

Основа верстки

Разметка и стилизация

# Подробнее про HTML

- табличная верстка
- div + float
- семантический HTML + flexbox/grid

# Структура HTML

<DOCTYPE> -- предназначен для указания типа текущего документа

<head> -- содержит специальную информацию, стили и favicon

<meta> -- определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем

<link> -- устанавливает связь с внешним документом вроде файла со стилями или со шрифтами

<body> -- содержит разметку страницы

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page title</title>
    <meta charset="utf-8" />
    <meta
      http-equiv="Content-Type"
      content="text/html; charset=utf-8"
    />
    <link
      rel="stylesheet"
      href="style.css"
    />
    <link
      rel="shortcut icon"
      href="http://htmlbook.ru/favicon.ico"
    />
  </head>
  <body>
    ...
  </body>
</html>
```

# Семантический HTML vs верстка на div'ах

```

<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>
    <header class="main-header">
      <!-- Шапка сайта -->
    </header>
    <main>
      <!-- Основное содержимое страницы -->
    </main>
    <footer class="main-footer">
      <!-- Подвал сайта -->
    </footer>
  </body>
</html>

```

```

<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>
    <div class="main-header">
      <!-- Шапка сайта -->
    </div>
    <div>
      <!-- Основное содержимое страницы -->
    </div>
    <div class="main-footer">
      <!-- Подвал сайта -->
    </div>
  </body>
</html>

```

# Необходимые и достаточные атрибуты

Специальные атрибуты:

data-\*, href, required, readonly, src, class, id, disable...

Атрибуты для доступности контента:

alt, accesskey...

# Блочные и строчные элементы

**Блочные элементы** занимают всю ширину своего родителя (контейнера), формально создавая «блок» (отсюда и название).

## Примеры тегов:

`<header>`, `<footer>`, `<main>`, `<div>`, `<p>`

**Строчные элементы** занимают только пространство, ограниченное тегами, определяющими строчный элемент (необходимое для отображения их содержимого) и не нарушая потока содержимого.

## Примеры тегов:

`<span>`, `<a>`, `<img>`, `<input>`, `<select>`

## Блочные элементы

DIV

P

UL

OL

## Встроенные элементы

SPAN SPAN

SPAN

SPAN A

# Чтобы появилась стилизация

1. добавьте файл с расширением \*.css
2. подключите его на страницу с помощью <link>
3. используя классы, селекторы или псевдоселекторы напишите стилизацию
4. добавьте классы к тегам
5. наслаждайтесь результатами

# Классы, селекторы и псевдоселекторы

```
body {  
  margin: 0;  
}  
  
header {  
  padding: 16px;  
  background: black;  
  color: #fff;  
}  
  
.userlist {  
  padding: 16px;  
}  
  
.userItem {  
  list-style-type: none;  
}  
  
.userlist>li:nth-child(2n) {  
  background: red;  
  color: white;  
}
```

```
...  
<body>  
  <header>  
    <h1>Native HTML & CSS project</h1>  
  </header>  
  <main>  
    <ul class='userlist'>  
      <li class='userItem'>user 1</li>  
      <li class='userItem'>user 2</li>  
      <li class='userItem'>user 3</li>  
      <li class='userItem'>user 4</li>  
      <li class='userItem'>user 5</li>  
    </ul>  
  </main>  
</body>  
...
```

## Native HTML & CSS project

user 1  
user 2  
user 3  
user 4  
user 5



# Единицы размеров

**Статические:** px pt

**Относительные:**

Название	Относительно чего считается
em	размера шрифта (1em = текущий размер шрифта)
rem	заданы относительно размера шрифта <html>
%	такого же свойства родителя
vw	ширины окна (1vw = 1% от ширины окна)
vh	высоты окна (1vh = 1% от высоты окна)
vmin	наименьшее из vw, vh
vmax	наибольшее из vw, vh

# Использование шрифтов

**@font-face** CSS правило позволяет указать шрифты для отображения текста на веб-страницах

часто используемые свойства для работы с текстом

**font-family, font-stretch, font-style, font-weight, font-size**

```
@font-face {
  font-family: MyHelvetica;
  src: local("Helvetica Neue Bold"),
       local("HelveticaNeue-Bold"),
       url(MgOpenModernaBold.ttf);
  font-weight: bold;
}
```

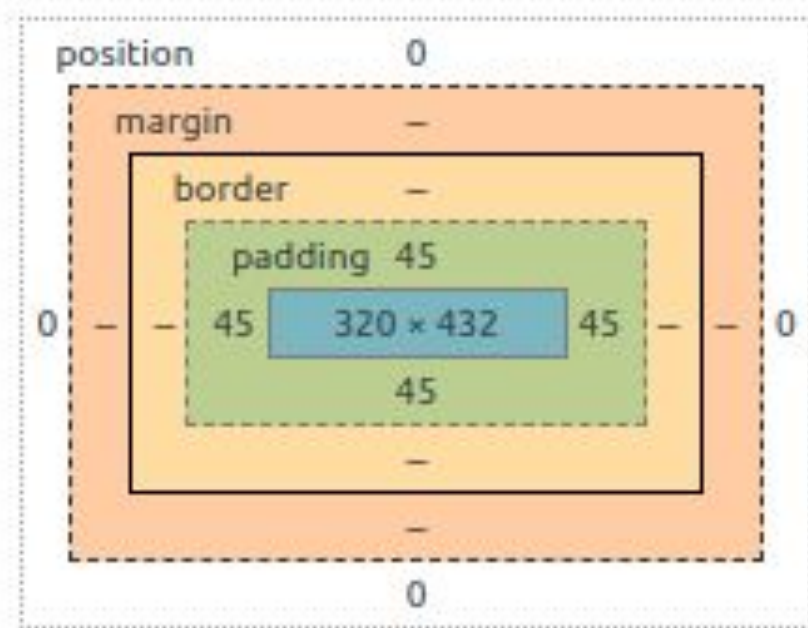
```
...
<link rel="preload" href="/fonts/roboto.woff2" as="font">
...
```

**Предварительная загрузка шрифтов**

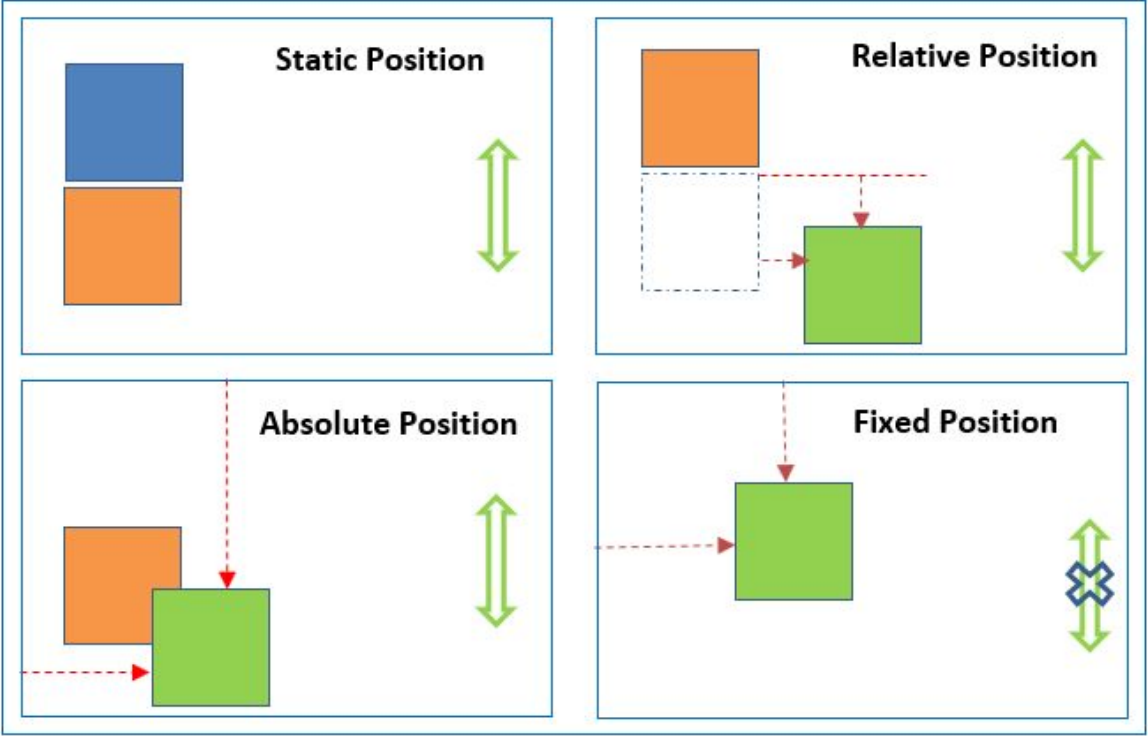
# Размеры объектов в браузере

**Вычисленное значение** - это преобразование относительных значений (em, rem, проценты и тп) в абсолютные значения.

```
.menu {  
  box-sizing: border-box  
}
```

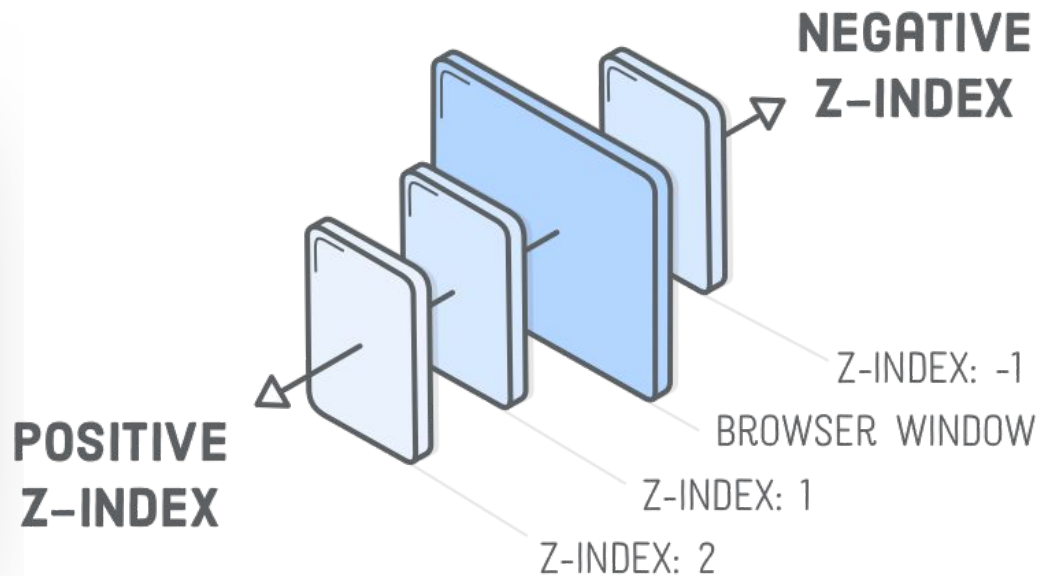


# Поток документа



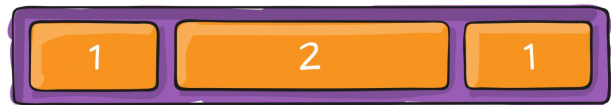
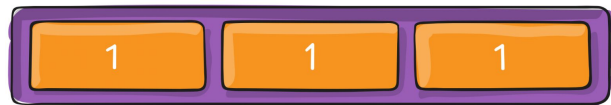
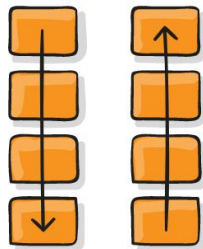
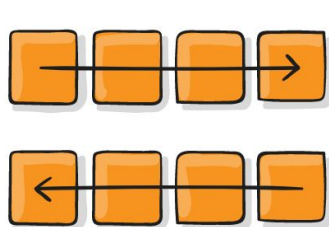
# Z-INDEX

```
.modal {  
  z-index: 1  
}  
  
.modal-background {  
  z-index: 0  
}
```



# Flexbox

предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS



flex-start



flex-end



center



space-between



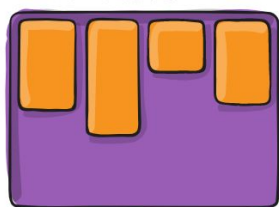
space-around



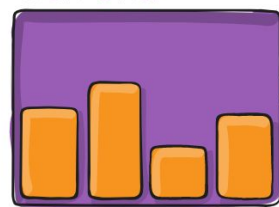
space-evenly



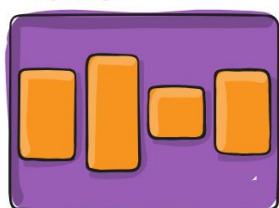
flex-start



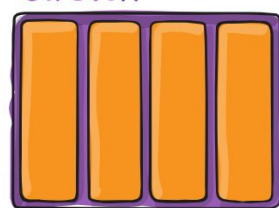
flex-end



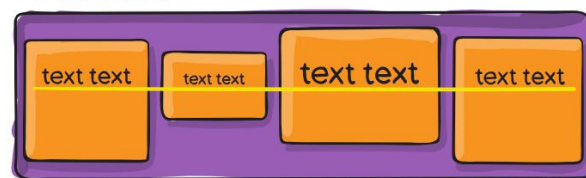
center



stretch



baseline



# Полезные ссылки

[Таблица атрибутов](#)

[Список блочных элементов](#)

[О строчных элементах](#)

[Подробнее об единицах измерения в CSS](#)

Подробнее о подключении шрифтов:

<https://developer.mozilla.org/ru/docs/Web/CSS/@font-face>

<https://htmlacademy.ru/blog/boost/tutorial/fonts-loading>

[Полное руководство по flexbox](#)