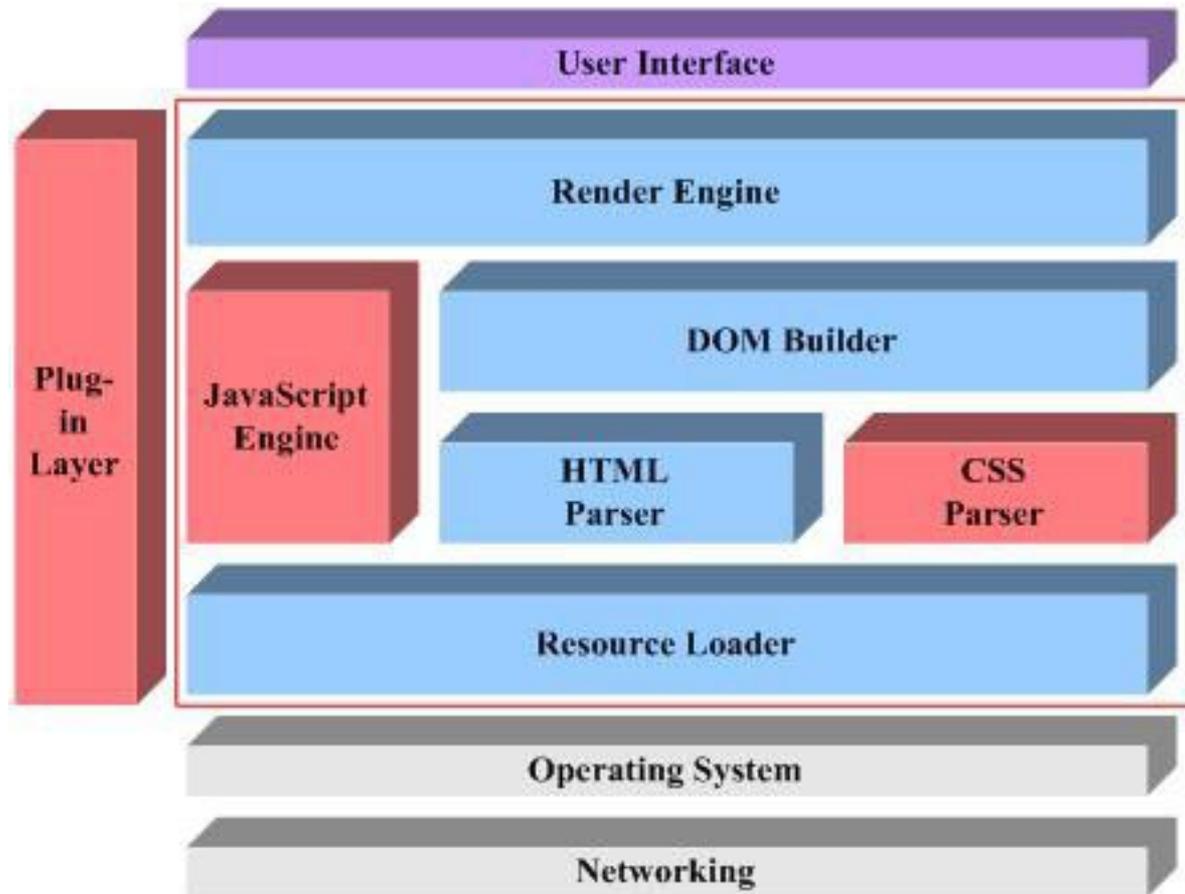


# ОСНОВЫ JS (7)

# DOM



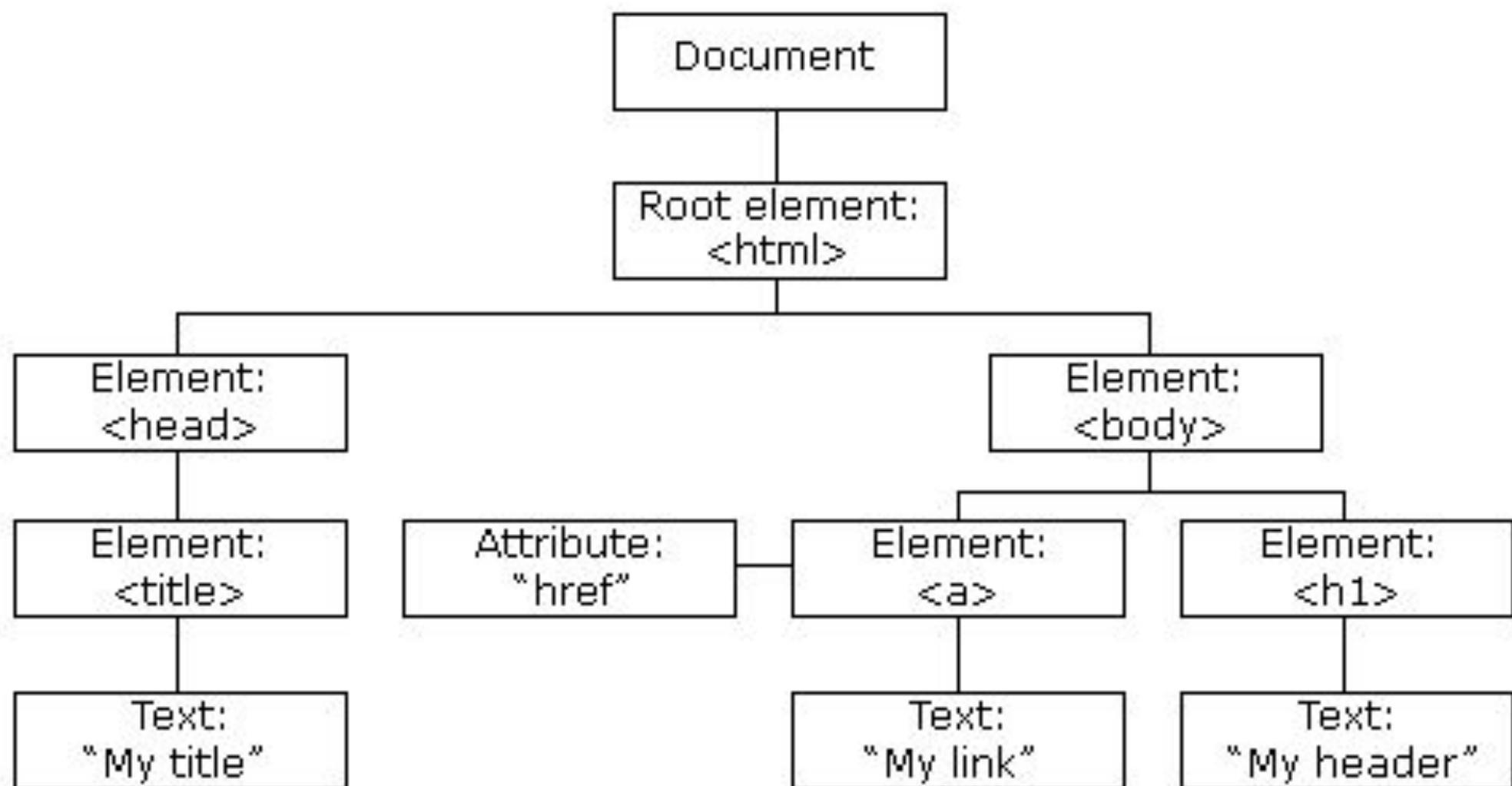
# DOOM

**DOM Builder** – компонента которая создает объектную модель документа (дерево узлов) - интерфейс позволяющий программам и скриптам получить доступ к содержимому HTML, XML документов, а также изменять содержимое и оформление таких документов.

# JS и DOM

Может быть использован для :

- Синтаксический анализ HTML.
- Загрузка внешних сценариев и таблиц стилей.
- Полное построение и управление DOM
- Загрузка изображений, данных, AJAX
- Обработка различных событий
- Установка и изменение свойств объектов (прозрачность/положение)
- Прочие манипуляции...



# DOM

- В **DOM** все элементы – это **объекты**
- **Свойство элемента** – значение, которое можно прочитать или задать
- **Метод** – действие, которое можно сделать (с объектом)

# Рассмотрим пример

```
<html>
```

```
  <body>
```

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

```
    <script>
```

```
document.getElementById("demo").innerHTML =  
"Hello World!";
```

```
    </script>
```

```
  </body>
```

```
</html>
```

# DOM и JS

```
<html>
```

```
  <body>
```

```
    <script>
```

```
document.getElementById("demo").innerHTML =  
"Hello World!";
```

```
    </script><!--Почему??-->
```

```
    <p id="demo">Тест!</p>
```

```
  </body>
```

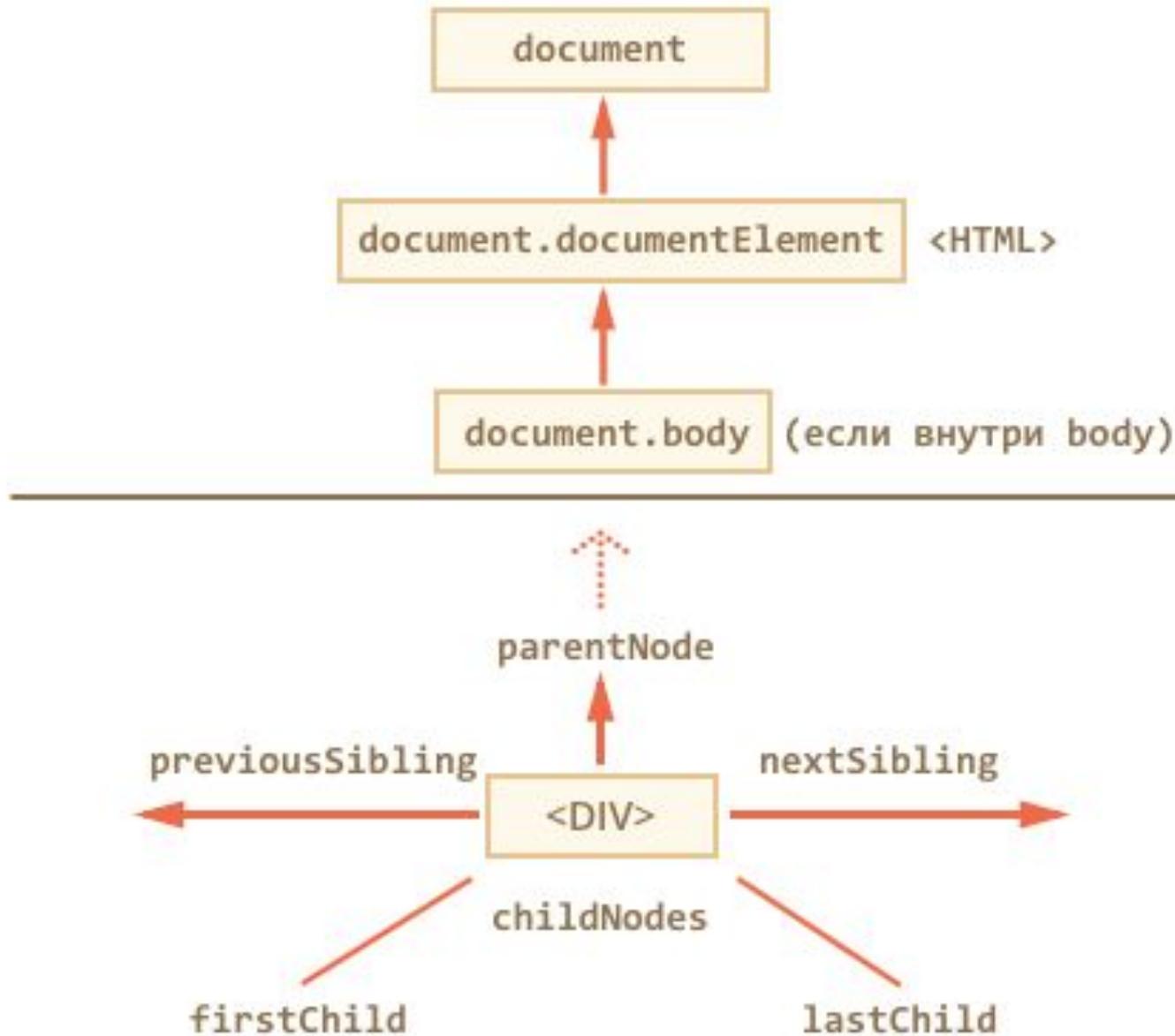
```
</html>
```

# Методы и свойства

- `getElementById()` – метод
- `innerHTML` – свойство

InnerHTML – наипростейший способ управления содержимым HTML элемента (подходит для `<html>` и `<body>`)

# Навигация...



# "document"

- Объект `document` – это Ваша веб страница
- Доступ к любому элементу необходимо начинать с объекта `"document"`
- Далее мы рассмотрим простейшие методы поиска и изменения элементов

# Поиск HTML элементов

- `document.getElementById(id)` – позволяет найти элемент по уникальному `id`.
- ! Убедитесь, что `id` элемента уникален
- `document.getElementsByTagName(name)` – находит элементы по "тегу"
- `document.getElementsByClassName(name)` – находит элементы по названию класса
- `document.getElementsByName(name)` – находит элементы по имени (`name`)

# Важно!

- `document.getElementsByTagName('p')`,  
`document.getElementsByClassName('mine')` и  
`document.getElementsByName('age')`  
возвращают коллекцию, к элементам  
которой можно обращаться по индексу  
`element[0].innerHTML`

# «Павел Александрович»

```
<html>
<body>
<div>
<p class='mine' id="demo">- Привет,</p>
<p class='mine' id="extra">я Павел
Александрович.</p>
<p class='mine' >Мне</p>
<p class='mine2' name='age'>18</p>
</div>
</body>
</html>
```

# Немножко «не JQuery»

!!! Это работает в консоли без подключения JQuery. Для работы в коде нужно подключать!

Также есть отличия по синтаксису

\$ (“selector”) – CSS селектор до первого элемента

\$\$ (“selector”) – CSS селектор массива всех элементов

# \*\* CSS селекторы

Основных видов селекторов всего несколько:

\* – любые элементы.

`div` – элементы с таким тегом.

`#id` – элемент с данным id.

`.class` – элементы с таким классом.

`[name="value"]` – селекторы на атрибут (см. далее).

## \*\* Отношения

Четыре вида отношений между элементами.

$\text{div } p$  – элементы  $p$ , являющиеся потомками  $\text{div}$ .

$\text{div } > p$  – только непосредственные потомки

Есть и два более редких:

$\text{div } \sim p$  – правые соседи: все  $p$  на том же уровне вложенности, которые идут после  $\text{div}$ .

$\text{div } + p$  – первый правый сосед:  $p$  на том же уровне вложенности, который идёт сразу после  $\text{div}$  (если есть).

# Упражнения

Сервис для тестирования работы в “доме”  
<https://jsfiddle.net/>

- Практические задания на DOM

[w3schools.com/js/js\\_html\\_dom\\_elements.asp](https://www.w3schools.com/js/js_html_dom_elements.asp)

- **Выполняем задание 1,2,3 и 5.**

# Установка значений

- `element.innerHTML` = (новое содержимое)  
Изменение содержимого элемента

```
element.innerHTML = 'Здравствуй мир!'
```

- `element.attribute` = (Новое значение)

Установка [attribute] равным (значению)

```
t.src='mountain.jpg'
```

- `element.style.property` = Изменение стиля

```
document.getElementById('m').style.color='red'
```

# Упражнения-2

- Проходим по ссылке:

[w3schools.com/js/js\\_html\\_dom\\_html.asp](https://www.w3schools.com/js/js_html_dom_html.asp)

- Выполняем задание 1,2,3,4 и 5.

- Проходим по ссылке:

[w3schools.com/js/js\\_html\\_dom\\_css.asp](https://www.w3schools.com/js/js_html_dom_css.asp)

- Выполняем задание 1,2,3,4 и 5.

# Вставка в DOM при помощи JS

// получим узел

```
var parent = document.getElementById("feed");
```

// создадим пустой узел

```
var child = document.createElement("div");
```

// добавим ему класс

```
child.setAttribute("class", "feed-container");//Аналог child.class=
```

// добавим в родительский узел созданного ребенка

```
parent.appendChild(child) // либо так
```

```
parent.insertBefore(child)
```

# Задание: Сгенерировать «зеленый» HTML

```
<div id='feed'>
```

```
  <div class='feed-container'>
```

```
    Текст
```

```
  </div>
```

```
</div>
```

# Либо через innerHTML

// получим узел

```
var parent = document.getElementById("feed");
```

// создадим еще один узел в родительском

```
parent.innerHTML = "<div>АБВГдейка</div>"
```

//либо это можно проделать так

```
var child = document.createElement("div");
```

```
var text =
```

```
document.createTextNode('АБВГдейка');
```

```
child.appendChild(text);
```

```
parent.appendChild(child);
```

# Что ещё прочитать?

- <https://learn.javascript.ru/traversing-dom>

Навигация по DOM

- <https://learn.javascript.ru/css-selectors>

\* Селекторы CSS

# Задание

- Сделать *select-option*, заполненный при помощи JS, в котором можно выбирать несколько вариантов

<http://htmlbook.ru/html/select> - изучить!

- Сделать страницу светофор, которая мелькает желтым цветом или постепенно зажигается и затухает

<http://htmlbook.ru/css/opacity>

<http://javascript.ru/setinterval>

- \* Реальный светофор, для которого можно задать время зеленого и красного света.