

Основы Программирования

Занятие №9
HTML, CSS
События
Взаимодействие с
пользователем



IT Education Academy

WWW.ITEA.UA

План занятия

- CSS, селекторы, свойства
- Атрибуты id и class
- События
- Элемент input
- DOM



Повторение материала

CSS

Язык описания стилей на странице (внешний вид элементов).

В CSS вы указываете правила отображения элементов.

Каждое правило CSS состоит из двух частей:

- селектор
- блок объявлений

Селектор указывает браузеру к каким элементам применимо текущее правило.

В блоке объявлений описывается как отображать элемент.

CSS

Чтобы описать стили на странице, используется элемент `style`. Располагается он внутри элемента `head`.

Все, что пишется внутри элемента `style` - CSS правила.

```
<html>
<head>
  <style>
    //your css goes here
  </style>
</head>
<body>
</body>
</html>
```

CSS Syntax

```
p {
  color: red;
  font-size: 40px;
}
```

p - селектор (все элементы “p”)

color, font-size - свойства (у каждого элемента свои, но есть и много общих)

red, 40px - значения свойств

HTML and CSS

Описать страничку

HTML and CSS

HTML is used for structuring a page. CSS is used for making it look good.

HTML and CSS

```
<html>
<head>
<style>
p { color: red; }
h1 { color: blue;}
</style>
</head>
<body>
  <h1>HTML and CSS</h1>
  <p> HTML is used for structuring a page.
    CSS is used for making it look good.</p>
</body>
</html>
```


Свойства

Можно менять:

- цвета (шрифта и фона)
- изменять границы элементов
- добавлять рамки
- менять шрифты
- расположение на странице
- настраивать анимации
- и тд...

Атрибуты в HTML

```
<tag atr1='value' atr2='value'> content </tag>
```

Есть два особых атрибута, которые широко используются при написании стилей - id и class.

С помощью первого - мы присваиваем уникальное имя для элемента.

С помощью второго - мы «объединяем» элемент в группу, которая обладает одинаковыми свойствами.

Селекторы для id и class

`div { }` //селектор по элементу

`#content { }` // описать стиль для одного элемента - а именно с `id='content'`

`.article { }` //описать стиль для группы элементов у которых `class='article'`

Селекторы для id и class

Описать страницу

HTML and CSS

HTML is used for structuring a page.

CSS is used for making it look good.

Селекторы для id и class

```
<html>
<head>
<style>
#red { color: red; }
#green { color: green; }
h1 { color: blue;}
</style>
</head>
<body>
<h1>HTML and CSS</h1>
<p id='red'> HTML is used for structuring a page.</p>
<p id='green'> CSS is used for making it look good.</p>
</body>
</html>
```

Комбинирование селекторов

В CSS есть возможно комбинировать несколько селекторов.
Например:

`p.red` - выбрать только те элементы `p` у которых есть класс `red`.

События

Когда пользователь взаимодействует со страничкой, он создает определенные события: нажимает на кнопку, пишет текст, двигает мышкой.

С помощью JavaScript есть возможность определенным образом реагировать на эти события — например, менять стили элементов или их содержимое.

Для этого в коде мы подписываемся на события и указываем что должно быть выполнено, когда это событие происходит.

Браузер будет выполнять указанный код если событие произойдет.

События

У каждого элемента существует свой набор событий, которые с ним могут произойти.

Например, на кнопку могут нажать, текст могут выделить и т.д.

Подписка

Подписаться на события можно в HTML коде и в JavaScript коде.

Подписка в HTML

```
<html>
<head>
</head>
<body>
  <button onclick='showMessage()'>Click me!</button>
  <script>
    function showMessage() {
      alert('Hello');
    }
  </script>
</body>
</html>
```

Подписка в JavaScript

```
<html>
<head>
</head>
<body>
  <button id='but'>Click me!</button>
  <script>
    function showMessage() {
      alert('Hello');
    }
    document.getElementById('but').onclick = showMessage
  </script>
</body>
</html>
```

Элемент input

Это пустой элемент, который обладает только атрибутами.
Бывает разных типов:

button	checkbox	color	date
datetime-local	email	file	hidden
image	month	number	password
radio	range	reset	search
submit	tel	text	time

Элемент input

```
<input type='button' value='click me'>  
<input type='text'>  
<input type='number'>
```



The image shows a horizontal row of three input elements. On the left is a button with the text 'click me'. In the middle is a text input field containing the string 'asdf1'. On the right is a number input field containing the number '1234'.

Элемент input

Если обратиться к элементу в JavaScript, то в его свойстве value находится введенное пользователем значение.

```
<input type='text' id='text'>
```

```
var textInputElement =  
    document.getElementById('text');
```

```
var content = textInputElement.value;
```

Элемент `input button`

У элемента `input` также есть событие `onclick`.

```
<input type='text' onclick='doSomething()' value='click this button'>
```

```
-----  
function doSomething() {  
    ...  
}
```

Взаимодействие

Написать страничку

Enter text:

Подтвердите действие

hello, guys

Взаимодействие

```
<html>
<head>
</head>
<body>
    <p> Enter text: <input id='text' type='text'> </p>
    <p> <input type='button' value='click me'
onclick='showMessage()'> </p>
<script>
function showMessage() {
    var text = document.getElementById('text');
    alert(text.value);
}
</script>
</body>
</html>
```

Изменение стиля

У всех элементов есть поле **classList**.

classList - это объект который обладает методами

add(className)

remove(className)

contains(className)

```
.article {  
  color: red;  
}
```

```
<p id='myElement'> Some text </p>
```

```
document.getElementById('myElement').classList.add('article')
```

Изменение стиля

Написать страничку

Some text

Toogle color

Some text

Toogle color

Some text

Toogle color

Изменение стиля

```

<html>
<head>
<style>
.red { color: red; }
.blue { color: blue;}
</style>
</head>
<body>
    <p id='myText' class='red'> Some text </p>
    <p> <input type='button' value='Toogle color'
onclick='toggle()'> </p>
<script>
//on next slide
</script>
</body>
</html>

```

Изменение стиля

```
function toggle() {  
    var text = document.getElementById('myText');  
    if (text.classList.contains('red')) {  
        text.classList.remove('red');  
        text.classList.add('blue');  
    }  
    else {  
        text.classList.remove('blue');  
        text.classList.add('red');  
    }  
}
```

DOM

DOM - document object model.

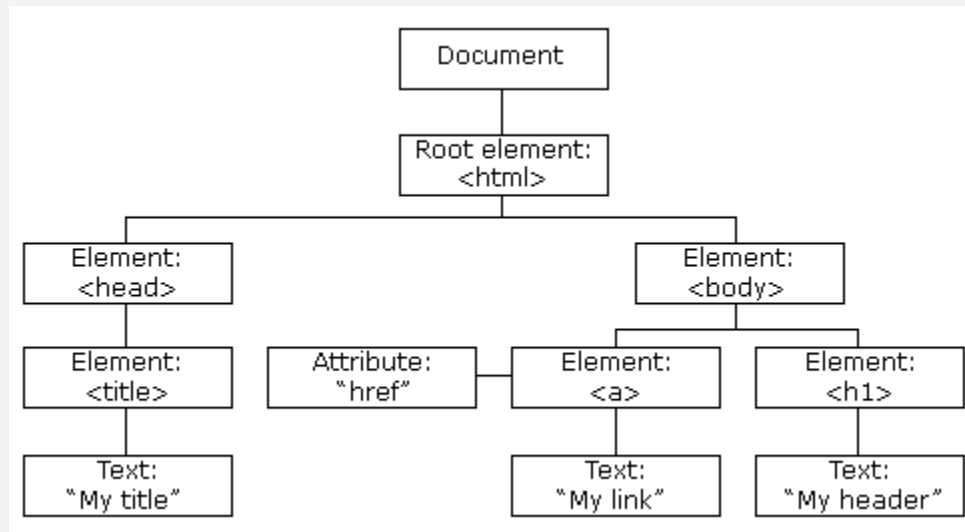
Это внутреннее представление странички в коде JavaScript. С помощью DOM мы имеем возможность взаимодействовать со страничкой - находить элементы, менять их свойства, удалять, создавать новые.

document - это объект который содержит в себе DOM, а метод getElementById находит в DOM нужный элемент.

Когда мы нашли элементы, можно менять его свойства и вызывать его методы. У элементов также есть поле innerHTML - его содержимое. Его можно менять.

```
var element = document.getElementById('text');
element.innerHTML = 'Another text'.
```

DOM





DOM

Поиск элементов:

`getElementById`
`getElementsByClassName`
`getElementsByTagName`

DOM

Создать страничку с 5ю параграфами.

1 и 3 и 4 присвоить класс chosen.

Создать кнопку 'dice'.

По нажати на кнопку случайным образом выбрать среди chosen параграфов один и поменять в нем текст на "I'm the chosen!".

Функция для выбора случайного числа от min до max:

```
function getRandom (min, max) {  
    return Math.random() * (max - min) + min;  
}
```

DOM

DOM позволяет динамически создавать элементы на странице.

```
document.createElement(elementName);
```

```
document.createElement('div');
document.createElement('p');
```

После того как элемент создан он еще не будет отображен на страничке. Его нужно добавить как содержимое для другого элемента.

Для этого есть метод `appendChild(elem);`

Например:

```
var created = document.createElement('input');
document.getElementById('myForm').appendChild(created);
```

DOM

```
<html>
<head>
</head>
<body>
    <p> Enter text <input id='text' type='text'> </p>
    <p> <input type='button' value='Create new element'
onclick='appendElement()'> </p>
```

DOM

```
<script>
var clickedTimes = 0;

function appendElement() {
    if (clickedTimes < 5) {
        var input = document.getElementById('text');
        var newP = document.createElement("p");
        newP.innerHTML = input.value + ' [addedAt=' + new Date() + ']';
        document.body.appendChild(newP);
        clickedTimes++;
    }
    else {
        alert("Cannot add more");
    }
}
</script>
```

События в Node.js

В Node.js также существуют разные события на которые можно соответствующе реагировать.

Например ввод текста в консоле. (до этого мы использовали библиотеку, которая позволяла этого избежать)

События в Node.js

```
var readline = require('readline');

var rl = readline.createInterface({
  'input': process.stdin,
  'output': process.stdout
});

function processLine(line) {
  console.log("User entered " + line);

  if (line == 'exit') {
    rl.close()
  }
}

rl.on('line', processLine)
```

События в Node.js

Помимо первого варианта подписывания на события в Node.js также существуют разные методы, которые выполняют переданный им код по событиям.

```
var http = require('http');
```

```
var times = 0;
```

```
function handleRequest(req, res) {
  res.write('Hello World!');
  res.write(' ' + times);
  times++;
  res.end();
}
```

```
var server = http.createServer(handleRequest)
server.listen(8080);
```

Домашнее задание

1. Создать параграф и описать событие, которое оповещает пользователя что он кликнул на этот параграф. (событие - onclick)
2. Создать страничку с текстовым полем и двумя кнопками. Когда пользователь нажимает на первую, значение в текстовом поле запоминается и сбрасывается (свойству value присвоить пустую строку). Когда пользователь нажимает на вторую кнопку на экран выводится введенный до этого текст.
3. Написать страничку на которой есть параграф и кнопка. По нажатию на кнопку все буквы в тексте становятся большими и красными. Когда нажимаешь снова на кнопку - текст обратно становится как был.

“Hello John. How are you?” -> “**HELLO JOHN. HOW ARE YOU?**” -> “Hello John. How are you?”

**** Написать интерфейс для задания 3 лекция 8.**

Главная задача странички - добавлять книгу и искать по заголовку и автору среди добавленных.

Спасибо за внимание!