

Компания «КОРЗИЛЛА»



День 3



JavaScript

План

1. Определения JavaScript
2. Для чего используется
3. Переменные
4. Подключение к сайту
5. Обращение к элементам страницы
6. Самообучение

О JavaScript

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.



Примеры использования

1. Добавление различных анимационных эффектов, которые не представляется реализовать с помощью HTML и CSS. (Например анимация fitgrad.ru)
2. Возможность изменения содержимого элементов HTML и добавления новых тегов без перезагрузки страницы.
3. Реагирование на события (реагирование на действия пользователя) – обработка нажатий клавиш на клавиатуре и передвижения курсора мыши.
4. Выполнение проверки данных, которые были введены в поля формы перед отправкой на сервер. Это уменьшает нагрузку на сервер и ускоряет работу сайта.
5. Определение версии браузера и загрузка требуемой страницы в зависимости от этого.

И многое другое.



Подключение

```
<script type="text/javascript" >
```

```
... // КОД
```

```
</script>
```

Или

```
<script type="text/javascript" src="jquery.min.js"></script>
```

Тестирование

The image shows a browser window with a testing overlay. The overlay consists of several colored boxes: a blue header box containing the text "Название сайта" (Site Name), "Слоган" (Slogan), and "(8552) 12-34-56"; a red box below it containing two underlined links: "Первая ссылка меню" (First menu link) and "Вторая ссылка меню" (Second menu link); a white box on the left containing the text "Заголовок первой статьи" (Title of the first article) and "Абзац первой статьи" (Paragraph of the first article); and a light red box on the right containing the text "Поиск по сайту" (Search on site), a search input field, and a "Найти" (Find) button. In the top right corner of the browser window, the dimensions "671px x 236px" are displayed. At the bottom, the browser's developer console is open, showing the "Console" tab with the following log entries:

```
> alert('dddd');  
< undefined  
>
```

Переменная

Переменная — поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным и изменять значение в ходе выполнения программы

Объявление (создание) переменной:

```
var ИмяПеременной;
```



Типы данных

Null (*ничто*)

Undefined (неопределенный)

Number (Число)

String (строка, текст)

Boolean (истина или ложь)

Object (набор данных)

Условия

```
if (условие) {
```

```
    // код выполнится, если условие удовлетворяет
```

```
} else {
```

```
    // код выполнится, если условие удовлетворяет
```

```
}
```

Циклы

Цикл for

- Циклу for в круглых скобках передаются 3 аргумента:
- Переменная цикла с начальным значением.
- Условие входа в цикл.
- Действие, выполняемое после каждого прохождения тела цикла.

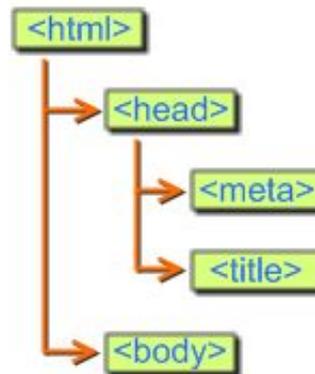
Цикл while

Простая конструкция: передается всего один аргумент - условие входа в цикл.

Дерево DOM

```
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
  </head>  
  <body>  
  </body>  
</html>
```

В виде дерева эта страница может быть представлена так:



Обращение к элементам страницы

- Элементы верхнего уровня
- Поиск элементов
- Обращение к родственным узлам

Элементы верхнего уровня

К элементам верхнего уровня можно обратиться напрямую, используя свойства объекта `document`:

`document.documentElement` - для обращения к элементу `<html>`.

`document.head` - для обращения к элементу `<head>`.

`document.body` - для обращения к элементу `<body>`.

Поиск элементов

`getElementsByName()`

`getElementsByClassName()`

`getElementsByTagName()`

`querySelectorAll, querySelector()`

Пример использования:

`document.getElementsByName(имя)`



Родственные узлы

- **parentNode, parentElement**
- **childNodes, firstChild, lastChild**
- **previousSibling, nextSibling**

Действия с атрибутами тэгов

hasAttribute(атрибут)

Проверяет наличие атрибута.

getAttribute(атрибут)

Возвращает строку со значением атрибута.

setAttribute(атрибут, значение)

Создаёт атрибут или присваивает ему новое значение.

removeAttribute(атрибут)

Удаляет атрибут.

Самостоятельное изучение

learn.javascript.ru

[codecademy.com](https://www.codecademy.com)

на английском, выбрать курс по javascript



Спасибо.

Вопросы?

