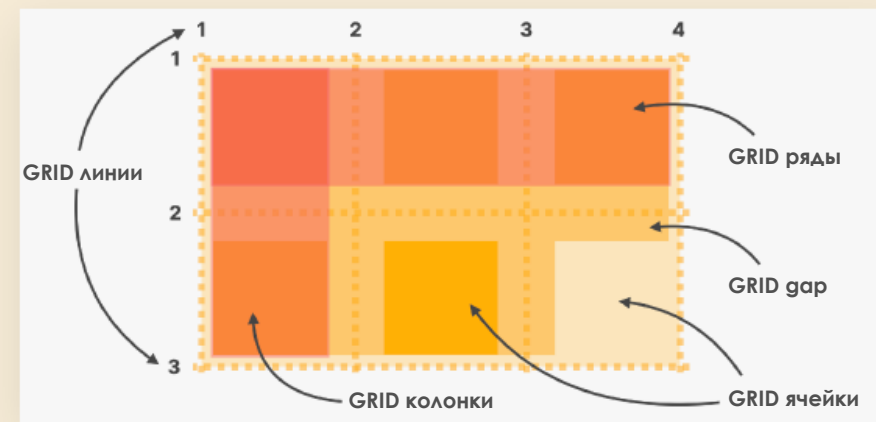
GRID элементы

1 `grid-column: <start line> / <end line> | span <number>`
`grid-row: <start line> / <end line> | span <number>`

👉 Задаёт место элементу в ячейке сетки.
Span регулирует количество ячеек

2 `justify-self: stretch | start | center | end`
`align-self: stretch | start | center | end`

👉 Перезаписывает justify-items/align-items элемента



👉 This list of CSS Grid properties is not exhaustive, but enough to get started.