



Продвинутый HTML

Лекция 3



Семантический HTML

Предназначен для внесения смысла в верстку

```
<body>
  <div class="header">
    шапка сайта, меню
  </div>
  <div class="main-content">
    <div class="section">
      <div class="header-1">Заголовок 1</div>
      <div class="paragraph">Абзац текста 1</div>
    </div>
    <div class="section">
      <div class="header-1">Заголовок 2</div>
      <div class="paragraph">Абзац текста 2</div>
    </div>
  </div>
  <div class="footer">
    контакты в футере
  </div>
</body>
```

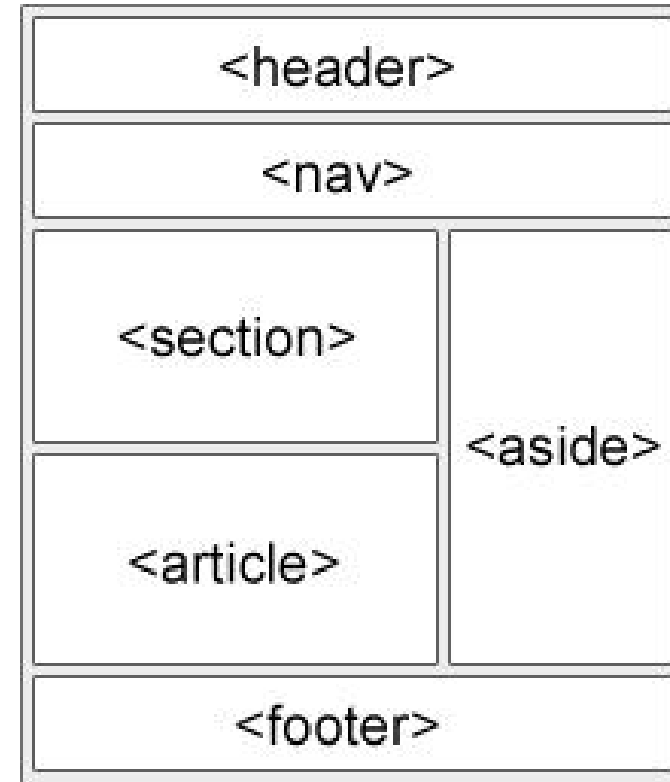
```
<body>
  <header>
    шапка сайта, меню
  </header>
  <main>
    <section>
      <h1>Заголовок 1</h1>
      <p>Абзац текста 1</p>
    </section>
    <section>
      <h1>Заголовок 2</h1>
      <p>Абзац текста 2</p>
    </section>
  </main>
  <footer>
    контакты в футере
  </footer>
</body>
```

Зачем это нужно?

- Помочь роботам, поисковикам интерпретировать содержание страницы (в том числе и *SEO*)
- Помочь слабовидящим работать с сайтом через программы-читалки
- Помочь коллегам-разработчикам воспринимать ваш код
- Следование стандартам

Семантические теги

- **`<article>`** — независимый контент (пост, статья).
- **`<aside>`** — боковая колонка, доп. информация.
- **`<details>`** — скрытый блок по клику.
- **`<figcaption>`** — подпись к **`<figure>`**.
- **`<figure>`** — блок для медиа и подписи.
- **`<footer>`** — подвал сайта/раздела.
- **`<header>`** — шапка сайта/раздела.
- **`<main>`** — основное содержимое страницы.
- **`<mark>`** — выделение текста (жёлтый фон).
- **`<nav>`** — навигационное меню.
- **`<section>`** — логический раздел страницы.
- **`<summary>`** — заголовок для **`<details>`**.
- **`<time>`** — метка времени/даты.



<header>

Не путать с *<head>*!

Предназначен для вводной части смыслового раздела или всего сайта.
Обычно содержит подсказки и навигацию. Чаще всего повторяется на всех страницах сайта.

Этих элементов может быть несколько на странице.

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of
    our planet's natural environment, and build a
    future in which humans live in harmony with nature.</p>
</article>
```

<footer>

Предназначен для заключительной части смыслового раздела или всего сайта, обычно содержит информацию об авторах, список литературы, копирайт и тд.
Чаще всего повторяется на всех страницах сайта.

Этих элементов может быть несколько на странице.
Тег **<footer>** не обязан находиться в конце раздела.

```
<footer>  
  <p>Author: Hege Refsnes</p>  
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>  
</footer>
```

<main>

Предназначен для основного и не повторяющегося на других страницах, содержания.

Должен быть один на странице, исходя из определения.

Типовая ошибка: включать в этот тег то, что повторяется на других страницах (навигацию, копирайты и так далее).

```
1 <header>
2   Бургерочная «Минутка»
3 </header>
4 <main>
5   <h1>0 команде</h1>
6   <p>Мы готовим бургеры с 2011 года. Это долго.</p>
7 </main>
```

<nav>

Предназначен для отображения навигационного раздела со ссылками на другие страницы или другие части страниц (#якоря).

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

A visual representation of a navigation bar. It consists of a horizontal line with a vertical line extending downwards from its left end. To the right of the vertical line, the text "HTML | CSS | JavaScript | jQuery" is displayed, where each word is underlined and separated by a vertical bar.

Используется для основной навигации, а не для всех групп ссылок.

<article>

—Предназначен для независимой, но отделяемой смысловой единицы верстки, например *комментарий, твит, статья, виджет ВК* и тд.

В нём желателен заголовок внутри.

```
1 <article>
2   <h1>Тег <article>, и с чем его едят</h1>
3   <h2>Кратко</h2>
4   <p>
5     Тег <article> обозначает законченный и самодостаточный
6     раздел документа, описывающий какую-то сущность: статью, товар, карточку
7     пользователя и т. д.
8   </p>
9   <h2>Пример</h2>
10  <pre>
11    <code>Рекурсия! 🔥</code>
12  </pre>
13 </article>
```

<section>

Предназначен для отображения смыслового раздела документа.
Однако он **неотделяемый** в отличие от **<article>**.

Здесь тоже желателен заголовок внутри

```
1 <section>
2   <h1>Заголовок первого блока</h1>
3   <p>Съешь ещё этих мягких французских булок, да выпей чаю.</p>
4 </section>
5
6 <section>
7   <h1>Заголовок второго блока</h1>
8   <p>Съешь ещё этих мягких французских булок, да выпей же чаю.</p>
9 </section>
```

<aside>

—Предназначен для отображения побочного или косвенного для страницы контента.

Он может иметь свой заголовок.
Может встречаться несколько раз на странице.

Типовая ошибка: размечать этим тегом основной контент.

```
1 <aside>
2   <h2>Последние комментарии</h2>
3   <ul>
4     <li>Комментарий 1</li>
5     <li>Комментарий 2</li>
6   </ul>
7 </aside>
```

Порядок применения

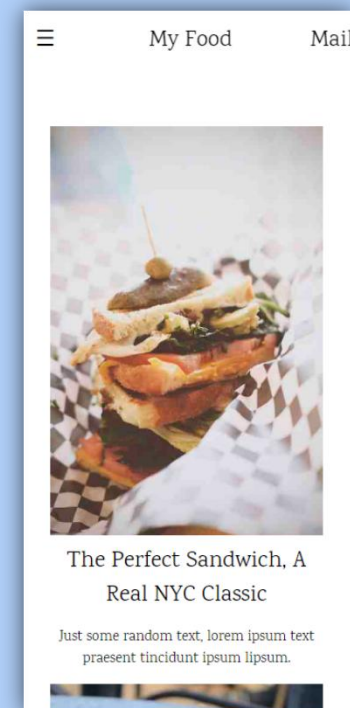
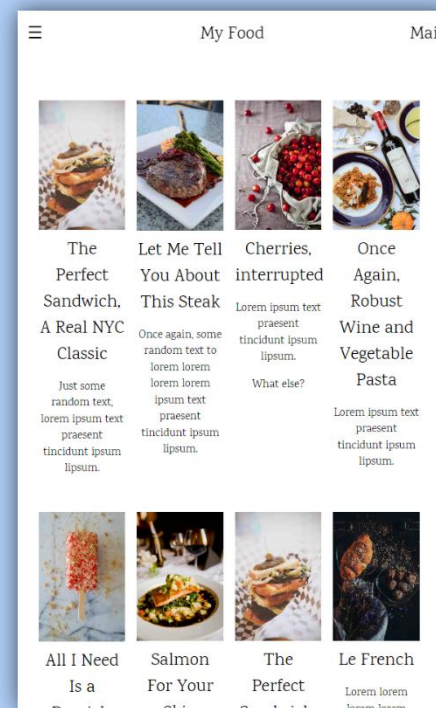
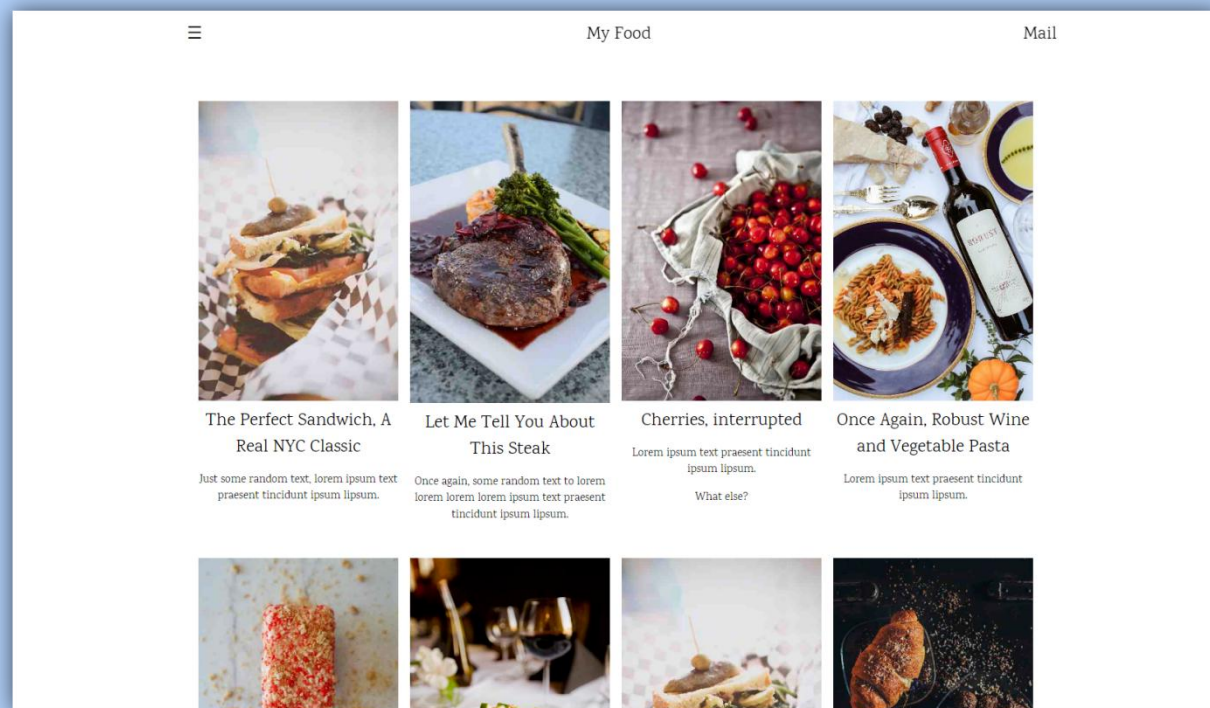
1. Крупные смысловые блоки на каждой странице сайта.
Теги: *<header>*, *<main>*, *<footer>*.
2. Крупные смысловые разделы в блоках.
Теги: *<nav>*, *<section>*, *<article>*, *<aside>*.
3. Заголовок всего документа и заголовки смысловых разделов.
Теги: *<h1>*-*<h6>*.
4. Мелкие элементы в смысловых разделах.
Списки, таблицы, демо-материалы, параграфы и переносы, формы, цитаты, контактная информация и прогресс.
5. Фразовые элементы.
Изображения, ссылки, кнопки, видео, время и мелкие текстовые элементы.

Что такое Layout?

Одно из значений

- Шаблонизированная, неизменная для всех страниц часть сайта.
- Компоновка элементов страницы по дизайнерскому замыслу.

Как выглядит сайт на разных страницах



Подходы к верстке

Float | Clear

Flex

Grid

Фреймворки

FLOAT | CLEAR

Свойство ***float*** обозначает куда элемент прижиматься относительно своего контейнера. А с остальных сторон его можно обтекать.

- ***none*** – по-умолчанию
- ***left***
- ***right***
- ***inherit*** – наследовать от контейнера

Свойство ***clear*** – с какой стороны элемент запрещено обтекать.

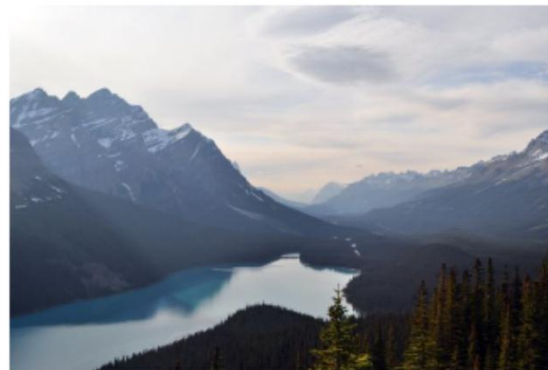
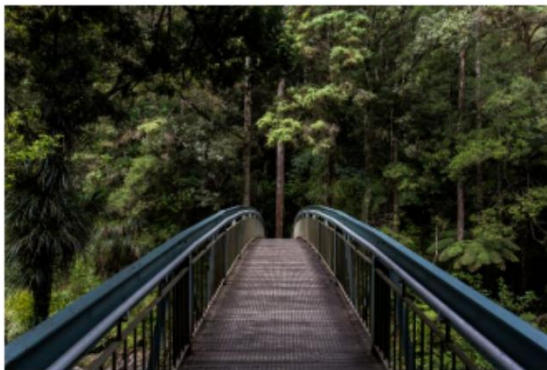
Может отменять ***float***.

- ***none*** - отменяет действие clear
- ***left*** - отменяет обтекание слева, другие элементы на этой стороне будут опущены вниз, и располагаться под текущим
- ***right*** – отменяет обтекание справа
- ***both*** – отменяет обтекание с обеих сторон
- ***inherit*** – наследовать от контейнера

П р и м е р ы

Images Side by Side

Float images side by side:



```
* {  
  box-sizing: border-box;  
}  
.img-container {  
  float: left;  
  width: 33.33%;  
  padding: 5px;  
}  
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```

```
<div class="clearfix">  
  <div class="img-container">  
      
  </div>  
  <div class="img-container">  
      
  </div>  
  <div class="img-container">  
      
  </div>  
</div>
```

Л а й ф х а к

Without Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



With Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



```
.clearfix {  
  overflow: auto;  
}
```

```
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```

Ограниченность данного подхода

Equal Height Boxes

Floating boxes with equal heights:



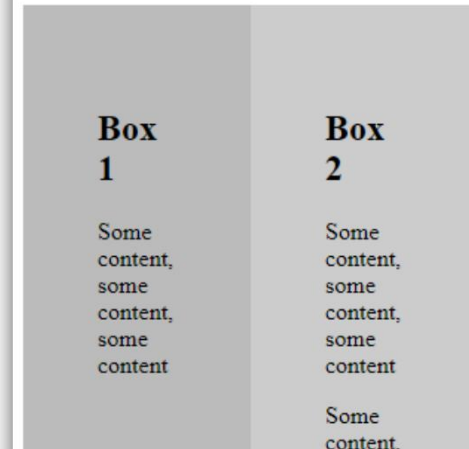
This example not very flexible. It is ok to use CSS height if you can guarantee that the boxes will always have the same amount of content in them, but that's not always the case. If you try the example above on a mobile phone (or resize the browser window), you will see that the second box's content will be displayed outside of the box.

Go back to the tutorial and find another solution, if this is not what you want.

```
.box {  
  float: left;  
  width: 50%;  
  padding: 50px;  
  height: 300px;  
}
```

Equal Height Boxes

Floating boxes with equal heights:



This example not very flexible. It is ok to use CSS height if you can guarantee that the boxes will always have the same amount of content in them, but that's not always the case. If you try the example above on a mobile phone (or resize the browser window), you will see that the second box's content will be displayed outside of the box.

Go back to the tutorial and find another solution, if this is not what you want.

F L E X B O X

Sample flexbox example

First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Third article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

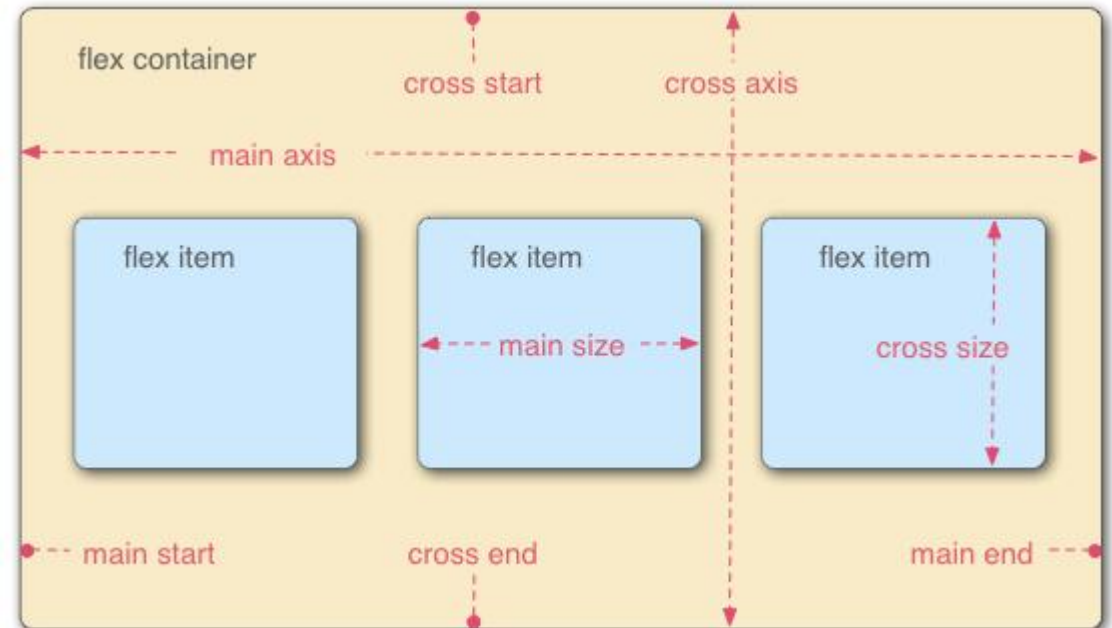
```
<section>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
</section>
```

```
section {
  display: flex;
}
```

Свойства FLEX

Свойство *flex-direction*:

- row
- column
- row-reverse
- column-reverse



Свойства FLEX

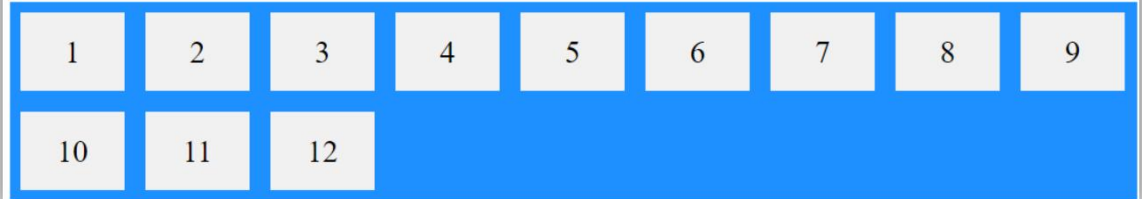
Свойство *flex-wrap*:

- wrap
- nowrap
- wrap-reverse

flex-flow = flex-direction + flex-wrap

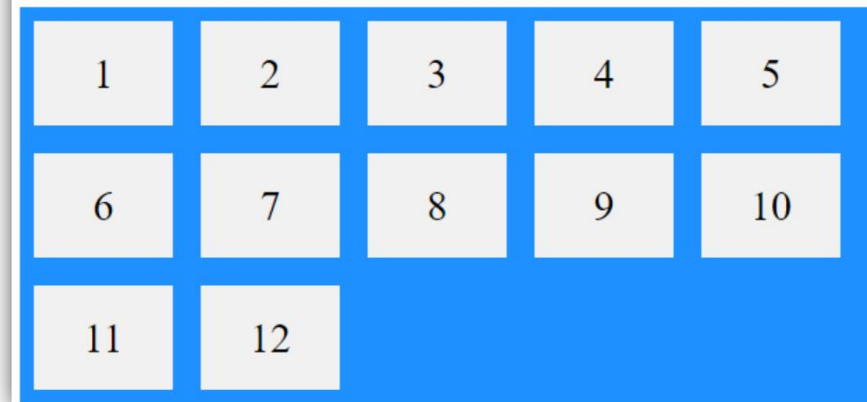
The flex-wrap Property

The "flex-wrap: wrap;" specifies that the flex items will wrap if necessary:



The flex-wrap Property

The "flex-wrap: wrap;" specifies that the flex items will wrap if necessary:



Свойства FLEX

Свойство *justify-content*:

- center
- start | flex-start
- end | flex-end
- left
- right
- stretch
- space-around (вокруг)
- space-between (между)
- space-evenly (одинаково везде)

The justify-content Property

The "justify-content: space-around;" displays the flex items with space before, between, and after the lines:



Свойства FLEX

Свойство *justify-content*:

- center
- start | flex-start
- end | flex-end
- left
- right
- stretch
- space-around (вокруг)
- space-between (между)
- space-evenly (одинаково везде)

The justify-content Property

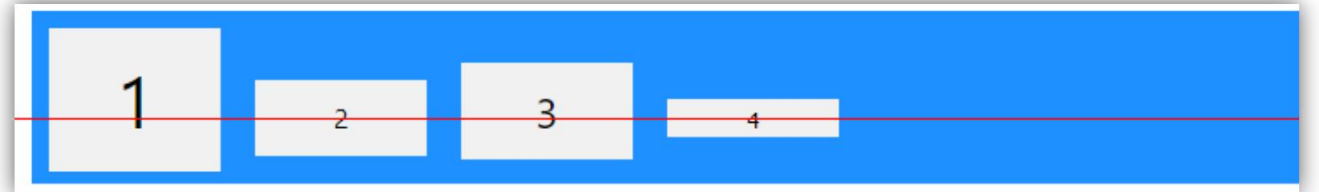
The "justify-content: space-around;" displays the flex items with space before, between, and after the lines:



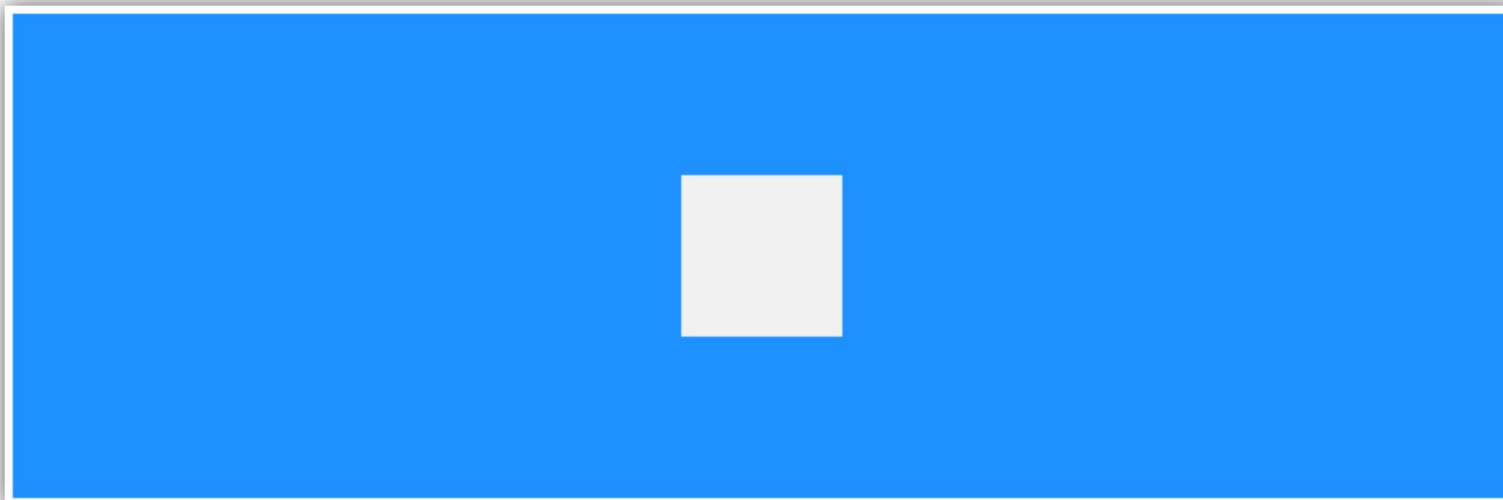
Свойства FLEX

Свойство *align-items*:

- center
- start | flex-start
- end | flex-end
- stretch
- baseline



Идеальное центрирование



```
.flex-container {  
  display: flex;  
  height: 300px;  
  justify-content: center;  
  align-items: center;  
}
```

```
<div class="flex-container">  
  <div></div>  
</div>
```

G R I D

1	2	3
4	5	6
7	8	9

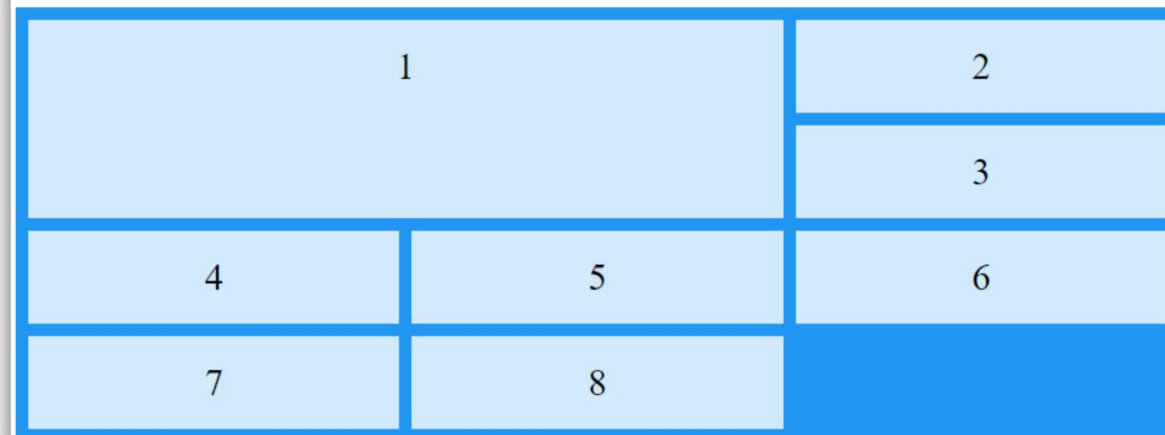
```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

```
.grid-container {
  display: grid;
}
```

Свойства GRID

- grid-template-columns
- grid-template-rows
- grid-gap | column-gap | row-gap
- grid-column-start | grid-row-start
- grid-column-end | grid-row-end

Grid Lines



You can refer to line numbers when placing grid items.

```
grid-template-columns: 200px auto 100px;  
grid-template-columns: 200px auto auto 100px;  
grid-template-columns: 200px repeat(4, 15%);
```

```
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

Так что? FLEX или GRID?

- На основе содержимого
- Удобнее для простых случаев

- На основе содержимого
- Удобнее для простых случаев

Использование фреймворков

Bootstrap

Foundation

W3.CSS

Дополнительная литература



1. [HTML для начинающих: вопросы и ответы / Хабр](#)



1. [Что такое SEO?](#)