

# HTML: Containers

# Повторение;)

Из чего состоит документ?

Что такое тег? Какие бывают теги?

Что такое атрибут?

Что такое элемент?

Какой тег используется для создания гиперссылки?

Какой тег используется для создания нумерованного списка

Какой тег используется для добавления изображения? С какими атрибутами?

# НАШИ ПРАВИЛА




Включенная камера




Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

# ЦЕЛЬ

Изучить способ добавления таблиц, типы ссылок, семантическую верстку

# ПЛАН ЗАНЯТИЯ

- Виды ссылок
- Таблицы в html
- Контейнеры: div, span
- Семантические теги: header, footer, main, section и пр.

# Виды ссылок



## Создание якорной ссылки. Шаг 1

Создадим якорь. Для этого можно воспользоваться одним из двух методов  
1 способ.

```
<p> <a name="закладка"></a></p>
```

2 способ.

```
<p id="закладка">Текст для закладки</p>
```

*Примечание:* якорь — это ссылка внутри страницы на какой-то заголовок или другую часть страницы.

## Создание якорной ссылки. Шаг 2

**Создание элемента ссылки с указанием имени закладки  
через знак шарп “#”**

`<a href="#закладка">Текст ссылки</a>`

*Примечание:* можно сделать ссылку на закладку из другой веб-страницы, тогда она будет иметь следующий вид

`<a href="text.html#закладка">Перейти к закладке на странице text</a>`



# Абсолютные ссылки

Абсолютные ссылки создаются путем добавления в атрибут **href** тега `<a>` абсолютного URL адреса.

Абсолютный URL предполагает указание полного адреса страницы, на которую вы ссылаетесь, включая протокол и доменное имя.

Примеры значения атрибута **href**:

`“https://www.example.com/”`

`“https://www.example.com/courses/”`

# Относительные ссылки

Относительные ссылки создаются путем добавления в атрибут **href** тега **<a>** относительного URL адреса.

Относительный путь означает, что указание пути на нужный файл или страницу вашего сайта начинается относительно каталога, в котором расположена страница со ссылкой, либо относительно корневого каталога сайта

Примеры значения атрибута **href**:

**"main.html"**

**"./flowers.pdf"**

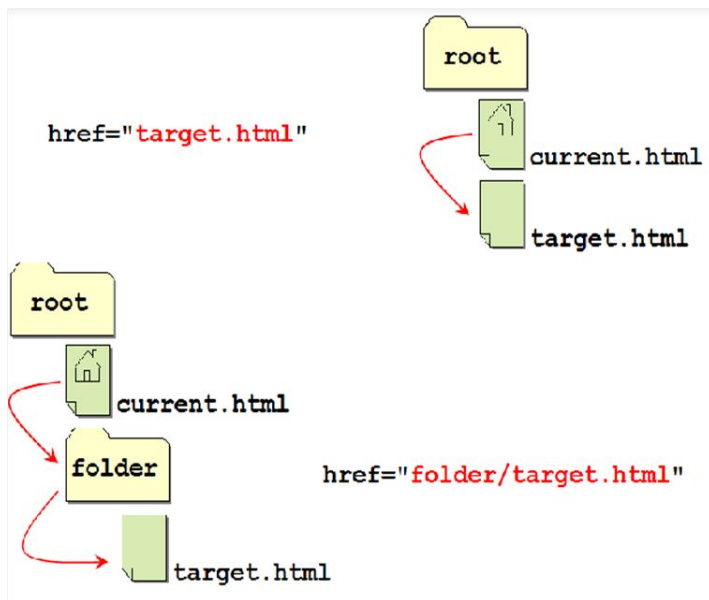
**"file1/main.html"**

**"folder1/folder2/main.html"**

**"../main.html"**

**"../../main.html"**

# Схема адреса относительной ссылки



# Создание таблицы

**<table>**

Sr. No	Roll No	Name	Team
1	1001	John	Red
2	1002	Peter	Blue
3	1003	Henry	Green

**</table>**

## Создание таблицы

1. Содержимое любой таблицы заключается между двумя тегами `<table></table>`
2. Самым маленьким контейнером в таблице является ячейка, она создаётся элементом `<td>` ('td' - сокращение от 'table data')

На данном этапе можем посмотреть промежуточный вариант и понять, что нам делать дальше

### Код

```
<table border="3">  
  <td>Name</td>  
  <td>Age</td>  
  <td>Street</td>  
</table>
```

### Результат

Name	Age	Street
------	-----	--------

## Создание таблицы

3. Чтобы строка перестала расти, а новые ячейки перешли на вторую строку, необходимо использовать элемент `<tr>` ('tr' - сокращение от 'table row'). В него нужно обернуть все ячейки, которые должны быть в этой строке

### Код

```
<table border="3">  
  <tr>  
    <td>Name</td>  
    <td>Age</td>  
    <td>Street</td>  
  </tr>  
  <tr>  
    <td>Tom</td>  
    <td>19</td>  
    <td>Sesame</td>  
  </tr>  
</table>
```

### Результат

Name	Age	Street
Tom	19	Sesame

# Заголовки столбцов

Чтобы передать ячейке роль заголовка вместо тега `<td>` используем тег `<th>` и заменим всю первую строку

## Код

```
<table border="3">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>Street</th>
  </tr>
  <tr>
    <td>Tom</td>
    <td>19</td>
    <td>Sesame</td>
  </tr>
</table>
```

## Результат

Name	Age	Street
Tom	19	Sesame

## Заголовок таблицы

Тег `<caption>` следует включать в любую таблицу. Где бы вы его ни разместили, его содержимое будет выведено перед таблицей.

Для единообразия и доступности тег заголовка размещают в самом начале — сразу после тега `<table>`.

Примечание: тег `<caption>` по умолчанию выравнивает своё содержимое по центру.

### Код

```
<table border="3">
  <caption>
    Users
  </caption>
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>Street</th>
  </tr>
  <tr>
    <td>Tom</td>
    <td>19</td>
    <td>Sesame</td>
  </tr>
</table>
```

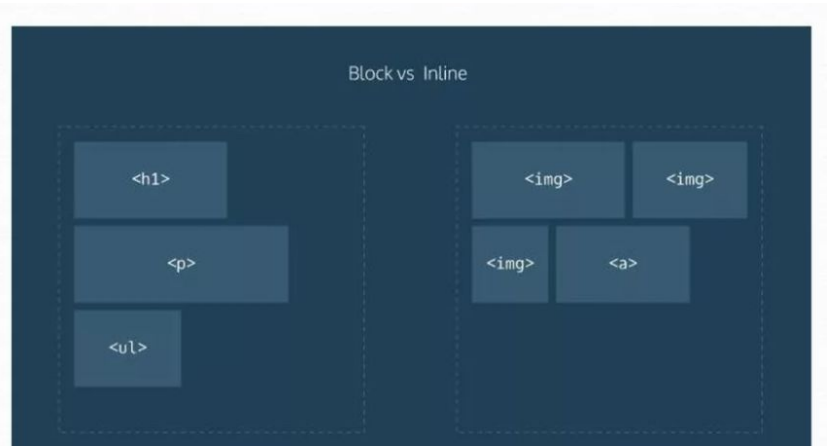
### Результат

Users

Name	Age	Street
Tom	19	Sesame



# Блочный и строчный контейнеры



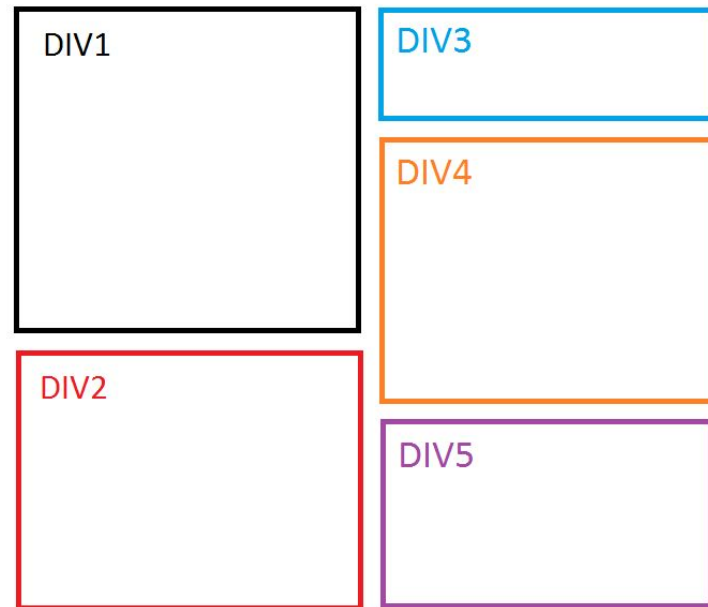
**Блочные элементы** - это элементы, которые занимают всю строку горизонтального пространства на веб-странице. Это означает, что они автоматически начинаются с новой строки, а последующие элементы автоматически переносятся на новую строку.

**Строчные элементы** – это элементы, которые являются частью строки и занимают такое количество пространства, которое необходимо для отображения их содержимого.

# div - блочный контейнер

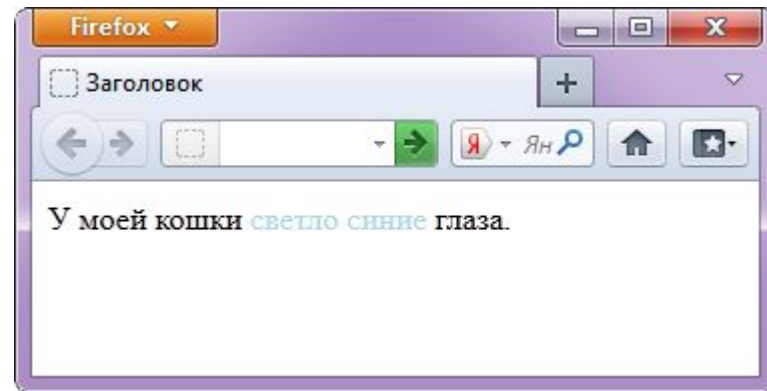
Является универсальным контейнером для контента веб-страницы.

```
<div>Контент</div>
```



## span - строчный контейнер

Тег `<span>` предназначен для определения строчных элементов документа. С помощью тега `<span>` можно выделить часть информации внутри других тегов и установить для нее свой стиль.

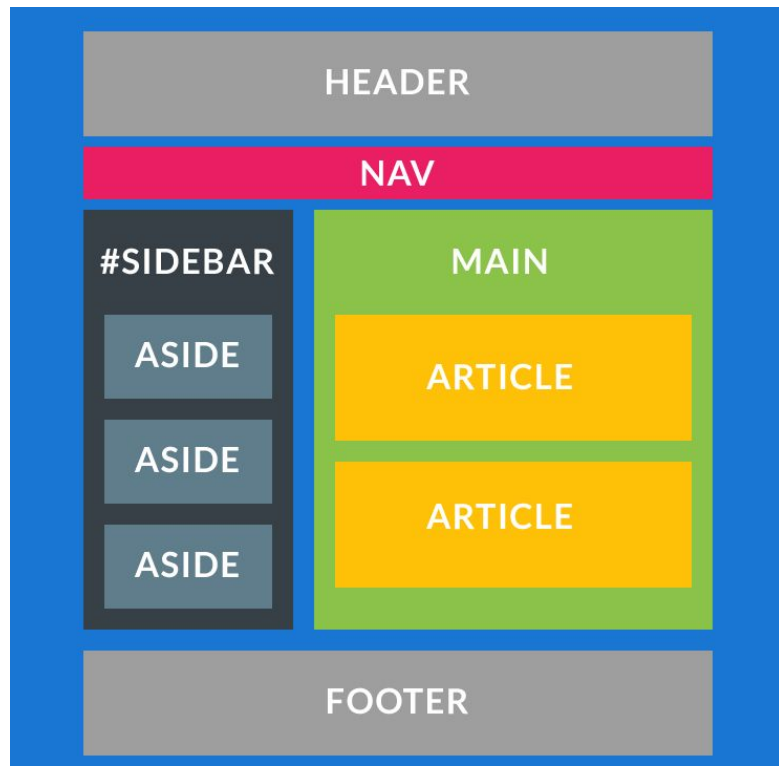


`<span>`Контент`</span>`

# Семантические теги



**Семантическая вёрстка —**  
подход к разметке, который  
опирается не на содержание  
сайта, а на смысловое  
предназначение каждого  
блока и логическую  
структуру документа.



## Для чего нужна семантика:

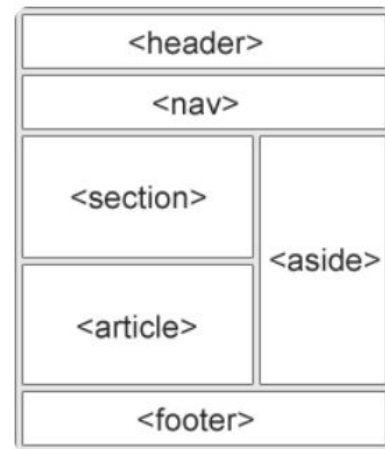
1. Для accessibility\*. Основной инструмент незрячих или частично незрячих людей для просмотра сайтов не браузер, а скринридер, который читает текст со страницы вслух. Этот инструмент «зачитывает» содержимое страницы, и семантическая структура помогает ему лучше определять, какой сейчас блок, а пользователю понимать, о чём идёт речь. Таким образом семантическая разметка помогает большему количеству пользователей работать с вашим сайтом.
2. Чтобы сайт был выше в поисковиках. Поисковики не разглашают правила ранжирования, но известно, что наличие семантической разметки страниц помогает поисковым ботам лучше понимать, что находится на странице, и в зависимости от этого ранжировать сайты в поисковой выдаче.
3. Семантика прописана в стандартах. При работе, например, над поддержкой существующего приложения вам будет проще работать с кодом, написанным по общим стандартам

*\*Accessibility – это область знания, которая занимается изучением вопросов доступности сайтов, мобильных приложений и программного обеспечения для людей с ограниченными возможностями.*



# В html, чтобы выполнить требования семантики используют специальные теги

Тег	Описание
<code>&lt;article&gt;</code>	Определяет статью
<code>&lt;aside&gt;</code>	Определяет содержание в стороне от содержимого страницы
<code>&lt;details&gt;</code>	Определяет дополнительные сведения, которые пользователь может просматривать или скрывать
<code>&lt;figcaption&gt;</code>	Определяет заголовок для элемента <code>&lt;Figure&gt;</code>
<code>&lt;figure&gt;</code>	Задаёт автономное содержимое, например иллюстрации, диаграммы, фотографии, списки кодов и т.д.
<code>&lt;footer&gt;</code>	Определяет нижний колонтитул для документа или раздела
<code>&lt;header&gt;</code>	Задаёт заголовок для документа или раздела
<code>&lt;main&gt;</code>	Указывает основное содержимое документа
<code>&lt;mark&gt;</code>	Определяет выделенный/выделенный текст
<code>&lt;nav&gt;</code>	Определяет навигационные ссылки
<code>&lt;section&gt;</code>	Определяет раздел в документе
<code>&lt;summary&gt;</code>	Определяет видимый заголовок для элемента <code>&lt;Details&gt;</code>
<code>&lt;time&gt;</code>	Определяет дату и время







# **Ваша новая IT-профессия – Ваш новый уровень жизни**

Программирование с нуля в  
немецкой школе AIT TR GmbH