

# DOM. Работа со страницей

Раман Крот

**АДУКАР**

# Повестка дня

1. DOM, BOM
2. Методы работы со страницей
3. Получение и добавление элементов
4. Способы изменения элементов





# Дерево DOM

- DOM (Document Object Model) – внутреннее представление HTML-страницы в виде дерева
- Все элементы страницы, включая теги, текст, комментарии, являются узлами DOM
- У элементов DOM есть свойства и методы, которые позволяют изменять их

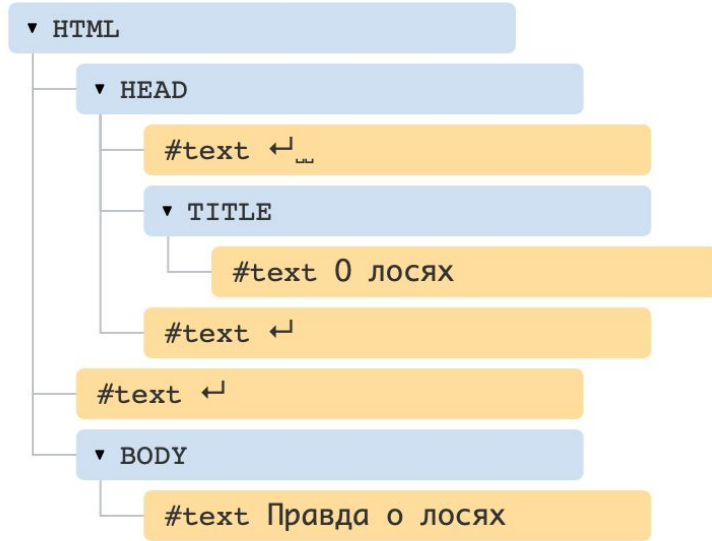
DOM - это представление документа в виде дерева объектов, доступное для изменения через JavaScript.

# Пример DOM

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>0 лосях</title>
5 </head>
6 <body>
7   Правда о лосях
8 </body>
9 </html>
```

© javascript.ru

# Пример DOM



© javascript.ru

# BOM

- BOM - объектная модель браузера
- window: navigation, location, history, screen, frames

Большинство возможностей BOM стандартизированы в **HTML 5**, хотя различные браузеры и предоставляют зачастую что-то своё, в дополнение к стандарту.

# Методы работы со страницей

- Доступ к DOM начинается с объекта `document`
- Самые верхние элементы дерева - `document.documentElement`, `document.body`
- Дочерние элементы - `childNodes`, `firstChild`, `lastChild`
- Соседние элементы - `previousSibling`, `nextSibling`
- Родитель доступен по `parentNode`

# Получение элементов

**document.getElementById** - по id

**document.getElementsByClassName** - по имени класса

**document.getElementsByTagName** - по имени тэга

**document.querySelectorAll** - по css селектору



# Добавление элементов

`document.createElement(tag)` - создает элемент

`document.createTextNode(text)` - создает текстовый узел

`elem.cloneNode(deep)` - клонирует узел со всеми потомками , если `deep === true`, или без потомков

`parent.appendChild(elem)` - добавляет `elem` в конец дочерних элементов

`parent.insertBefore(elem, nextSibling)` - добавляет `elem` перед `nextSibling`

# Практика

```
<!DOCTYPE HTML><html><head> <meta charset="utf-8"></head><body> <form
name="search"> <label>Поиск по сайту: <input type="text" name="search"> </label> <input
type="submit" value="Искать!"> </form> <hr> <form name="search-person"> Поиск по
посетителям: <table id="age-table"> <tr> <td>Возраст:</td><td id="age-list"> <label> <input
type="radio" name="age" value="young">до 18</label> <label> <input type="radio" name="age"
value="mature">18-50</label> <label> <input type="radio" name="age" value="senior">более
50</label> </td></tr><tr> <td>Дополнительно:</td><td> <input type="text" name="info[0]">
<input type="text" name="info[1]"> <input type="text" name="info[2]"> </td></tr></table> <input
type="submit" value="Искать!"> </form></body></html>
```

Найдите (получите в переменную) на странице:

# Практика

1. Все элементы `label` внутри таблицы. Должно быть 3 элемента.
2. Первую ячейку таблицы (со словом "Возраст").
3. Вторую форму в документе.
4. Форму с именем `search`, без использования её позиции в документе.
5. Элемент `input` в форме с именем `search`. Если их несколько, то нужен первый.
6. Элемент с именем `info[0]`, без точного знания его позиции в документе.
7. Элемент с именем `info[0]`, внутри формы с именем `search-person`.

# Удаление элементов

`parent.removeChild(elem)` - удаляет дочерний элемент

`parent.replaceChild(newElem, elem)` - заменяет `elem` на `newElem`

`elem.remove()` - удаляет напрямую, не требует родителя (IE11-)



# Способы изменения элементов

**`elem.setAttribute(name, val)`** - устанавливает атрибут

**`elem.className`** - свойство имени класса элемента

**`elem.href`, `elem.src`, `elem.alt`, `elem.name`** - специфические свойства элементов

**`elem.hidden`** - свойство `hidden`

# Практика

Нужно создать пустой html документ и заполнить его используя различные методы работы с документом. Тело документа должно содержать следующий код:

```
<ul>  
  <li><a href="http://google.com">http://google.com</a></li>  
  <li><a href="/tutorial">/tutorial.html</a></li>  
  <li><a href="local/path">local/path</a></li>  
  <li><a href="ftp://ftp.com/my.zip">ftp://ftp.com/my.zip</a></li>  
  <li><a href="http://nodejs.org">http://nodejs.org</a></li>  
  <li><a href="http://internal.com/test">http://internal.com/test</a></li>  
</ul>
```

Сделайте красными все внешние ссылки (имеющие начало с http://), поменяйте местами два произвольных элемента списка

# Внеклассное чтение

<https://learn.javascript.ru/document> (пункты 1-8)