

# JS: DOM

# НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

# Повторим;)

Сколько способов создать функцию вы узнали?

Какие функции называются анонимными?

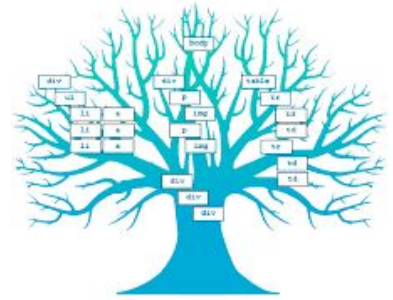
# ЦЕЛЬ

Изучить различные методы работы с DOM

# ПЛАН ЗАНЯТИЯ

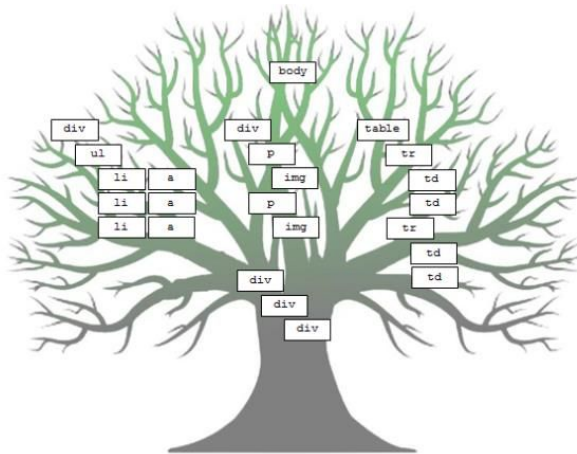
- Определение DOM
- Методы DOM

# DOM (Document Object Model)

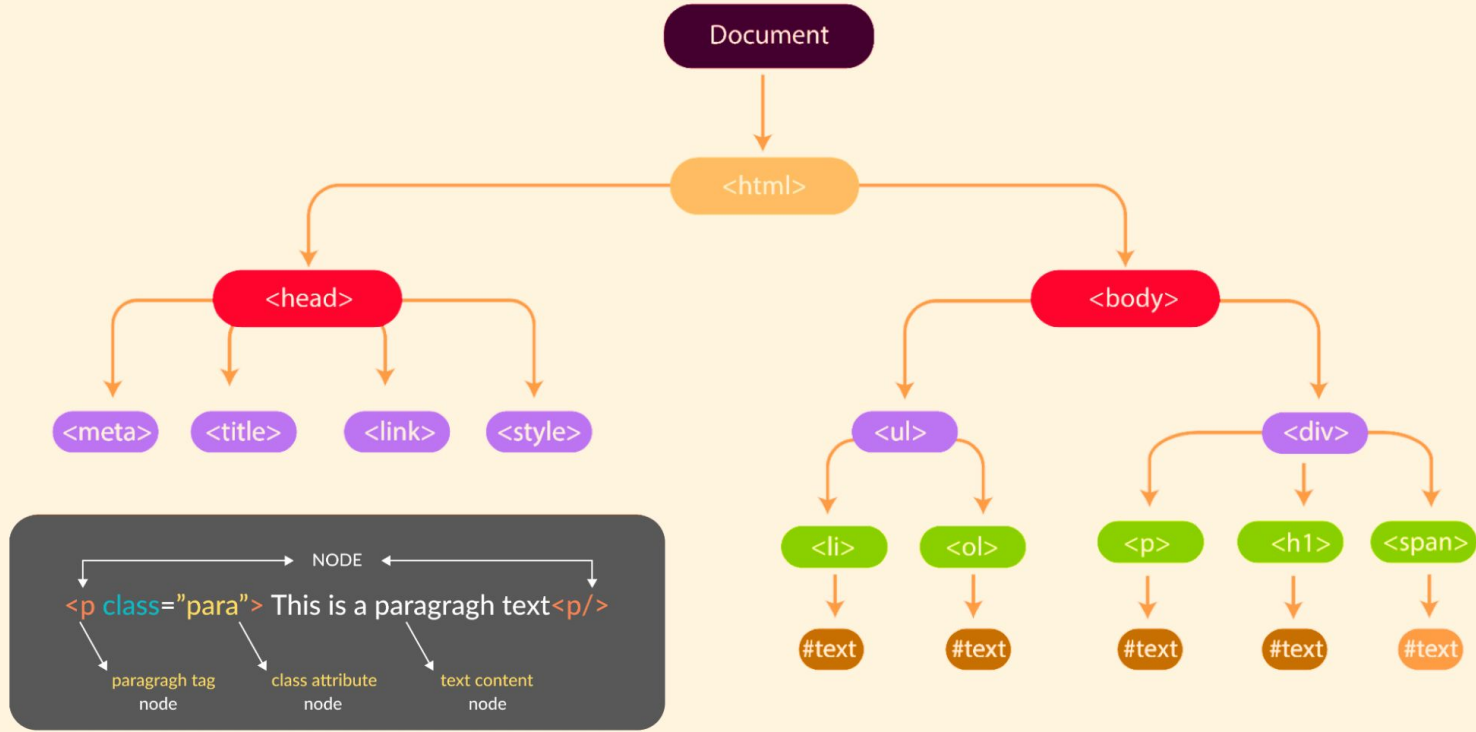


**DOM (Document Object Model)** – это объектная модель документа, которую браузер создает в памяти компьютера на основании HTML-кода.

Иными словами, это представление HTML-документа в виде дерева тегов. Такое дерево нужно для правильного отображения сайта и внесения изменений на страницах с помощью JavaScript



# The DOM Structure/ DOM TREE





**Все что имеется в разметке отображается в DOM дереве и выступает узлом этого дерева**

## Document node

Это вся страница браузера. Все остальное вложено в этот узел – как дети.

## Element nodes

Все элементы, такие как заголовки (`<h1>` to `<h6>`) или параграфы (`<p>`) представлены отдельными узлами в дереве. Мы можем получить даже доступ к их атрибутам и текстовому содержанию

# Все что имеется в разметке отображается в DOM дереве и выступает узлом этого дерева

## Attribute nodes

Если тег элемента содержит атрибуты – эти атрибуты будут представлены в качестве отдельных узлов атрибутов. По сути, это уже не дочерние элементы по отношению к данному элементу, а свойства данного элемента.

## Text nodes

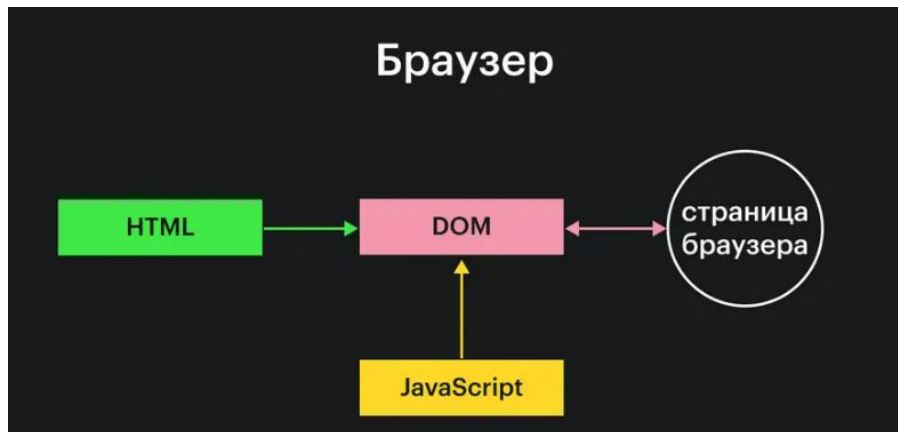
Текст тоже создает отдельный текстовый узел

```
+-----+
|           DOM Node Object           |
+-----+
|           <div> Tag                 |
+-----+
| Properties:                        |
| - tagName: "DIV"                   |
| - id: ""                           |
| - className: ""                    |
| - children: []                     |
| - parentNode: null                 |
| - innerHTML: ""                    |
| - textContent: ""                  |
|                                     |
| Methods:                           |
| - appendChild(child)               |
| - removeChild(child)               |
| - setAttribute(name, value)        |
| - getAttribute(name)               |
| - addEventListener(event, func)    |
| - removeEventListener(event)        |
+-----+
```

## DOM позволяет управлять HTML-разметкой из JavaScript-кода.

Управление обычно состоит из:

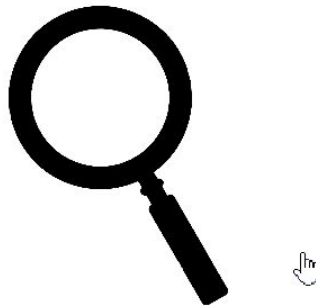
- добавления элементов
- удаления элементов
- изменения стилей и содержимого элементов



**Прежде чем управлять элементом его нужно  
выбрать!**



# Методы поиска элементов



# Метод 1

## Поиск элемента по ID

```
document.getElementById(id)
```

Примечание: возвращает элемент с заданным id. элемент должен иметь атрибут id

# Метод 1

## Поиск элемента по ID

Задача: получить элемент с id="elem"

```
let elem = document.getElementById('elem');
```



## Метод 2

### Поиск элемента по тегу

```
document.getElementsByTagName (tag)
```

Примечание: метод ищет элементы с данным тегом и возвращает их коллекцию.

## Метод 2

# Поиск элемента по тегу

Задача: получить все элементы div в документе

```
let divs = document.getElementsByTagName('div');
```

## Метод 3

### Поиск элементов по названию класса

```
document.getElementsByClassName(className)
```

Примечание: метод возвращает элементы, которые имеют данный класс

# Метод 3

## Поиск элементов по названию класса

Задача: получить все элементы с классом class='article'

```
let articles = document.getElementsByClassName('article');
```

## Метод 4

# Поиск элементов по значению атрибута name

```
document.getElementsByTagName(name)
```

Примечание: возвращает элементы с заданным атрибутом name

# Метод 4

## Поиск элементов по значению атрибута name

Задача: получить все элементы со значением атрибута name="up"

```
let articles = document.getElementsByTagName("up");
```

# Метод 5

Поиск по CSS-селектору  
(самый универсальный метод поиска)

```
document.querySelectorAll(css selector)
```

```
document.querySelector(css selector)
```

# Метод 5

Поиск по CSS-селектору

Задача: получает все элементы `<li>`, которые являются дочерними для `<ul>`

```
let elements = document.querySelectorAll('ul > li');
```



# Добавление элементов



# 1 шаг - Создать элемент

Метод: `document.createElement("tag")`

Создание: `let el = document.createElement("div")`

## 2 шаг - Заполнить элемент

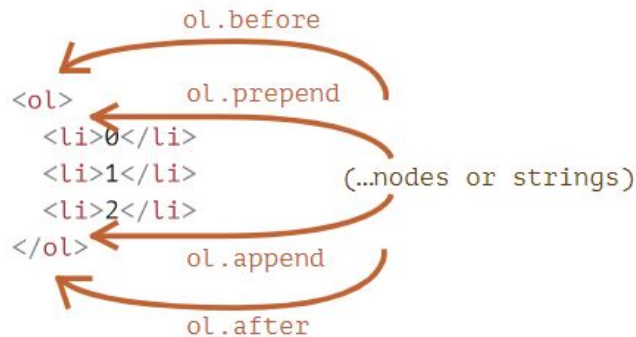
Метод: `node.textContent = "text"`

Заполнение: `el.textContent = "Text"`

## 3 шаг - Добавляем элемент в document

### Методы:

- `node.append(el)` – добавляет узлы или строки в конец `node`,
- `node.prepend(el)` – вставляет узлы или строки в начало `node`,
- `node.before(el)` – вставляет узлы или строки до `node`,
- `node.after(el)` – вставляет узлы или строки после `node`,



# Изменение элементов



- **textContent**

Позволяет задавать или получать текстовое содержимое элемента и его потомков.

```
let text = element.textContent  
  
element.textContent = "Just text"
```

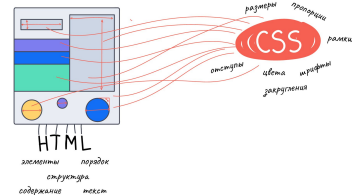
- innerHTML

Свойство innerHTML позволяет считать содержимое элемента в виде HTML-строки или установить новый HTML.

```
<form>
  <label>Логин</label>
  <input type="text" id="login" />
  <div class="error">Введите логин</div>
</form>
```

```
1 const form = document.querySelector('form')
2
3 console.log(form.innerHTML)
4 // '<label>Логин</label><input type="text" id="login" /><div class="error">Вве
5
6 // Меняем содержимое новым html
7 form.innerHTML = '<div class="success">Вход выполнен</div>'
```

- **изменение стилей**



HTML DOM позволяет JavaScript изменять стиль HTML элементов.

Синтаксис:

```
document.getElementById(id).style.свойство = новый стиль
```

Применение:

```
document.getElementById("p2").style.color = "blue";
```



- **удаление элементов**

Для удаления узла есть метод `node.remove()`

Пример использования: удаление элемента с `id="register"`

```
document.getElementById("register").remove();
```





# **Ваша новая IT-профессия – Ваш новый уровень жизни**

Программирование с нуля в  
немецкой школе AIT TR GmbH