

HTML - Урок 15

Работа с объектами DOM HTML



Геннадий Караченцев

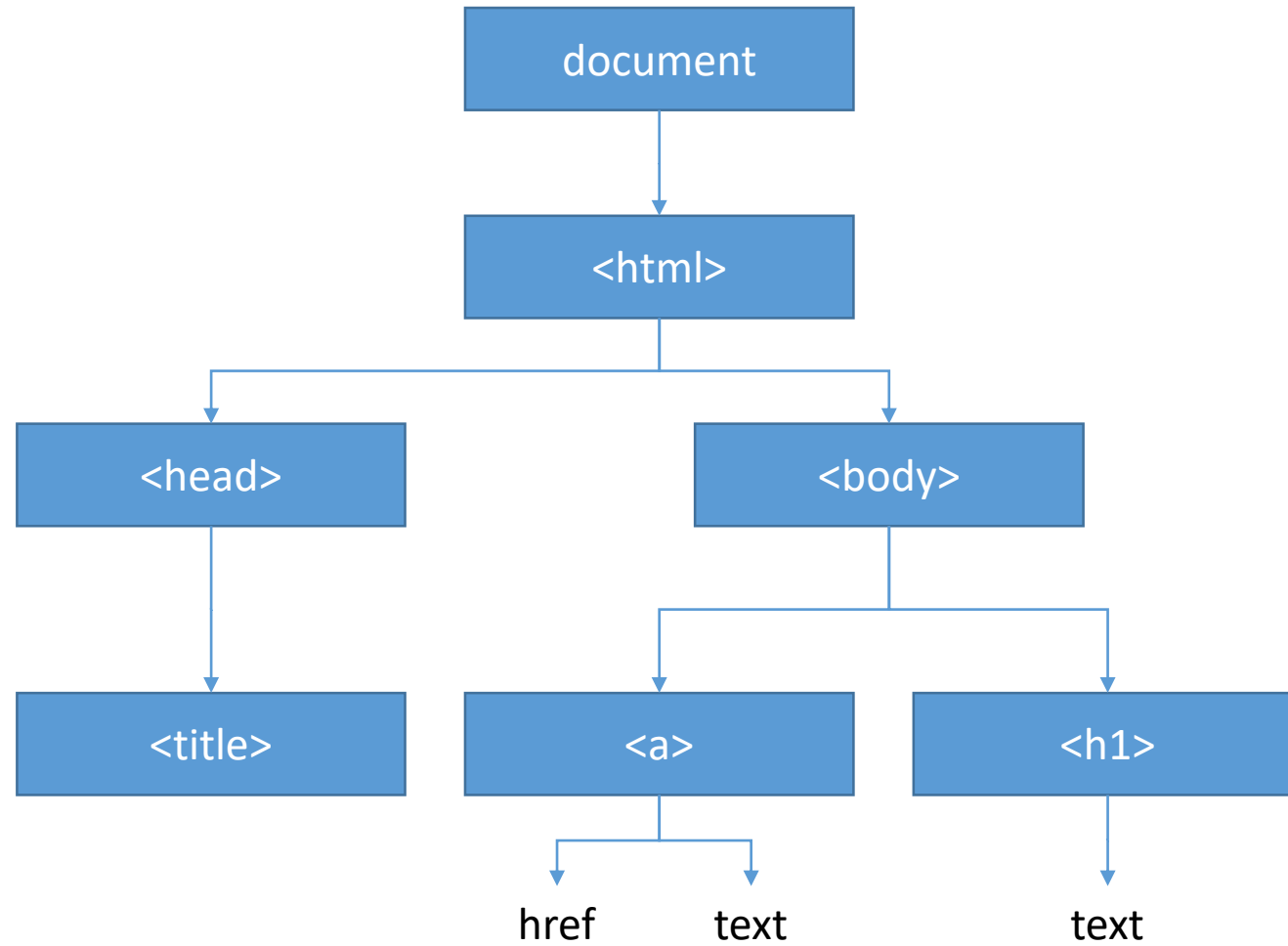
www.weblaboratory.in.ua

Что такое DOM?

DOM — объектная модель документа.

Понятие DOM в HTML, это по сути представление любого элемента(тега) верстки в виде объекта, у которого есть свои свойства, функции и методы.

DOM



Что можно делать в DOM?

- Создавать новые элементы
- Удалять элементы
- Редактировать существующие
- Изменять, добавлять и удалять CSS стили
- Обрабатывать события пользователя
- Читать положение мыши
- Работать с клавиатурой
- и многое другое...

Получение элемента

Чтоб начать работать с элементом как с объектом, необходимо в начале получать ссылку на этот объект. Для этого, необходимо построить селектор, который будет указывать на тот или иной элемент и возвращать нам ссылку на объект элемента.

- `getElementById` // Найти элемент по `id`
- `getElementsByTagName` // Найти элемент по названию тега
- `getElementsByClassName` // Найти элемент по `class`
- `querySelectorAll` // Найти элемент по CSS селектору

Пример

```
<h1 id="intro">Привет мир!</h1>
```

```
var myElement = document.getElementById("intro");
```

Основные свойства DOM объекта

```
var myElement = document.getElementById('Element');
```

```
myElement.innerHTML // Получить/заменить данные внутри тега
```

```
myElement.style.nameOfStyle // Получить/добавить/изменить свойство CSS
```

```
myElement.attribute // Получить/добавить/изменить значение атрибута тега
```

Для

```
myElement.src // Получить/изменить значение атрибута src у <img>
```

Для <a>

```
myElement.href // Получить/изменить значение атрибута href у <a>
```

Создание элементов в DOM

Иногда во время создания скриптов, нам необходимо добавлять свои HTML теги и кусочки вёрстки, в уже готовый код. Будь то сообщения в тексте или еще, что-либо.

```
var div = document.createElement('div');  
var p = document.createElement('p');  
var text = document.createTextNode('Привет мир!');  
  
p.appendChild(text); // Добавит в параграф как дочерний элемент текст  
div.appendChild(p); // Добавит в див как дочерний элемент параграф с текстом  
document.body.appendChild(div); // Добавить див, с дочерним параграфом и текстом в body
```

Работа с CSS стилями в DOM

Это часто используемая возможность в DOM, особенно после появления анимации и transition в CSS, когда подставляя новые значения стиля, можно создавать анимационные эффекты.

```
var div = document.getElementById('div');  
div.style.nameOfStyle; // стили у которых есть разделитель, пишутся через camelCase
```

Пример

```
div.style.borderRadius;  
div.style.backgroundColor;  
div.style.borderTop;
```


События в DOM

Очень часто в JS приходится обрабатывать события, сделанные пользователем. Наведение мышью, клик по объекту или нажатие клавиши.

Для этого используются так называемые «слушатели», чтобы отлавливать событие и производить какие-либо действия.

Либо можно воспользоваться атрибутом события для тегов.

```
<h1 onclick="changeText(this)">Click on this text!</h1>
```

```
function changeText(id) {  
    id.innerHTML = "Oops!";  
}
```

Передача функции через атрибут

```
<button id="myBtn">Клихни на меня</button>
```

```
<p id="demo"></p>
```

```
document.getElementById("myBtn").onclick = displayText;
```

```
function displayText() {  
    document.getElementById("demo").innerHTML = 'Привет, я клик от кнопки!';  
}
```

Важны момент!!!

Замете мы передаем в свойство `onclick` функцию без скобок, только название, т.к. если бы передали со скобками, то она выполнялась бы, на этапе передачи в свойство `onclick`.

Слушатели событий в DOM

Но все же, чаще всего используются «слушатели» событий, без вмешательства в значение атрибутов и т.д.

onmouseout Курсор покидает элемент.

onmouseover Курсор наводится на элемент.

onmouseup Левая кнопка мыши отпущена.

onreset Форма очищена.

onselect Выделен текст в поле формы.

onscroll Событие **onscroll** происходит при прокручивании содержимого веб-страницы.

onsubmit Форма отправлена.

onunload Заккрытие окна.

onblur Потеря фокуса.

onchange Изменение значения элемента формы.

onclick Щелчок левой кнопкой мыши на элементе.

ondblclick Двойной щелчок левой кнопкой мыши на элементе.

onfocus Получение фокуса.

onkeydown Клавиша нажата, но не отпущена.

onkeypress Клавиша нажата и отпущена.

onkeyup Клавиша отпущена.

onload Документ загружен.

onmousedown Нажата левая кнопка мыши.

onmousemove Перемещение курсора мыши.

Реакция на клик

```
<div id="myDiv">Кликни на меня</div>
```

```
var div = document.getElementById("myDiv");
```

```
div.addEventListener('click', hello);
```

```
function hello() {  
    alert('Привет');  
}
```

Важны момент!!!

Вторым параметром мы добавляем имя функции, которую нужно вызвать по клику, без скобок!

Безымянная фикция

Безымянная функция не имеет имени и не вызывается сразу, но может быть задана, как функция которая должна выполняться когда ее вызовет событие.

```
<div id="myDiv">Кликни на меня</div>
```

```
var div = document.getElementById("myDiv");
```

```
div.addEventListener('click', function(element) {  
    console.log(element); // вернет объект события  
    console.log(this); // вернет объект на котором сработало событие  
});
```

Остановить процесс

Иногда нам необходимо сделать свое событие, для элементов у которых уже предусмотрено выполнение при определенном действии. Например при клике на ссылку, браузер переносит нас по ссылке или при отправке формы, форма отправляет данные по адресу который написан в action.

```
preventDefault(); // останавливает событие по умолчанию
```

```
var form = document.getElementById('form1');
```

```
form.addEventListener('submit', function(evt) {  
    alert('Я остановил отправку формы!')  
    evt.preventDefault();  
});
```