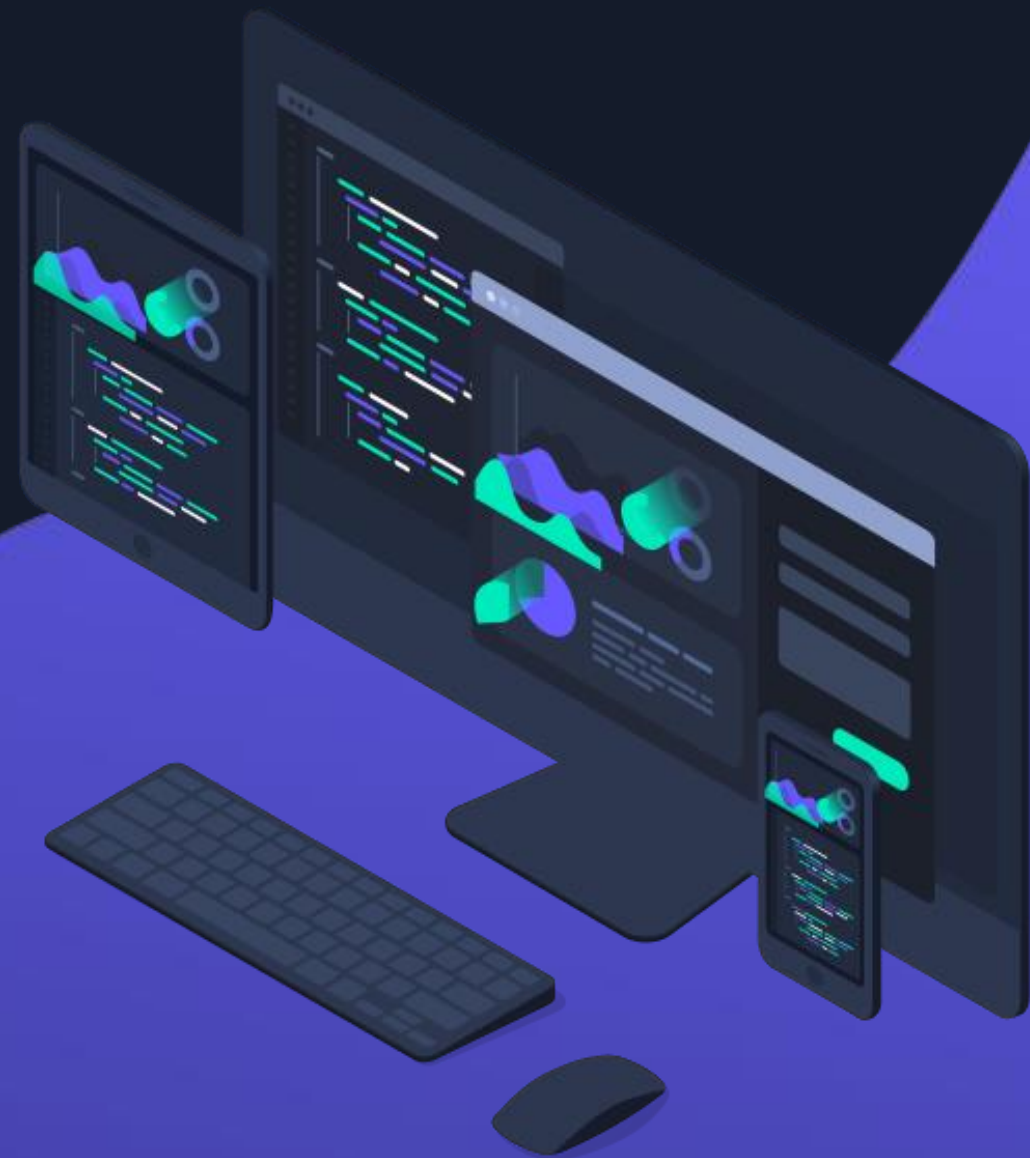


I T E A
O N L I N E



На сегодняшнем занятии:

1

div:позиционирование v2

2

CSS

3

Способы подключения CSS

4

Семантика

5

Селектор класс

6

Селектор идентификатор

7

Контекстные селекторы

8

Другие конструкции селекторов

9

Приоритеты стилей

div: позиционирование v2

Position

— свойство, отвечающее за позиционирование блочных элементов.

Идея позиционирования заключается в вынесении блочного элемента на новый слой по оси Z.

Возможные значения:

absolute;

relative;

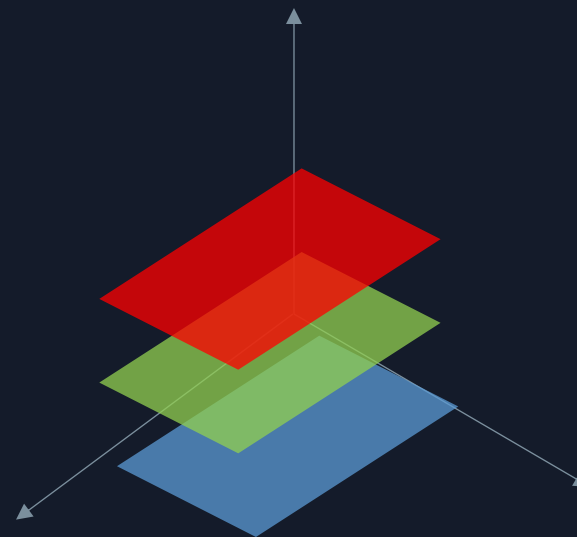
fixed;

static.

Для позиционирования на новой плоскости элемент использует следующие свойства:

top/left/right/bottom;

z-index.



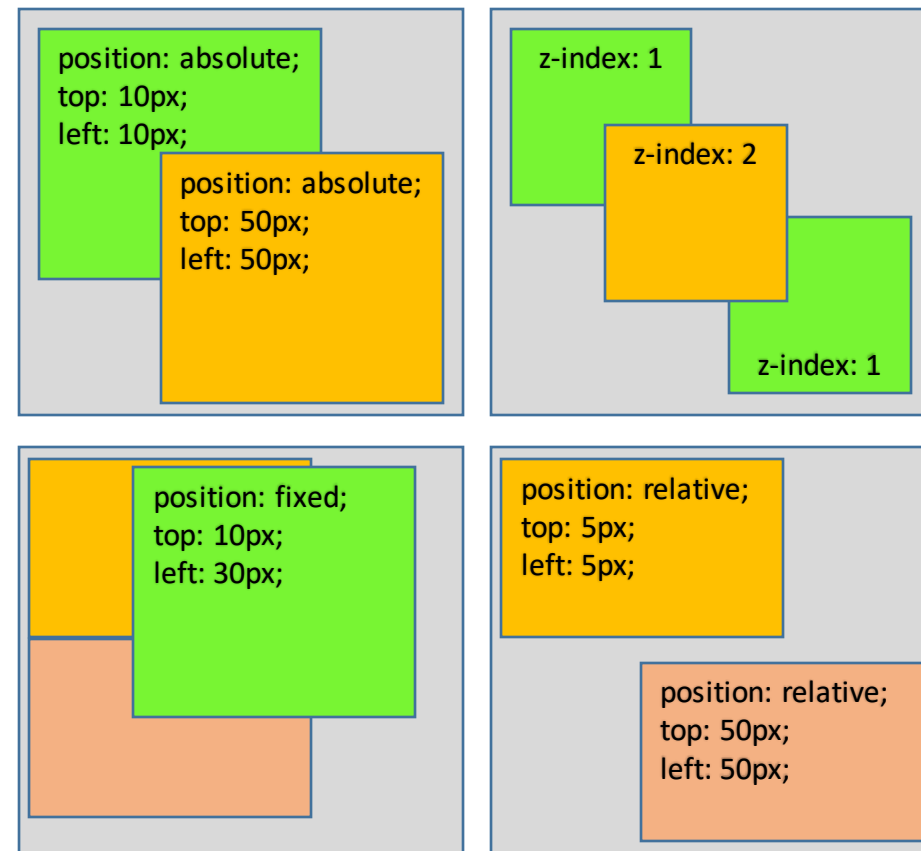
Значения свойства **position**

Значение **absolute** перемещает блочный элемент по оси z на плоскость выше. Место, которое он занимал на прежней плоскости, будет свободным, что повлияет на размещение других элементов.

Значение **relative** говорит блочному элементу переместиться по оси z на плоскость выше, при этом элемент будет позиционироваться относительно прежнего местоположения, и остальные элементы не будут смещаться, так как место не будет свободно.

Значение **fixed** размещает элемент поверх других, при этом позиционирует его относительно границ окна браузера.

При указании абсолютного позиционирования элементов, можно явно указать плоскость, на которую будет перемещен элемент при помощи свойства **z-index**.



CSS

CSS (**Cascading Style Sheets**) — это код, который вы используете для стилизации вашей веб-страницы.

Как и HTML, CSS на самом деле не является языком программирования. **Это не язык разметки** - это язык таблицы стилей. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML.

Первая версия была принята в 1996 году. В ней были доступны возможности изменения шрифта, цвета, атрибутов текста, выравнивание текста и другие.

В 1998 году вышла версия CSS2, в которой появилась возможность блочной верстки.

В 2006 году вышла спецификация CSS3, в которой появилась анимация и возможность использования переменных.



Хокон Виум Ли

Способы подключения CSS

Есть 3 основных способа интегрировать CSS в HTML разметку:

- используя атрибут `style`;
- используя тег `<style>`, размещенный в теге `<head>`;
- через отдельный файл с расширением `.css`, подключенный к странице;

```
<div style="color: red;"></div>
```

```
<style>  
  div {  
    color: red;  
  }  
</style>
```

```
<link href="StyleSheet.css" rel="stylesheet"/>
```

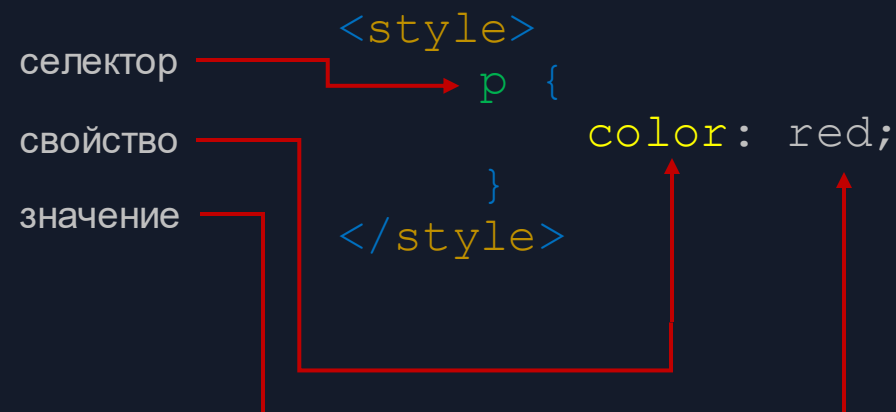
Семантика

В качестве селектора может выступать:

- тег
- отдельный класс
- идентификатор
- либо сложный селектор

Если в качестве селектора указано имя тега, то в этом случае все элементы данного тега будут соответствовать данному стилю.

При этом, если внутри этих тегов будут находиться текстовые элементы, они будут наследовать определенные правила.



Селектор класс

Класс — особенный вид селектора, которым можно пометить как один, так и несколько элементов.

Чтобы указать селектор класс, в CSS необходимо использовать конструкцию **.имяКласса**

Чтобы привязать к этому классу элемент, в открывающем теге необходимо указать атрибут **class = "имяКласса"**.

```
...  
<style>  
    .class1 {  
        color: red;  
    }  
</style>  
...  
<div class="class1">  
    <p>Text</p>  
</div>  
...  
<p class="class1">Test2</p>  
...
```

Text

Test2

Селектор идентификатор

Идентификатор — особенный вид селектора, которым можно пометить один элемент. По умолчанию предполагается, что элемент с таким селектором на странице один.

Чтобы указать селектор-идентификатор в CSS, необходимо использовать конструкцию **#имяИдентификатора**

Чтобы привязать к этому идентификатору элемент, в открывающем теге необходимо указать атрибут **id = "имяИдентификатора"**.

```
...
<style>
    #main {
        background-color: green;
        width: 100px;
        height: 150px;
    }
</style>
...
<div id="main">
    <p>Text</p>
</div>
...
```





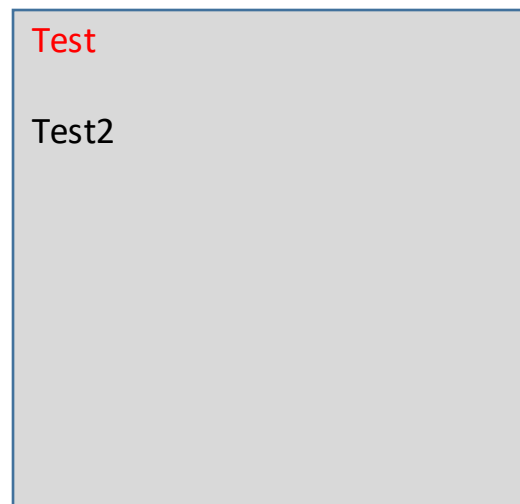
O N L I N E

Контекстные селекторы

Если необходимо определить стиль для элемента, вложенного в другой элемент, то лучшим решением является конструкция вложенных (контекстных) селекторов.

Первым указывается селектор родительского элемента, а вторым селектор дочернего элемента, который будет в него вложен.

```
...  
<style>  
    div p {  
        color: red;  
    }  
</style>  
...  
<div>  
    <p>Test</p>  
</div>  
...  
<p>Test2</p>  
...
```



Другие конструкции селекторов

Соседний селектор.

Необходим для задания стилей для элемента, следующего в коде непосредственно за указанным селектором.

Дочерний селектор.

Необходим для задания стиля для элемента, являющегося прямым потомком указанного селектора.

Селектор атрибута.

Необходим для задания стиля для селектора, который имеет конкретный атрибут.

Универсальный селектор.

Необходим для задания стилей одновременно для всех элементов разметки.

```
<style>
    div + p {
    }

    div > span {
    }

    img[title] {
    }

    * {
    }
</style>
```



O N L I N E

Приоритеты стилей

Элемент HTML может быть целью нескольких правил CSS. Давайте воспользуемся простым абзацем в качестве примера:

```
<p class="msg" id="mates">  
  Hi, dear teammates!  
</p>
```

Мы можем изменить этот абзац просто используя:

```
имя тега   p { color: blue; }  
имя класса .msg { color: green; }  
идентификатор #mates { color: red; }
```

В нашем примере абзац будет **красным**, потому что идентификатор более специфичен и, таким образом, более важен, чем другие селекторы.



O N L I N E

Приоритеты стилей

Порядок правил CSS

Если в вашем CSS есть одинаковые селекторы, то последний из них будет иметь приоритет.

```
p { color: green; }  
p { color: red; }  
/* Абзац будет красным */
```

Вычисление 100

Есть один быстрый способ выяснить, насколько правило CSS «сильное», путём вычисления специфичности селекторов:

- идентификаторы стоят 100;
- классы стоят 10;
- селекторы тега стоят 1.

Селектор с наивысшим «счётом» будет преобладать, независимо от порядка, в котором появляются правила CSS.

```
#introduction { color: red; }  
.message { color: green; }  
p { color: blue; }
```

Правило `#introduction { color: red; }` является более *специфичным*, чем другие, потому что идентификаторы должны быть **уникальными** по всей веб-странице, таким образом, может быть только **один** целевой элемент.

`.message { color: green; }` будет нацелен на *любой* HTML-элемент с атрибутом `class="message"` и, следовательно, менее специфичен. То же самое относится и к `p { color: blue; }`, который может предназначаться для *любого* абзаца.



O N L I N E

Q&A

I T E A
O N L I N E

