

# **FULL-STACK LABS 2019**

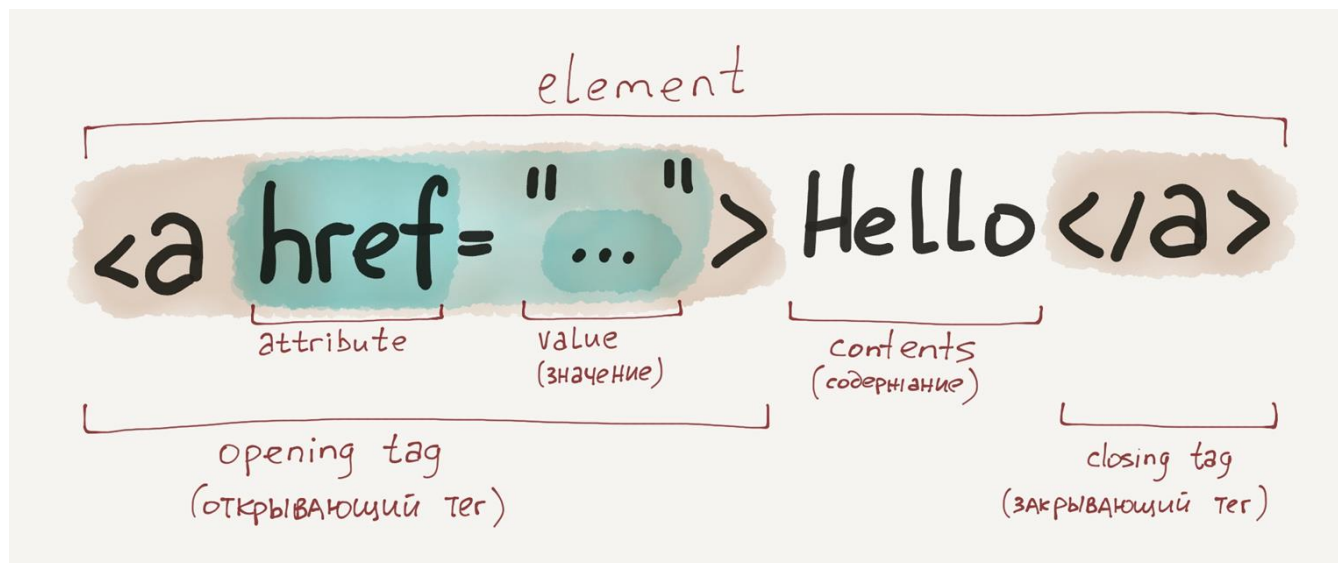
## **HTML CSS**



**iTechArt**

**HTML**

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — язык разметки, определяющий структуру веб-страниц.



```

```

```
<p>Мой кот <strong>очень</strong> сердитый.</p>
```



```
<p>Мой кот <strong>очень сердитый.</p></strong>
```



Элементы блочного уровня формируют видимый блок на странице — они окажутся на новой строке после любого контента, который шел до них, и любой контент после них также окажется на новой строке.

Строчные элементы — это те, которые содержатся в элементах блочного уровня и окружают только малые части содержимого документа, не целые абзацы и группировки контента. Строчные элементы не приводят к появлению новой строки в документе.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>My test page</title>
6    </head>
7    <body>
8      <p>This is my page</p>
9    </body>
10 </html>
```

```
1 <meta name="author" content="Chris Mills">
2 <meta name="description" content="The MDN Learning Area aims to provide
3 complete beginners to the Web with all they need to know to get
4 started with developing web sites and applications.">
```

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

```
1 <meta property="og:image" content="https://developer.cdn.mozilla
2 <meta property="og:description" content="The Mozilla Developer N
3 information about Open Web technologies including HTML, CSS, and
4 and HTML5 Apps. It also documents Mozilla products, like Firefox
5 <meta property="og:title" content="Mozilla Developer Network">
```

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction to HTML/Document and website structure](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction%20to%20HTML/Document%20and%20website%20structure)



## Document and website structure

At this point you should have a better idea about how to structure a web page/site. In the last article

 <https://developer.mozilla.org>

Чтобы добавить на страницу favicon:

1. Сохраните изображение в формате .ico, .gif или .png, в папку со своим документом.
2. Добавьте ссылку на иконку в `<head>` документа. Для разных устройств можно указывать разные иконки.

```
1 <!-- Для iPad 3 с Retina-экраном высокого разрешения: -->
2 <link rel="apple-touch-icon-precomposed" sizes="144x144" href="https://developer.cdn.mozilla.net
3 <!-- Для iPhone с Retina-экраном высокого разрешения: -->
4 <link rel="apple-touch-icon-precomposed" sizes="114x114" href="https://developer.cdn.mozilla.net
5 <!-- Для iPad первого и второго поколения: -->
6 <link rel="apple-touch-icon-precomposed" sizes="72x72" href="https://developer.cdn.mozilla.net/
7 <!-- Для iPhone, iPod Touch без Retina, и устройств с Android 2.1+: -->
8 <link rel="apple-touch-icon-precomposed" href="https://developer.cdn.mozilla.net/static/img/fav
9 <!-- Для других случаев - обычный favicon -->
10 <link rel="shortcut icon" href="https://developer.cdn.mozilla.net/static/img/favicon32.e02854f
```



```
1 | <html lang="en-US">
```

# Включение специальных символов в HTML

10

Буквенный символ	Символьный эквивалент
<	&lt;
>	&gt;
"	&quot;
'	&apos;
&	&amp;

```
1 <p>In HTML, you define a paragraph using the <p> element.</p>
2
3 <p>In HTML, you define a paragraph using the &lt;p&gt; element.</p>
```

**<h1>** ➞ **Animal Farm**

**<h2>** ➞ George Orwell ➞ **<a href="...">**

1945

They explained that by their studies of the past three months the pigs had succeeded in reducing the principles of Animalism to Seven Commandments. These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

**<p> and <strong>**



**<h3>** ➞ **THE SEVEN COMMANDMENTS**

1. Whatever goes upon two legs is an enemy.
2. Whatever goes upon four legs, or has wings, is a friend<sup>1</sup>.
3. No animal shall wear clothes<sub>2</sub>.
4. No animal shall sleep in a bed.
5. No animal shall drink alcohol.
6. No animal shall kill any other animal.
7. All animals are equal.

**<sub> and <sup>**  
**<a href="#note">**

**<ol> - <li>**

It was very neatly written, and except that 'friend' was written 'freind'

**<em>**

and one of the 'S's' was the wrong way round,  
the spelling was correct all the way through.

**<pre>**

**<hr>**



**<h4>** ➞ **Character Set**

Pigs

Old Major  
Napoleon

Humans

Mr Jones  
Mr Frederick

**<dl> - <dt> - <dd>**

# BASIC HTML TAGS

12

**Animal Farm**

Enter your name  **<input>**

Choose the Commandments

- ☐ Whatever goes upon two legs is an enemy.
- ☐ Animal Farm must live for ever after.
- ☐ No animal shall drink alcohol.

**<input type="checkbox">**

Write something  **<textarea>**

Choose one option

- ☐ Old Major
- ☐ Napoleon

Choose a character--  
Choose a character--  
Old Major  
Napoleon  
Mr Jones  
Mr Frederick

**<select>**

Send  **<input type="submit">**

**<input type="radio">**

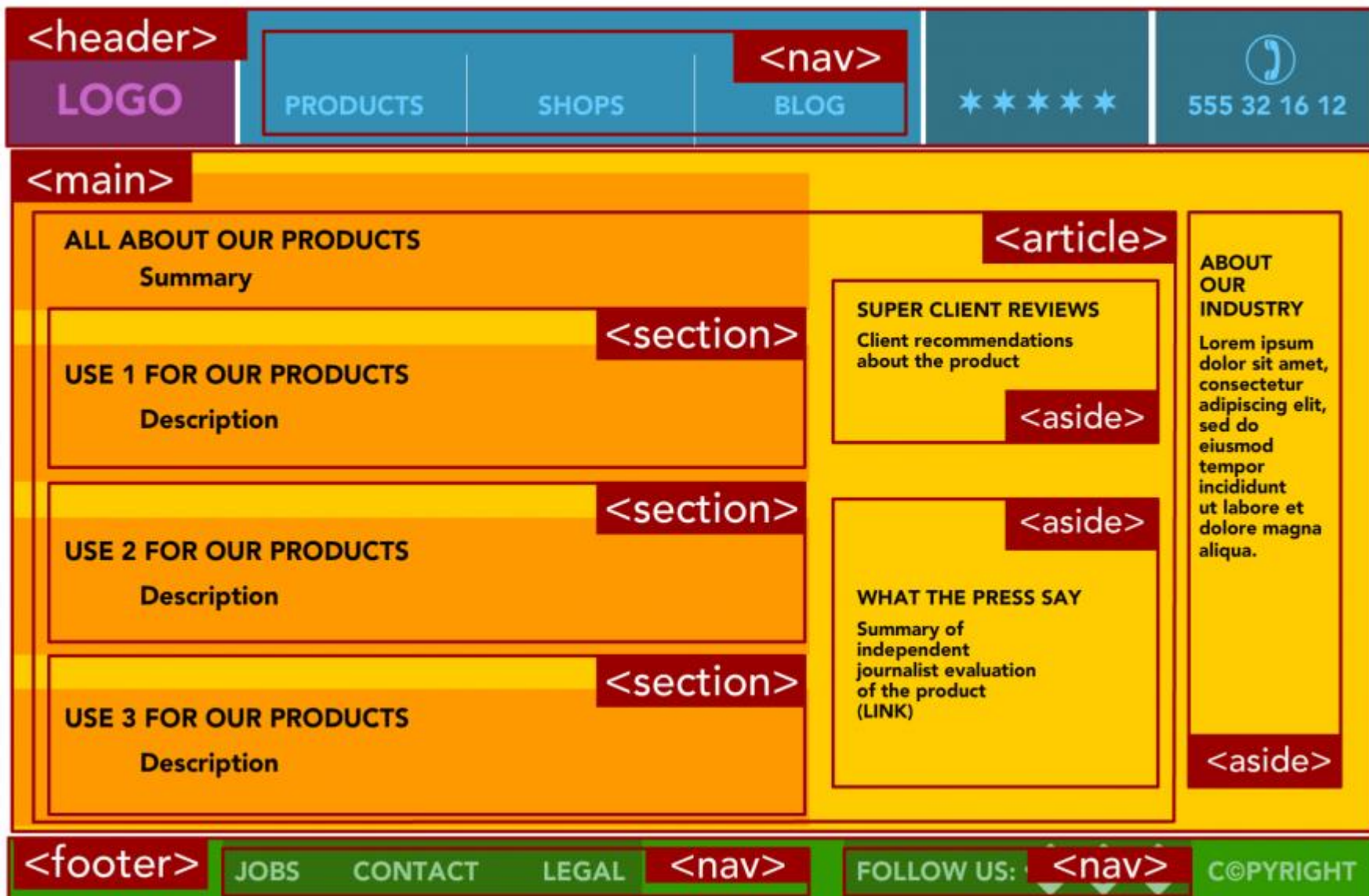
**<img>**

Animal Farm (1954 - Cartoon) George O..

**<iframe>**

**<audio>**

- **Правильность:** валидный документ корректно отображается в браузере.
- **Отладка:** некорректный код HTML может вызвать ошибки, сложные для выявления.
- **Поддержка:** валидный документ легче обновлять позже, даже кому-то другому.



**CSS**

CSS – «каскадные таблицы стилей». Этот язык отвечает за внешний вид HTML-страницы. Синтаксис языка достаточно прост: он состоит из селекторов и свойств.

```
H1 {  
    font-size: 120%;  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
    color: #333366;  
}
```



# **Способы добавления стилей на страницу**

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Внутренние стили</title>
  </head>
  <body>
    <p style="font-size: 120%; font-family: monospace; color:
      #cd66cc">Пример текста</p>
  </body>
</html>
```

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Глобальные стили</title>
    <style>
      H1 {
        font-size: 120%;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #333366;
      }
    </style>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Стили</title>
```

```
    <link rel="stylesheet" href="http://htmlbook.ru/mysite.css">
```

```
    <link rel="stylesheet" href="http://www.htmlbook.ru/main.css">
```

```
  </head>
```

```
  <body>
```

```
    <h1>Заголовок</h1>
```

```
    <p>Текст</p>
```

```
  </body>
```

```
</html>
```

- \* – любые элементы.
- div – элементы с таким тегом.
- #id – элемент с данным id.
- .class – элементы с таким классом.
- [name="value"] – селекторы на атрибут
- :visited – «псевдоклассы», остальные разные условия на элемент (см. далее).
- .c1.c2 –!без пробела! элементы одновременно с двумя классами c1 и c2
- a#id.c1.c2:visited – элемент a с данным id, классами c1 и c2, и псевдоклассом visited
- div p – элементы p, являющиеся потомками div.
- div > p – только непосредственные потомки
- div ~ p – правые соседи: все p на том же уровне вложенности, которые идут после div
- div + p – первый правый сосед: p на том же уровне вложенности, который идёт сразу после
- :not(селектор) – все, кроме подходящих под селектор.
- :focus – в фокусе.
- :hover – под мышью.
- :empty – без детей (даже без текстовых).
- :checked, :disabled, :enabled – состояния INPUT.
- :target – этот фильтр сработает для элемента, ID которого совпадает с анкором #... текущего U

# Основные виды селекторов

21

- :first-child – первый потомок своего родителя.
- :last-child – последний потомок своего родителя.
- :only-child – единственный потомок своего родителя, соседних элементов нет.
- :nth-child(a) – потомок номер a своего родителя, например :nth-child(2) – второй потомок. Нумерация начинается с 1.

**ОЧ ПЛОХО!**

```
#search-results div.search-content ~ div + ul li {  
    margin: 0;  
    list-style-type: square;  
}
```

[attr= val] – атрибут равен val или на начало val, например равен val 1.

На содержание:

- [attr\*= "val"] – атрибут содержит подстроку val, например равен "myvalue".
- [attr~="val"] – атрибут содержит val как одно из значений через пробел.

Например: [attr~="delete"] верно для "edit delete" и неверно для "undelete" или "no-delete".

На конец атрибута:

- [attr\$="val"] – атрибут заканчивается на val, например равен "myval".

# Специфичность селекторов CSS

22

```
li:first-child h2 .title
```

0

Inline styles

0

IDs

2

Classes, attributes  
and pseudo-classes

2

Elements and  
pseudo-elements

+ Duplicate

```
#nav .selected > a:hover
```

0

Inline styles

1

IDs

2

Classes, attributes  
and pseudo-classes

1

Elements and  
pseudo-elements

+ Duplicate

**Display**



```
<div style="border: 1px solid black">
```

Невидимый div (

```
  <div style="display: none">Я - невидим!</div>
```

) Стоит внутри скобок

```
</div>
```

Невидимый div ( ) Стоит внутри скобок

# Значение block

```
<div style="border:1px solid black">  
  <div style="border:1px solid blue; width: 50%">Первый</div>  
  <div style="border:1px solid red">Второй</div>  
</div>
```

Первый

Второй

```
<span style="border:1px solid black">
```

```
  <span style="border:1px solid blue; width:50%">Ширина</span>
```

```
  <a style="border:1px solid red">Игнорируется</a>
```

```
</span>
```

```
<div style="width:400px">
```

```
  ...<span style="background: lightgreen">
```

```
    Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля
```

```
    Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля
```

```
  </span>...
```

```
</div>
```

```
... Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля  
Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля Ля  
Ля Ля Ля Ля Ля ...
```

```
<div style="border:1px solid black">
```

```
  <span style="border:1px solid red">Инлайн</span>
```

```
  <div style="border:1px solid blue; width:50%">Блок</div>
```

```
  <span style="border:1px solid red">Инлайн</span>
```

```
</div>
```

Инлайн	
Блок	
Инлайн	

# Значение inline-block

27

```
<style>
```

```
li { display: inline-block; list-style: none; border: 1px solid red; vertical-align: middle; }
```

```
</style>
```

```
<ul style="border: 1px solid black; padding: 0">
```

```
<li>Инлайн Блок<br>3 строки<br>высота/ширина явно не заданы</li>
```

```
<li style="width: 100px; height: 100px">Инлайн<br>Блок 100x100</li>
```

```
<li style="width: 60px; height: 60px">Инлайн<br>Блок 60x60</li>
```

```
<li style="width: 100px; height: 60px">Инлайн<br>Блок 100x60</li>
```

```
<li style="width: 60px; height: 100px">Инлайн<br>Блок 60x100</li>
```

```
</ul>
```

Инлайн Блок 3 строки высота/ширина явно не заданы	Инлайн Блок 100x100	Инлайн Блок 60x60	Инлайн Блок 100x60	Инлайн Блок 60x100
---	------------------------	-------------------------	-----------------------	--------------------------

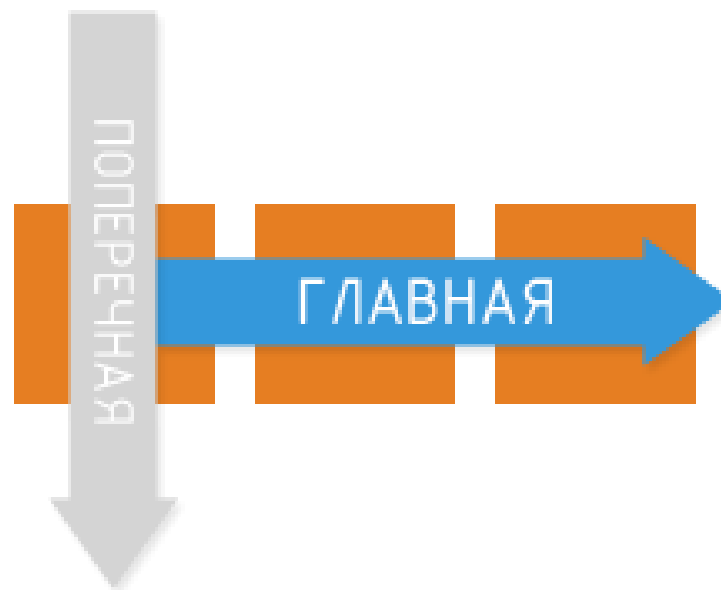
# Значение table

```
<form style="display: table">
  <div style="display: table-row">
    <label style="display: table-cell">Имя:</label>
    <input style="display: table-cell">
  </div>
  <div style="display: table-row">
    <label style="display: table-cell">Фамилия:</label>
    <input style="display: table-cell">
  </div>
</form>
```

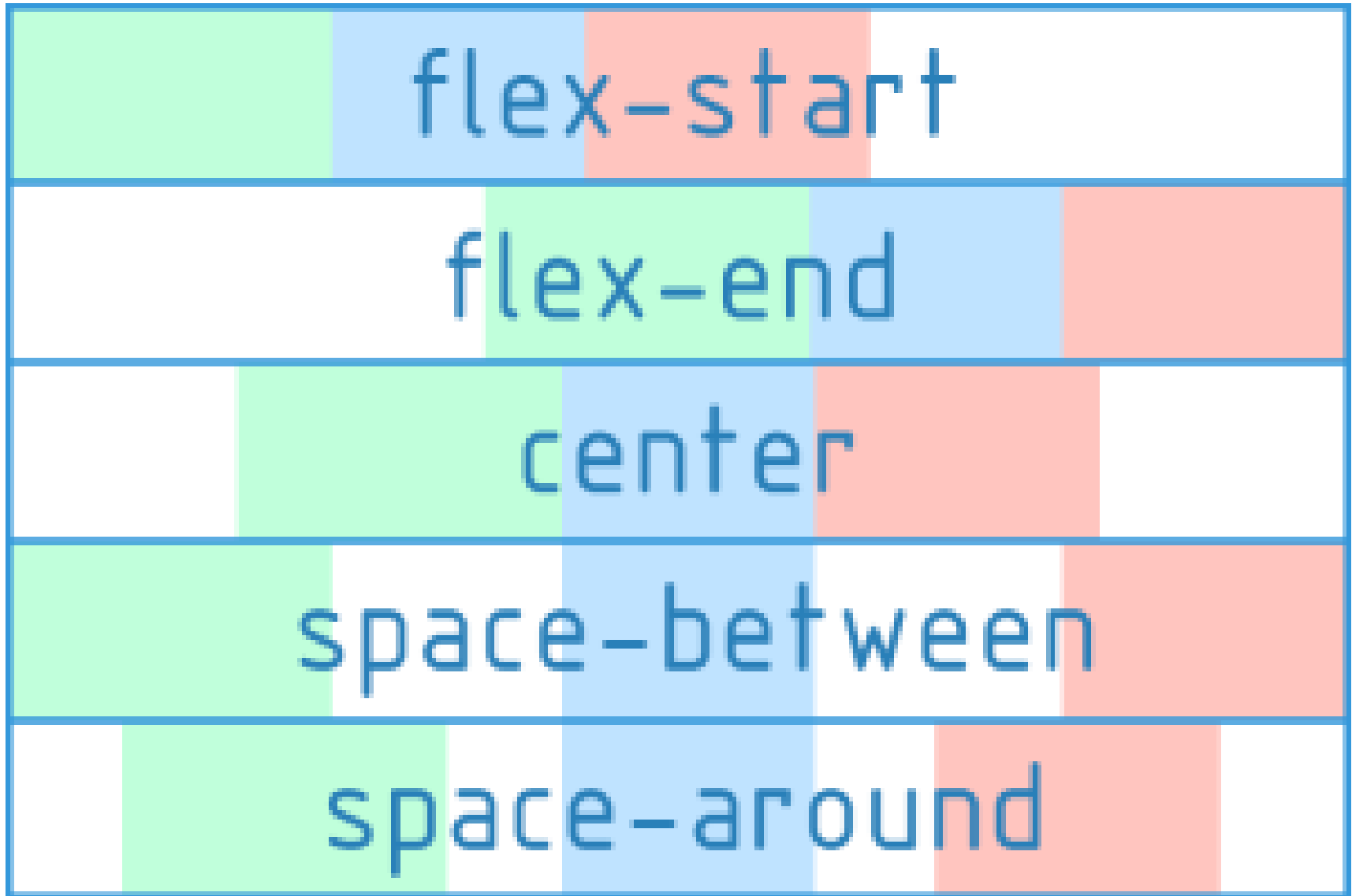
Имя:	<input type="text"/>
Фамилия:	<input type="text"/>

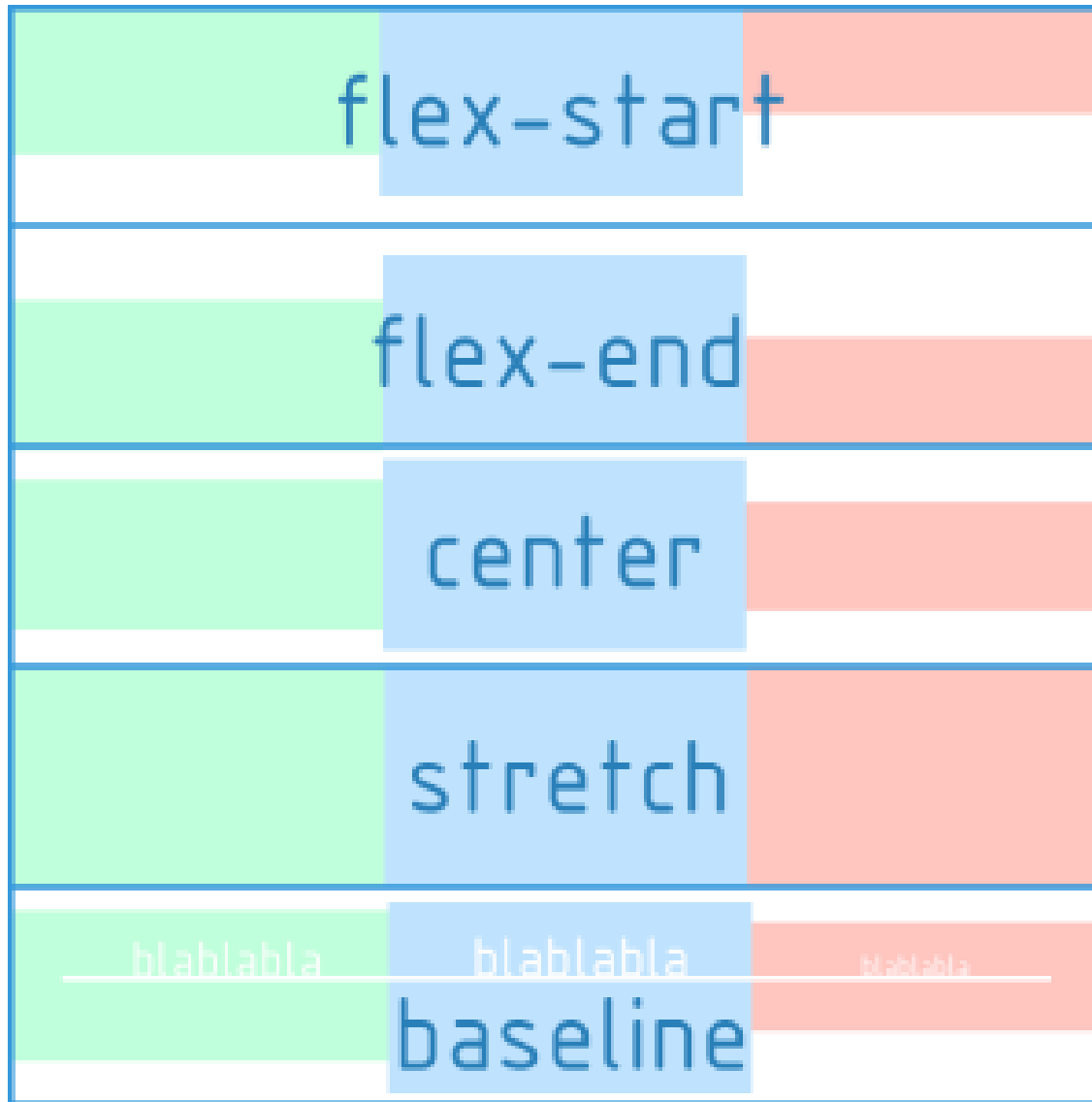


flex-direction: **column**



flex-direction: **row**







## **flex-wrap:**

nowrap

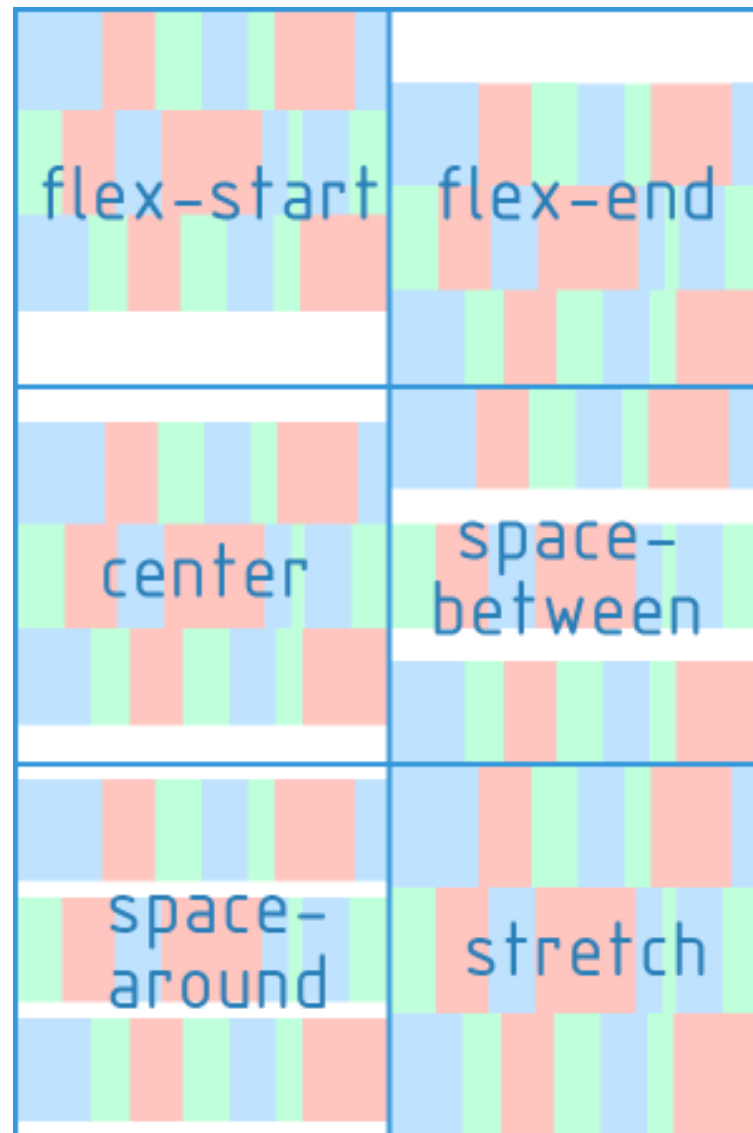
wrap

wrap-reverse

## **flex-flow** – удобное

сокращение для flex-direction

+ flex-wrap



**flex-basis** – базовый размер отдельно взятого flex-блока

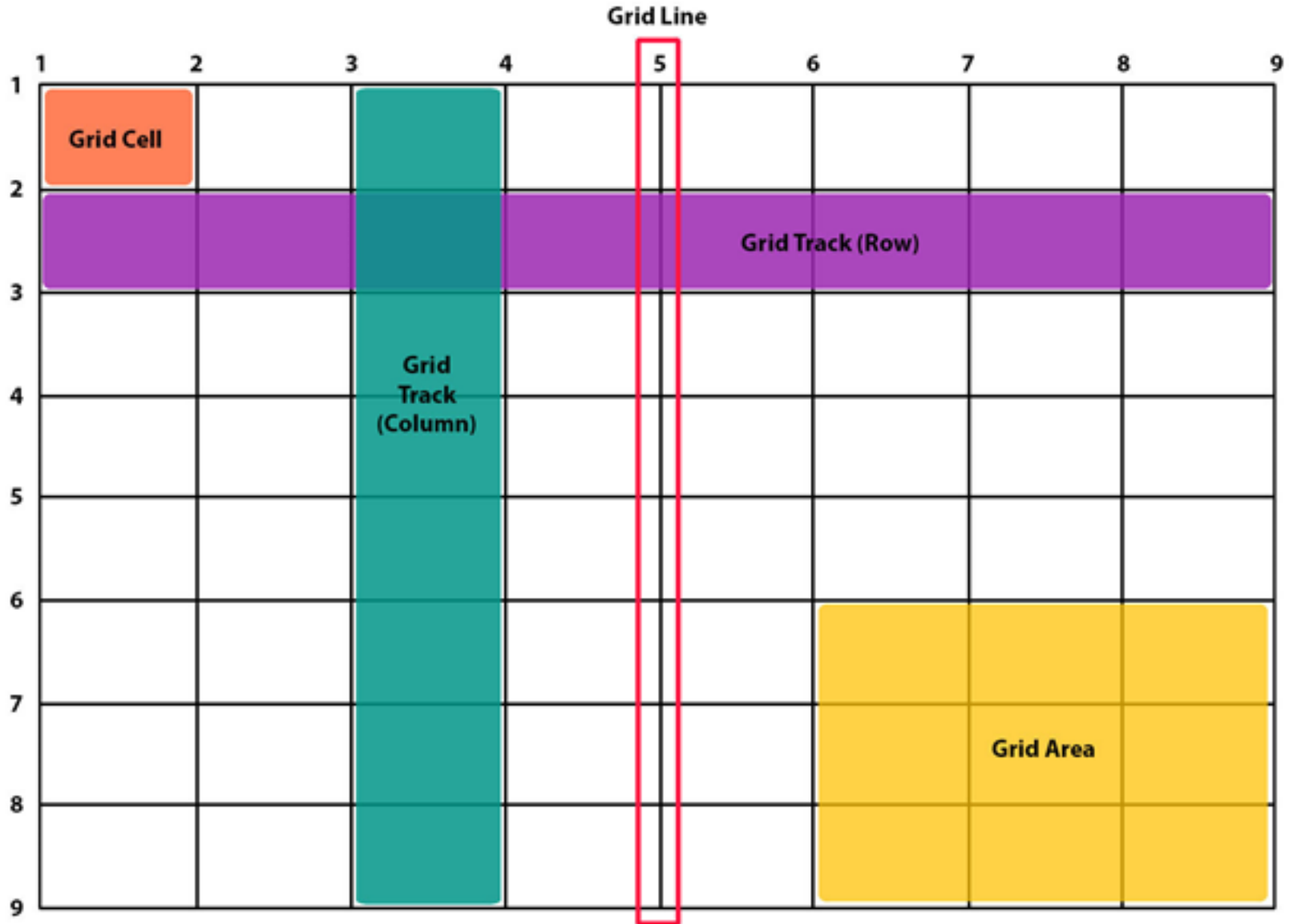
**flex-grow** – “жадность” отдельно взятого flex-блока

**flex-shrink** – фактор “сжимаемости” отдельно взятого flex-блока

**align-self** – выравнивание отдельно взятого flex-блока по поперечной оси.

**order** – порядок следования отдельно взятого flex-блока внутри flex-контейнера.

**flex** – короткая запись для свойств flex-grow, flex-shrink и flex-basis



```
#grid {  
  display: grid;  
  grid-template-rows: 1fr 1fr 1fr;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-gap: 2vw;  
}
```

```
#grid > div {  
  font-size: 5vw;  
  padding: .5em;  
  background: gold;  
  text-align: center;  
}
```

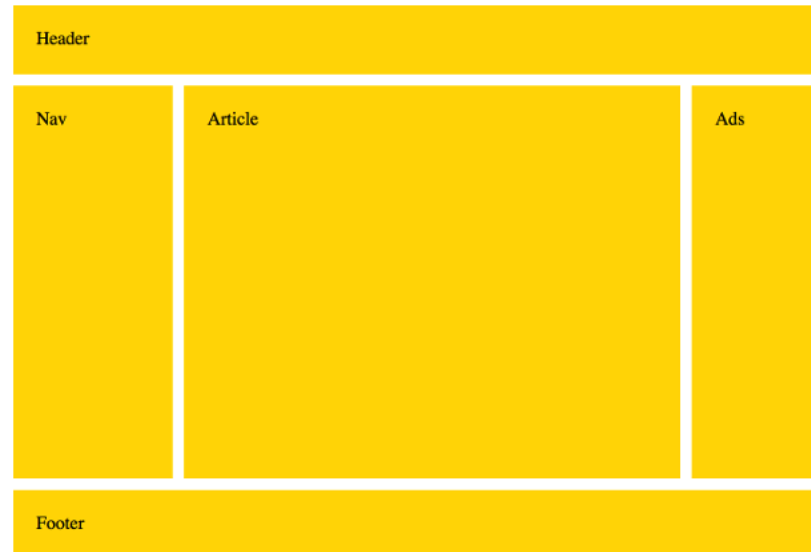


```
grid-template-rows: 1fr 1fr 1fr 1fr 1fr;
```

```
grid-template-rows: repeat(5, 1fr);
```

```
body {  
  display: grid;  
  grid-template-areas:  
    "header header header"  
    "nav article ads"  
    "footer footer footer";  
  grid-template-rows: 60px 1fr 60px;  
  grid-template-columns: 20% 1fr 15%;  
  grid-gap: 10px;  
  height: 100vh;  
  margin: 0;  
}  
header, footer, article, nav, div {  
  padding: 20px;  
  background: gold;  
}  
#pageHeader { grid-area: header;}  
#pageFooter {grid-area: footer;}  
#mainArticle {grid-area: article;}  
#mainNav {grid-area: nav;}  
#siteAds {grid-area: ads;}
```

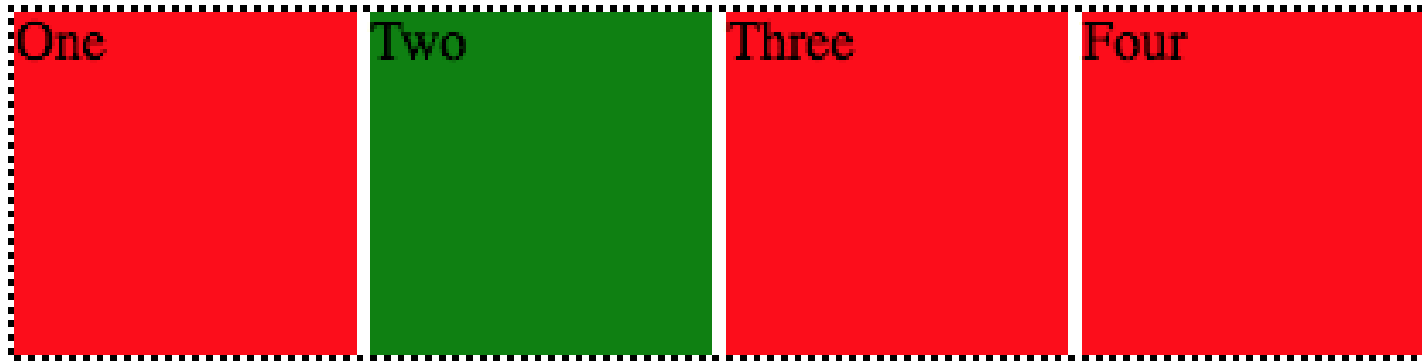
```
<body>  
  <header id="pageHeader">Header</header>  
  <article id="mainArticle">Article</article>  
  <nav id="mainNav">Nav</nav>  
  <div id="siteAds">Ads</div>  
  <footer id="pageFooter">Footer</footer>  
</body>
```



**Position**

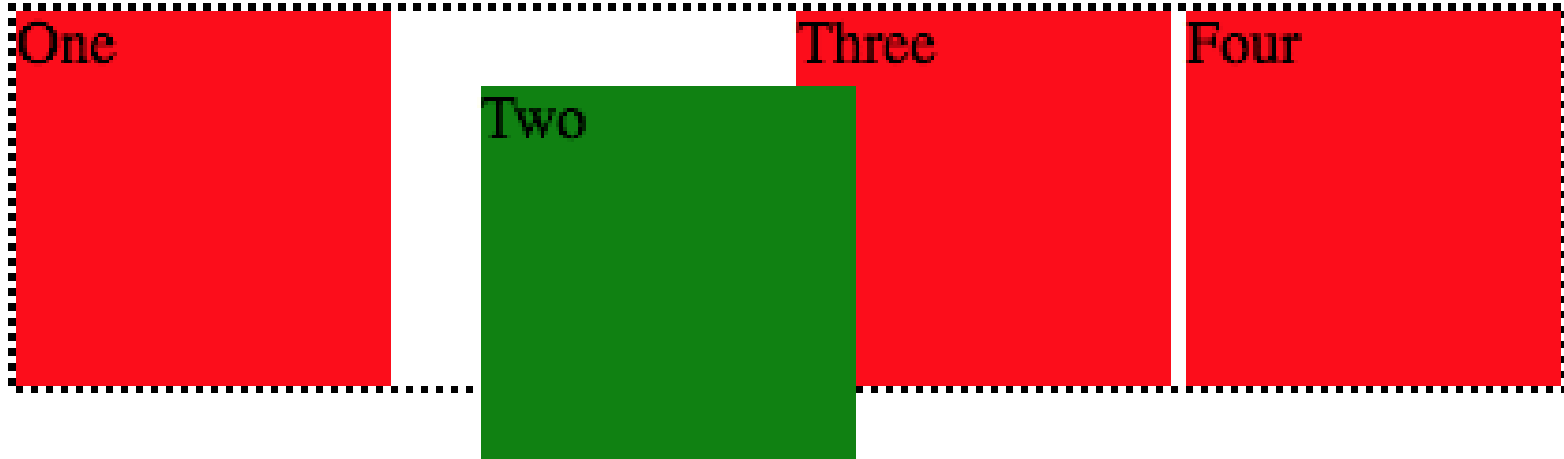
# position: static

39



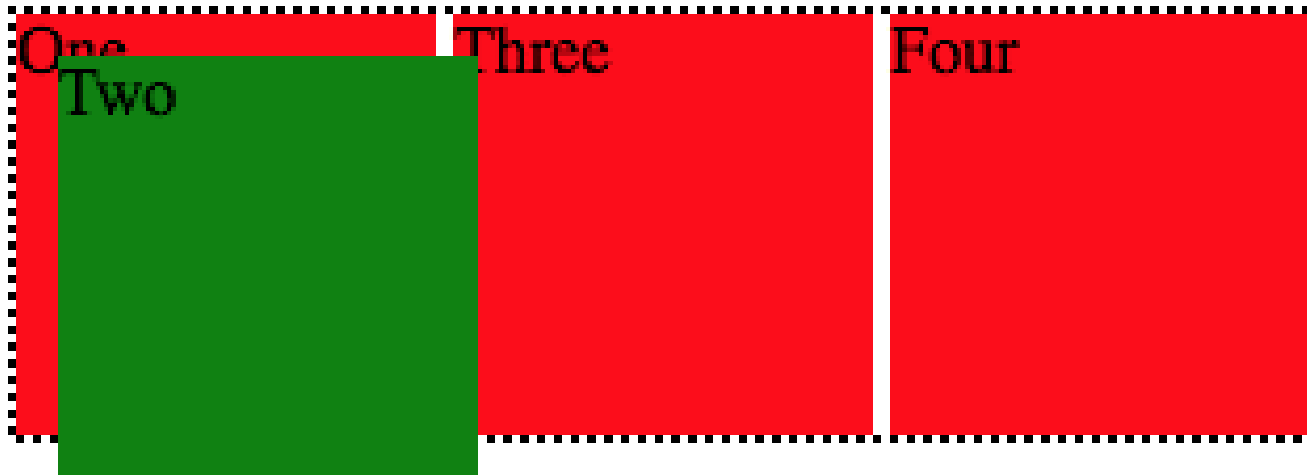
# position: relative

40



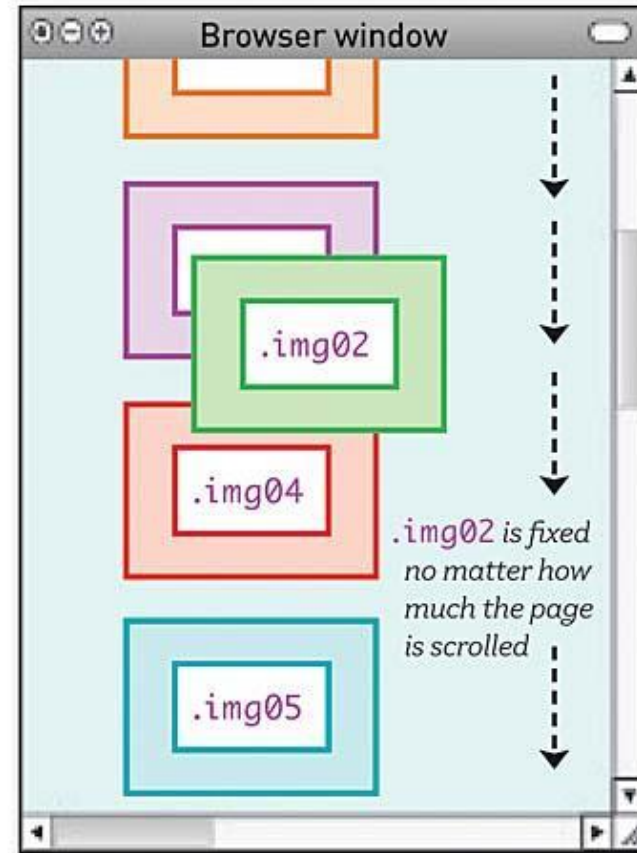
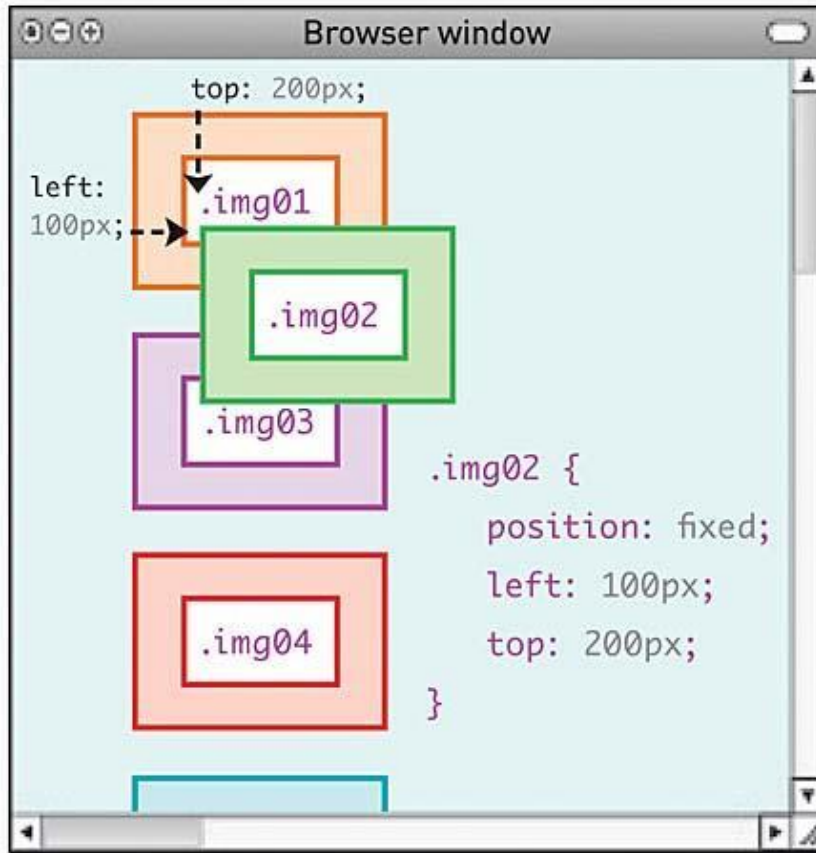


# position: absolute



# position: fixed

42



**Блочная модель**

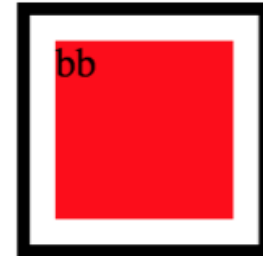
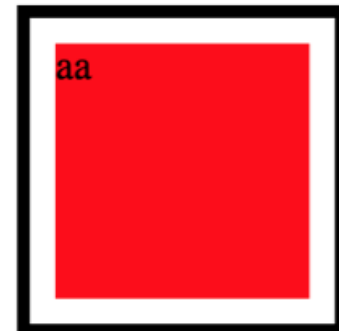
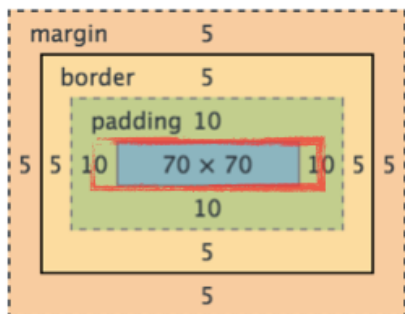
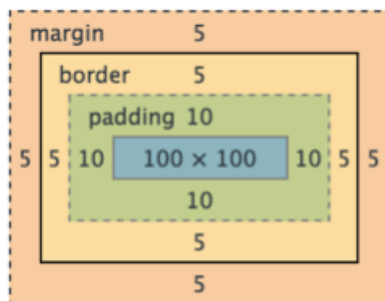
# box-sizing

44

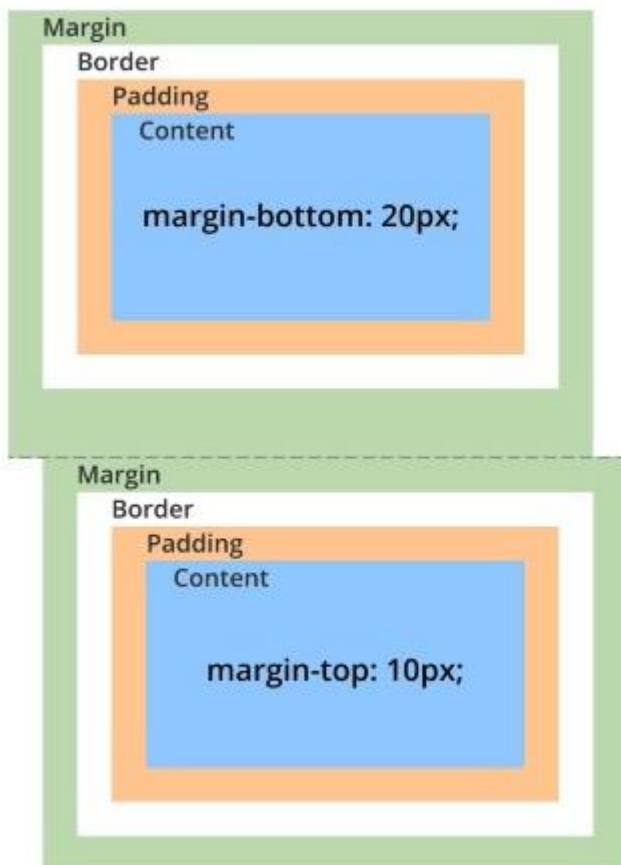
**content-box** -- значение по умолчанию. В этом случае свойства width/height обозначают то, что находится *внутри padding*.

**border-box** -- значения width/height задают высоту/ширину *элемента* обозначают то, что находится *внутри margin*.

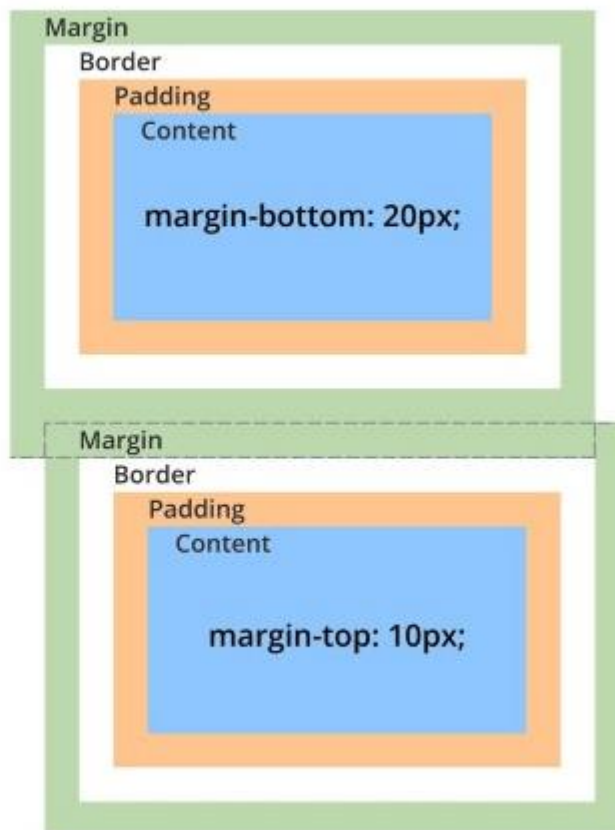
```
5 #aa{  
6   height:100px;  
7   width: 100px;  
8   padding: 10px;  
9   border: 5px solid black;  
10  margin: 5px;  
11 }  
12  
13 #bb{  
14   height:100px;  
15   width: 100px;  
16   padding: 10px;  
17   border: 5px solid black;  
18   margin: 5px;  
19   box-sizing:border-box;  
20 }
```



## Ожидание

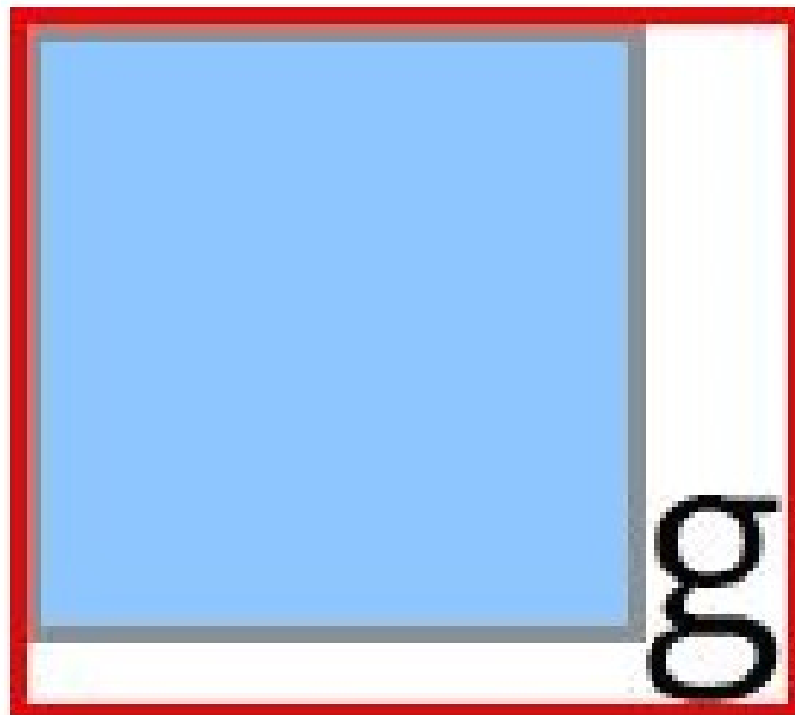
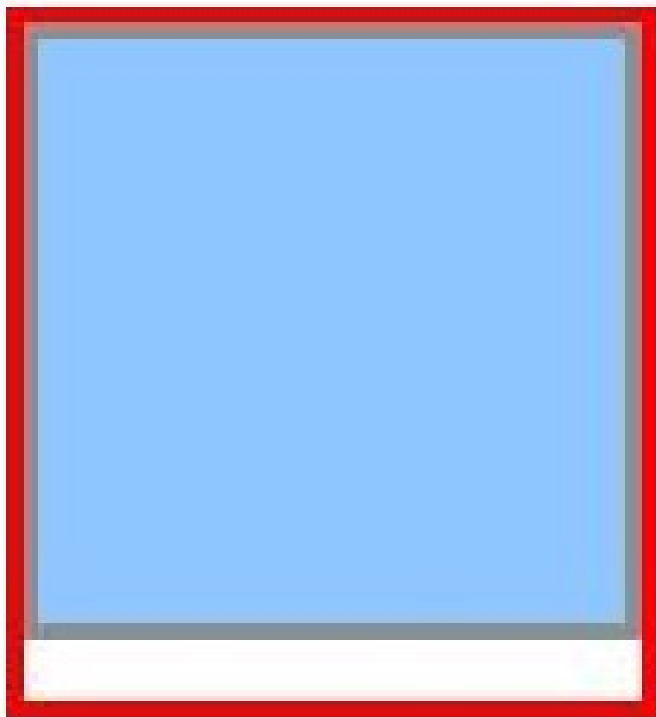


## Реальность



# Лишнее место под тег `<img>`

46



Решение:  
`display: block;`

**Контекст наложения**

# Создание контекста и поведение элементов

Контекст может формироваться в любом месте документа, любым элементом, у которого выполняется одно из следующих условий:

- ❑ Корневой элемент `body` всегда содержит корневой контекст наложения.
- ❑ Элемент с `position` отличным от `static` и значением `z-index` отличным от `auto`. Кроме исключения для `position: fixed`.
- ❑ Мобильные браузеры на основе WebKit, всегда создают новый контекст наложения для `position: fixed`-элементов, даже с `z-index: auto`.
- ❑ Элемент имеет значение `opacity` меньше, чем 1.
- ❑ Flex-элемент со значением `z-index` отличным от `auto`, даже в случае `position: static`.
- ❑ Трансформированные элементы со значением `transform` отличное от `none`, `transform-style` со значением `preserve-3d` и `perspective` со значением отличным от `none`.
- ❑ CSS Regions с установленными значениями `flow-from` отличное от `none` для элемента с `content` отличным от `normal`.



# Пример наложения контекстов



Чтобы этого не происходило, блоки с подобным поведением следует оборачивать блоками с уже заданным новым контекстом наложения, можно использовать `position: relative + z-index: 10`

**Единицы измерения**

# Единицы измерения

Пиксели: px

Относительно шрифта: em

Единица `rem` задаёт размер относительно размера шрифта элемента `<html>`.

```
1 <div style="font-size:1.5em">
2   Страусы
3   <div style="font-size:1.5em">Живут также в Африке</div>
4 </div>
```

Страусы  
Живут также в Африке

Проценты %

Относительно экрана: vw, vh, vmin, vmax

**Медиазапросы**

## 1. Оператор and

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

## 2. Оператор запятая

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

## 3. Оператор not

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

===

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

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

===

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

all	Подходит для всех типов устройств.
print	Предназначен для страничных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.
screen	Предназначен в первую очередь для экранов цветных компьютерных мониторов.
speech	Предназначен для синтезаторов речи.

Width	<p>Проверяет ширину области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (width: 800px). Обычно для проверки используются минимальные и максимальные значения ширины.</p> <p>min-width применяет правило если ширина области просмотра больше значения, указанного в запросе, max-width — ширина области просмотра меньше значения, указанного в запросе.</p>
Height	<p>Проверяет высоту области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (height: 500px). Обычно для проверки используются минимальные и максимальные значения высоты.</p> <p>min-height применяет правило если высота области просмотра больше значения, указанного в запросе, max-height — высота области просмотра которого меньше значения, указанного в запросе.</p>
orientation	<p>Проверяет ориентацию области просмотра. Принимает два значения: (orientation: portrait) и (orientation: landscape).</p>
resolution	<p>Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (dpi) или количество точек на сантиметр (dpcm), например, (resolution: 300dpi).</p> <p>min-resolution проверяет минимальное разрешение экрана, max-resolution — максимальное.</p>

Для управления разметкой в мобильных браузерах используется метатег viewport. Мобильные браузеры отображают страницы в виртуальном окне просмотра, которое обычно шире, чем экран устройства. С помощью метатега viewport можно контролировать размер окна просмотра и масштаб.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



# Стратегии использования медиа-запросов

- 1) Уменьшение количества колонок (столбцов) и постепенная отмена обтекания для мобильных устройств.
- 2) Использование свойства `max-width` вместо `width` при задании ширины блока-контейнера.
- 3) Уменьшение полей и отступов на мобильных устройствах (например, нижних отступов между заголовком и текстом, левого отступа для списков и т.п.).
- 4) Уменьшение размеров шрифтов для мобильных устройств.
- 5) Превращение линейных навигационных меню в раскрывающиеся.
- 6) Скрытие второстепенного содержимого на мобильных устройствах с помощью `display: none`.
- 7) Подключение фоновых изображений уменьшенных размеров.

**БЭМ**

БЭМ (Блок, Элемент, Модификатор) — компонентный подход к веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки. Он позволяет легко и быстро разрабатывать интерфейсы любой сложности и повторно использовать существующий код, избегая «Copy-Paste».

- ❑ Название блока характеризует смысл
- ❑ Блок не должен влиять на свое окружение
- ❑ В CSS по БЭМ также не рекомендуется использовать селекторы по тегам или id.
- ❑ Блоки можно вкладывать друг в друга.

```
<!-- Блок `header` -->
```

```
<header class="header">
```

```
  <!-- Вложенный блок `logo` -->
```

```
  <div class="logo"></div>
```

```
  <!-- Вложенный блок `search-  
form` -->
```

```
  <form class="search-  
form"></form>
```

- ❑ Составная часть блока, которая не может использоваться в отрыве от него.
- ❑ Название элемента характеризует смысл
- ❑ Структура полного имени элемента соответствует схеме: имя-блока\_\_имя-элемента.
- ❑ Элементы можно вкладывать друг в друга. Но! Элемент — всегда часть блока, а не другого элемента.

<!-- Неверно. -->

<!-- Блок `search-form` -->

<form class="search-form">

<form class="search-form">

<div class="search-form\_\_content">

<!-- Элемент `input` блока `search-form` -->

<!-- Рекомендуется:

<input class="search-form\_\_input">

`search-form\_\_input` или

`search-form\_\_content-input`

<!-- Элемент `button` блока `search-form` -- -->

>

<input class="search-

<button class="search-form\_\_button"> form\_\_content\_\_input">

Найти

</button>

</div>

</form>

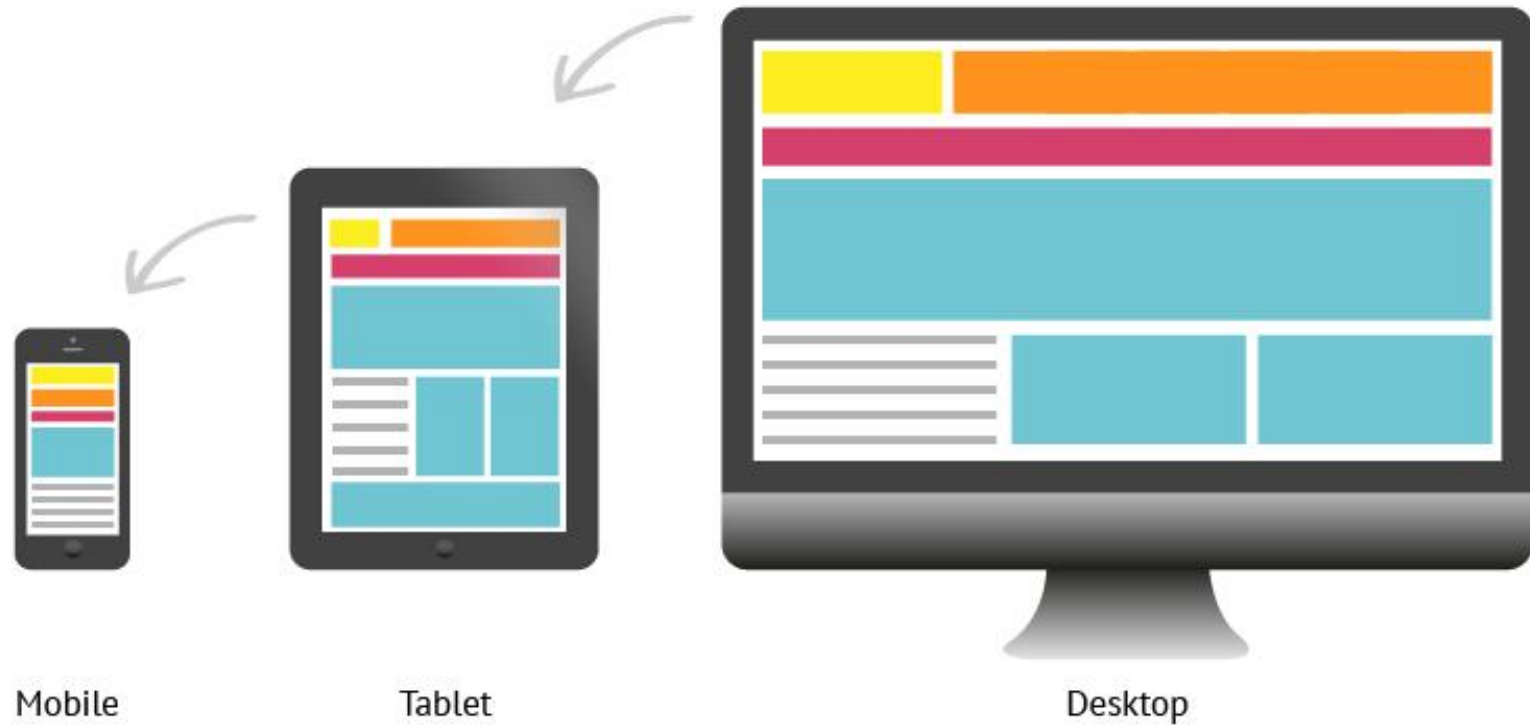
</form>

- ❑ Название модификатора характеризует внешний вид, состояние и поведение
- ❑ Имя модификатора отделяется от имени блока или элемента одним подчеркиванием (\_).
- ❑ Модификатор нельзя использовать самостоятельно

```
<!-- Блок `search-form` имеет булевый модификатор `focused` -->
<form class="search-form search-form_focused">
  <input class="search-form__input">








  <!-- Элемент `button` имеет модификатор `disabled` -->
  <button class="search-form__button search-
form__button_disabled">Найти</button>
</form>
```

**Responsive design**





# Отсечки размеров responsive дизайна 63

	<code>@media only screen and (max-width: 767px) and (orientation: landscape) {...}</code>
	<code>@media only screen and (min-width: 481px) and (max-width: 767px) and (orientation: landscape) {...}</code>
	<code>@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) {...}</code>
	<code>@media only screen and (max-width: 767px) and (orientation: portrait) {...}</code>
	<code>@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) {...}</code>
	<code>@media only screen and (max-width: 480px) {...}</code>
	<code>@media only screen and (min-width: 1601px) {...}</code>

**Полезные техники**

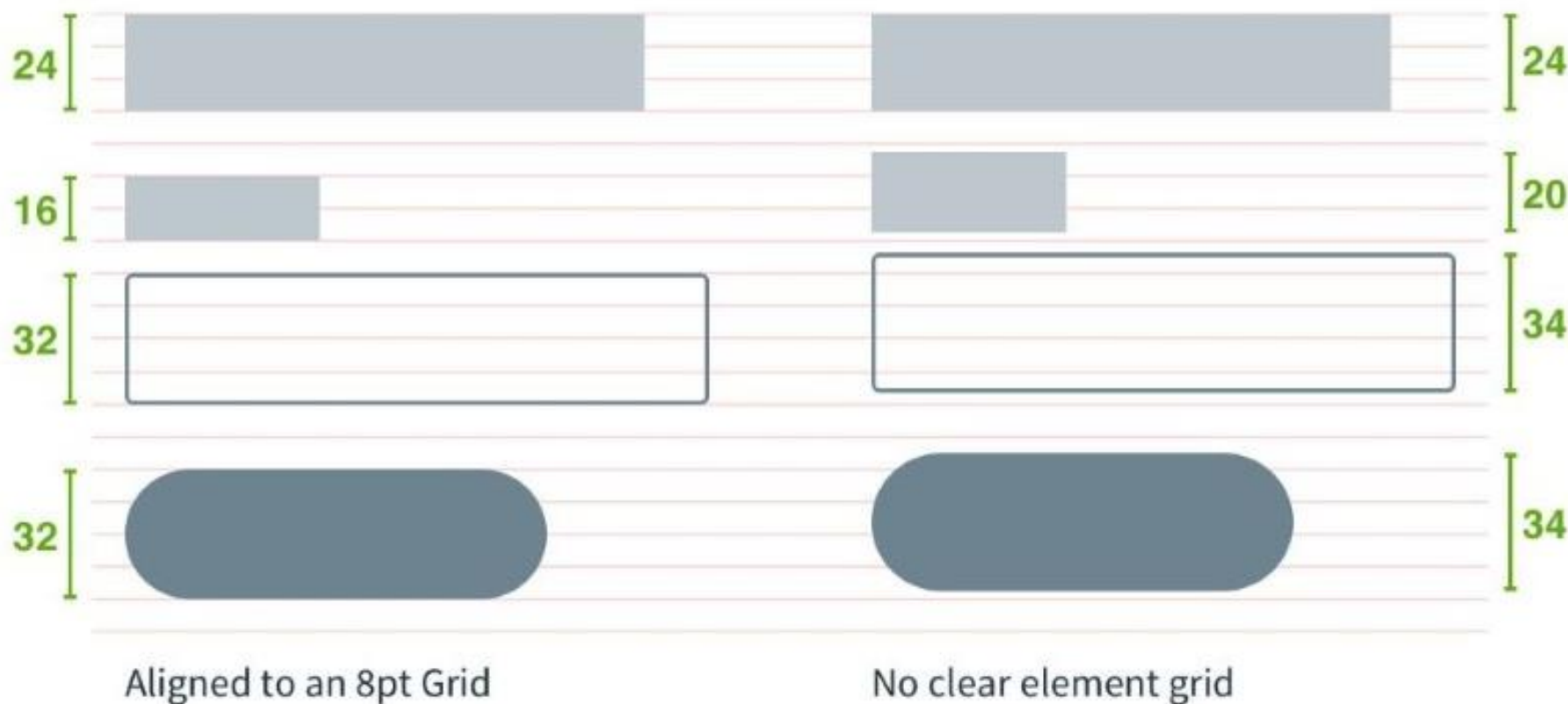
```
.block-snow {  
    background: url(images/snow.png) no-repeat  
center center black;  
}
```

```
.block-sunny {  
    background: url(images/sunny.png) no-repeat  
center center black;  
}
```

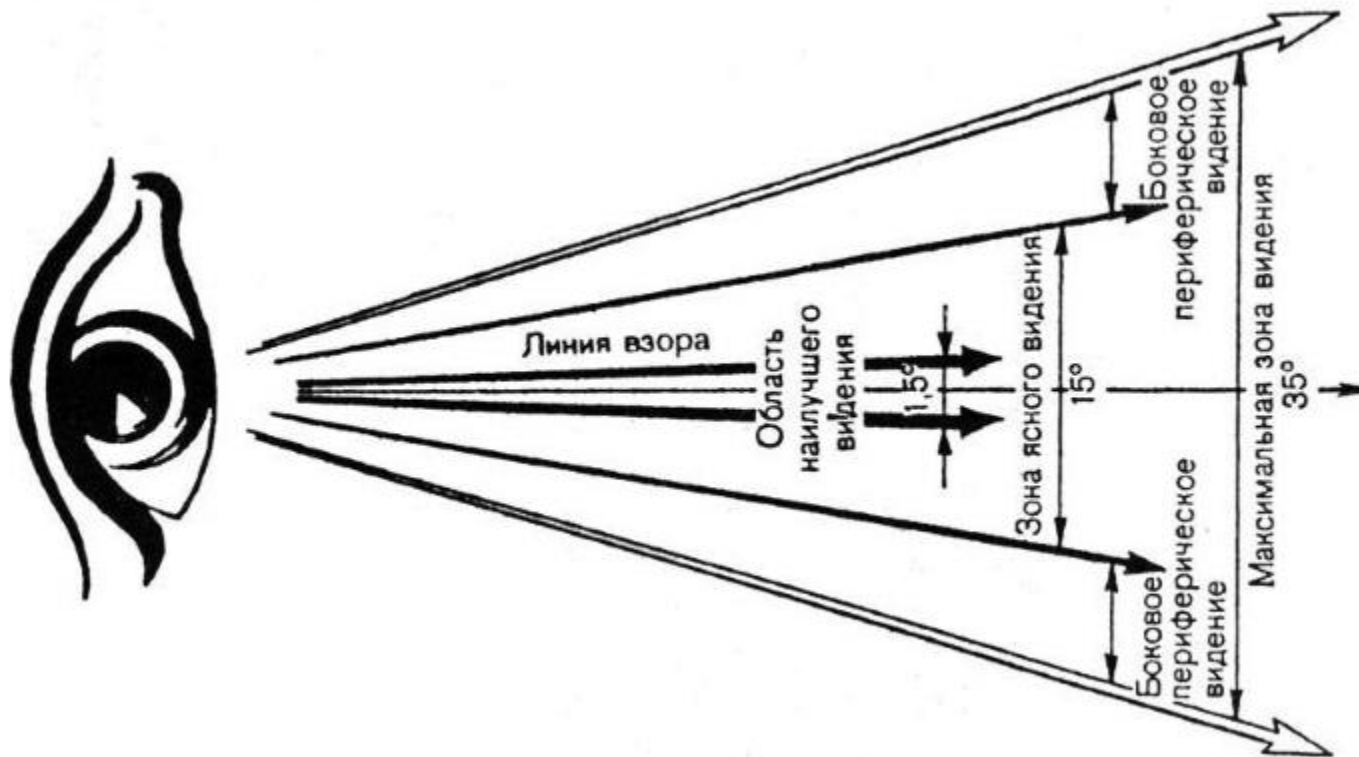
```
<body class="block block-snow">  
</body>
```

# 8-Point Grid System

65



Рассчитывайте размеры всех элементов и расстояния между ними с шагом в 8, т.е. любая заданная высота, ширина, размер поля или отступа будут кратными восьми.



Рекомендуемое расстояние до монитора 0,65м

Диагональ монитора 24" = 0.5м

Зона наилучшего зрения -  $0.65\text{м} * \text{tg}(15^\circ) = 0.17\text{м}$

Зона максимального зрения -  $0.65\text{м} * \text{tg}(36^\circ) = 0.47\text{м}$

# CSS PREPROCESSORS

70

## CSS PREPROCESSOR

LESS

SASS & SCSS

STYLUS

### SCSS

```
1  section {  
2    height: 100px;  
3    width: 100px;  
4  
5    .class-one {  
6      height: 50px;  
7      width: 50px;  
8  
9      .button {  
10       color: #074e68;  
11     }  
12   }  
13 }
```

### CSS

```
1  section {  
2    height: 100px;  
3    width: 100px;  
4  }  
5  
6  section .class-one {  
7    height: 50px;  
8    width: 50px;  
9  }  
10  
11 section .class-one .button {  
12   color: #074e68;  
13 }
```

## Variables

These are pretty self-explanatory:

```
@width: 10px;
@height: @width + 10px;

#header {
  width: @width;
  height: @height;
}
```

Outputs:

```
#header {
  width: 10px;
  height: 20px;
}
```

## Mixins

Mixins are a way of including ("mixing in") a block of CSS. If we have the following class:

```
.bordered {
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}
```

And we want to use these properties inside other classes where we want the properties, like so:

```
#menu a {
  color: #111;
  .bordered();
}

.post a {
  color: red;
  .bordered();
}
```

## Nesting

Less gives you the ability to use nesting in the following CSS:

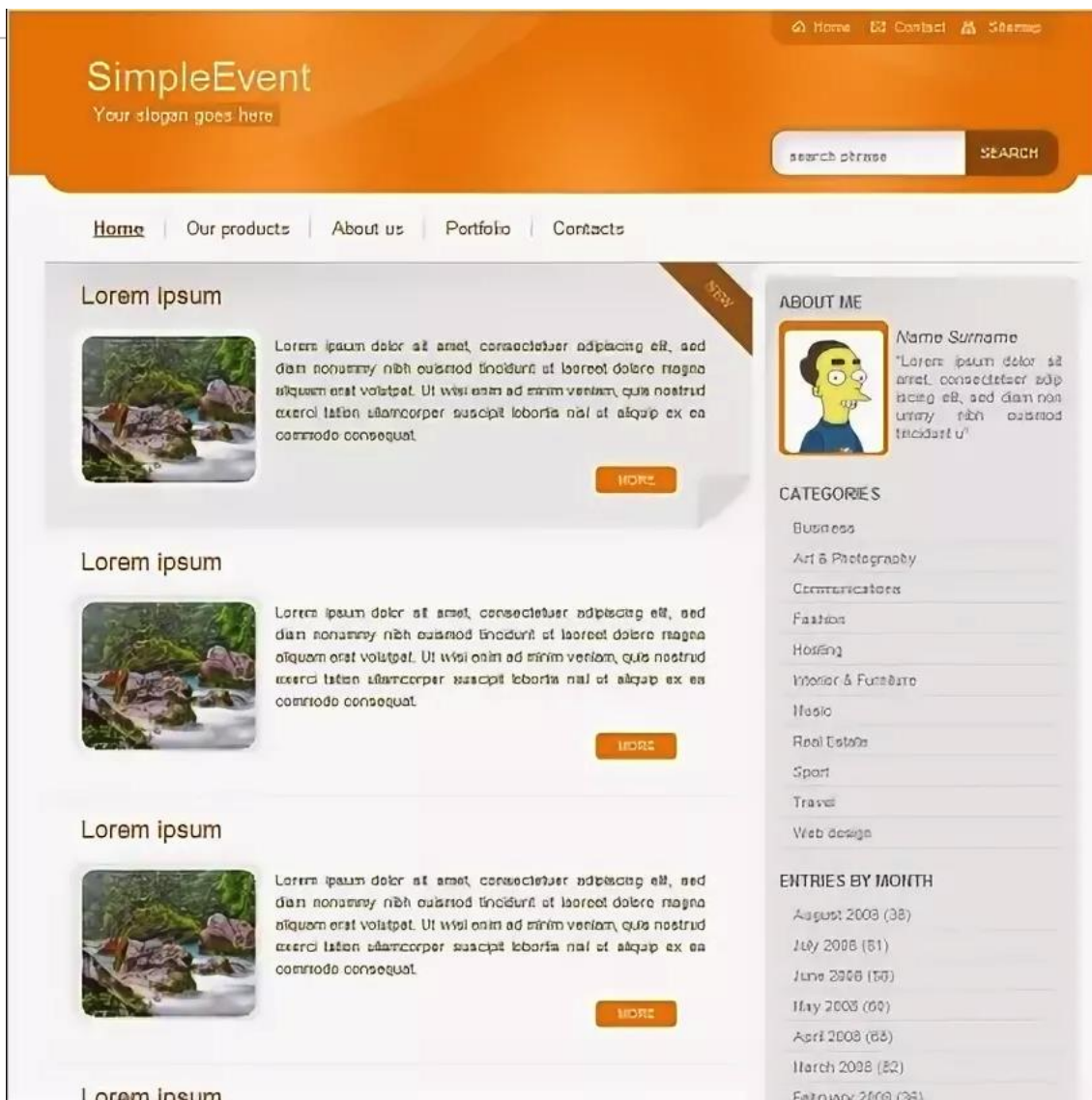
```
#header {
  color: black;
}
#header .navigation {
  font-size: 12px;
}
#header .logo {
  width: 300px;
}
```

In Less, we can also write it this way:

```
#header {
  color: black;
  .navigation {
    font-size: 12px;
  }
  .logo {
    width: 300px;
  }
}
```

# Домашнее задание

72



Что проверяем:  
Семантичность  
БЭМ  
Адаптивность  
(медиазапросы)  
Метаданные

Разместить на  
гитхаб/пейджес?  
/скрин?  
<https://pages.github.com/>  
В ридми добавить  
ссылку на гитхаб  
пейджес



## Полезные ссылки:

1. <http://caniuse.com/>
2. <http://css-tricks.com/>
3. <https://github.com/hail2u/html-best-practices>
4. <https://css-tricks.com/snippets/css/complete-guide-grid/>
5. <https://html5.by/blog/flexbox/>
6. [https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Document\\_and\\_web\\_site\\_structure](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_web_site_structure)
7. <http://validator.w3.org/>
8. <https://fontawesome.com/> - для д/з