

Frontend Basic

LESSON 11



Anna Khachaturyan



Анна Хачатурян

Front End/Gen Tech Teacher

- Since 2018 in IT
- Full Stack Developer at Web Magnat
- QA Engineer/Web Developer at Central Bank of RA
- Lecturer at Plekhanov Russian University of Economics
- TA at Picsart Academy
- Teacher at Tel-Ran

https://t.me/anny_khachaturyan



ВАЖНО:

- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

ПЛАН ЗАНЯТИЯ

1. Повторение изученного
2. Вопросы по повторению
3. DOM – работа с классами
4. DOM - создание элементов
5. Практика

1

ПОВТОРЕНИЕ

2

ВОПРОСЫ ПО ПОВТОРЕНИЮ

3

ДОМ – РАБОТА С КЛАССАМИ

Alert

Функция `alert()` показывает сообщение и ждёт, пока пользователь нажмёт кнопку «OK».

```
alert('Hello');
```

Это небольшое окно с сообщением называется *модальным окном*. Понятие *модальное* означает, что пользователь не может взаимодействовать с интерфейсом остальной части страницы, нажимать на другие кнопки и т.д. до тех пор, пока взаимодействует с окном. В данном случае – пока не будет нажата кнопка «OK».

Изменения, создание и удаление класса

`className` и `classList`

Изменение класса является одним из наиболее часто используемых действий в скриптах.

Когда-то давно в JavaScript существовало ограничение: зарезервированное слово типа "class" не могло быть свойством объекта. Это ограничение сейчас отсутствует, но в то время было невозможно иметь свойство `elem.class`. Поэтому для классов было введено схожее свойство "**className**": `elem.className` соответствует атрибуту "class".

Например:

HTML

```
<p class="p1 text">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quis illum at odit accusantium obcaecati magnam veritatis velit magni, perferendis, aperiam consectetur, labore qui? Accusamus, modi natus. Eligendi asperiores odit a.</p>
```

JS

```
let paragraph = document.querySelector('p');
let classes = paragraph.className;
console.log(classes); // p1 text

paragraph.className = 'paragraph1'; //изменили значение
```


Если мы присваиваем что-то `elem.className`, то это заменяет всю строку с классами. Иногда это то, что нам нужно, но часто мы хотим добавить/удалить один класс.

Для этого есть другое свойство: `elem.classList`.

classList – это специальный объект с методами для добавления/удаления одного класса.

Кроме того, **classList** является **перебираемым**, поэтому можно перечислить все классы при помощи **for...of**.

Например:

```
let classes = paragraph.classList;
console.log(classes); //список

for(let elem of classes){
    console.log(elem); // вывод всех классов элемента
}
```

Так что мы можем работать как со строкой полного класса, используя `className`, так и с отдельными классами, используя `classList`. Выбираем тот вариант, который нам удобнее.

Методы `classList`:

- `elem.classList.add/remove("class")` – добавить/удалить класс.
- `elem.classList.toggle("class")` – добавить класс, если его нет, иначе удалить.
- `elem.classList.contains("class")` – проверка наличия класса, возвращает `true/false`.

```
paragraph.classList.add('paragraph1');
paragraph.classList.remove('p1');
paragraph.classList.toggle('p1');
console.log(paragraph.classList.contains('p2'));
```

4

DOM – СОЗДАНИЕ ЭЛЕМЕНТОВ

Создание элемента

`document.createElement(tag)`

Создаёт новый элемент с заданным тегом:

```
let div = document.createElement('div');
```

Создание сообщения

В нашем случае сообщение – это div с классом block1 и текст в нём:

```
let div = document.createElement('div');  
div.classList.add('block1');  
div.innerText = 'Hello World!';
```

Мы создали элемент, но пока он только в переменной. Мы не можем видеть его на странице, поскольку он не является частью документа.

Методы вставки

Чтобы наш div появился, нам нужно вставить его где-нибудь в document. Например, в document.body.

Для этого есть метод **append**, в нашем случае: `document.body.append(div)`.

Вот пример:

```
let div = document.createElement('div');  
let div = document.createElement('div');  
div.classList.add('block1');  
div.innerText = 'Hello World!';  
document.body.append(div);
```

Вот методы для различных вариантов вставки:

- **node.append(...nodes or strings)** – добавляет узлы или строки в конец node,
- **node.prepend(...nodes or strings)** – вставляет узлы или строки в начало node,
- **node.before(...nodes or strings)** – вставляет узлы или строки до node,
- **node.after(...nodes or strings)** – вставляет узлы или строки после node,

Вот пример использования этих методов, чтобы добавить новые элементы в список и текст до/после него:

HTML

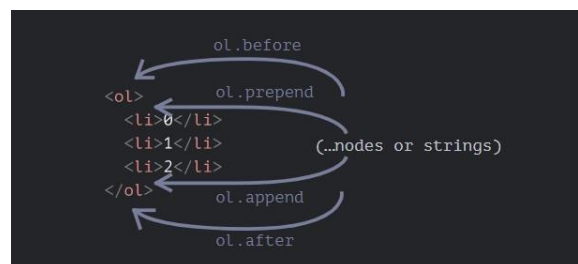
```
<ol>
  <li>First</li>
  <li>Middle</li>
  <li>Last</li>
</ol>
```

JS

```
let div = document.createElement('div');
div.classList.add('block1');
div.innerText = 'Hello World!';

let list = document.querySelector('ol');
list.before(div); // вставить блок перед <ol>
list.after(div);  // вставить блок после <ol>
list.append(div); // вставить блок в конец <ol>
list.prepend(div); // вставить блок в начало <ol>
```

Наглядная иллюстрация того, куда эти методы вставляют:



Удаление узлов

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

Например, сделаем так, чтобы наш первый блок удалился:

HTML

```
<div class="block1">
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Corrupti fugiat voluptatem sint unde voluptatum
quaerat odio repellendus, nesciunt optio, suscipit assumenda hic accusamus beatae, illum recusandae amet ipsum. Quos,
eaque.</p>
</div>
<div class="block2">
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Corrupti fugiat voluptatem sint unde voluptatum
quaerat odio repellendus, nesciunt optio, suscipit assumenda hic accusamus beatae, illum recusandae amet ipsum. Quos,
eaque.</p>
</div>
```

JS

```
let div = document.querySelector('.block1');
div.remove();
// либо
document.querySelector('.block1').remove();
```

6

ПРАКТИКА

Задание: Создайте функцию clear(elem), которая удаляет все элементы с заданным селектором.

HTML

```
<ol>
  <li>First</li>
  <li>Middle</li>
  <li>Last</li>
</ol>

<p>texttexttext</p>
<p class="p1">texttexttext</p>
<p>texttexttext</p>
<p class="p1">texttexttext</p>
<p>texttexttext</p>
```

JS

```
function clear(elem){ // elem - параметр
  let elements = document.querySelectorAll(elem); //список всех элементов с заданным селектором
  for(let list of elements){
    list.remove(); // удаление элементов
  }
}

clear('li'); // вызов функции для тега <li>
clear('p'); // вызов функции для тега <p>
clear('.p1'); // вызов функции для тега с классом 'p1'
```

Задание: Напишите скрипт для создания списка.

Для каждого пункта:

1. Запрашивайте содержимое пункта (тега) у пользователя с помощью prompt 10 раз.
2. Создавайте элементы и добавляйте его к .

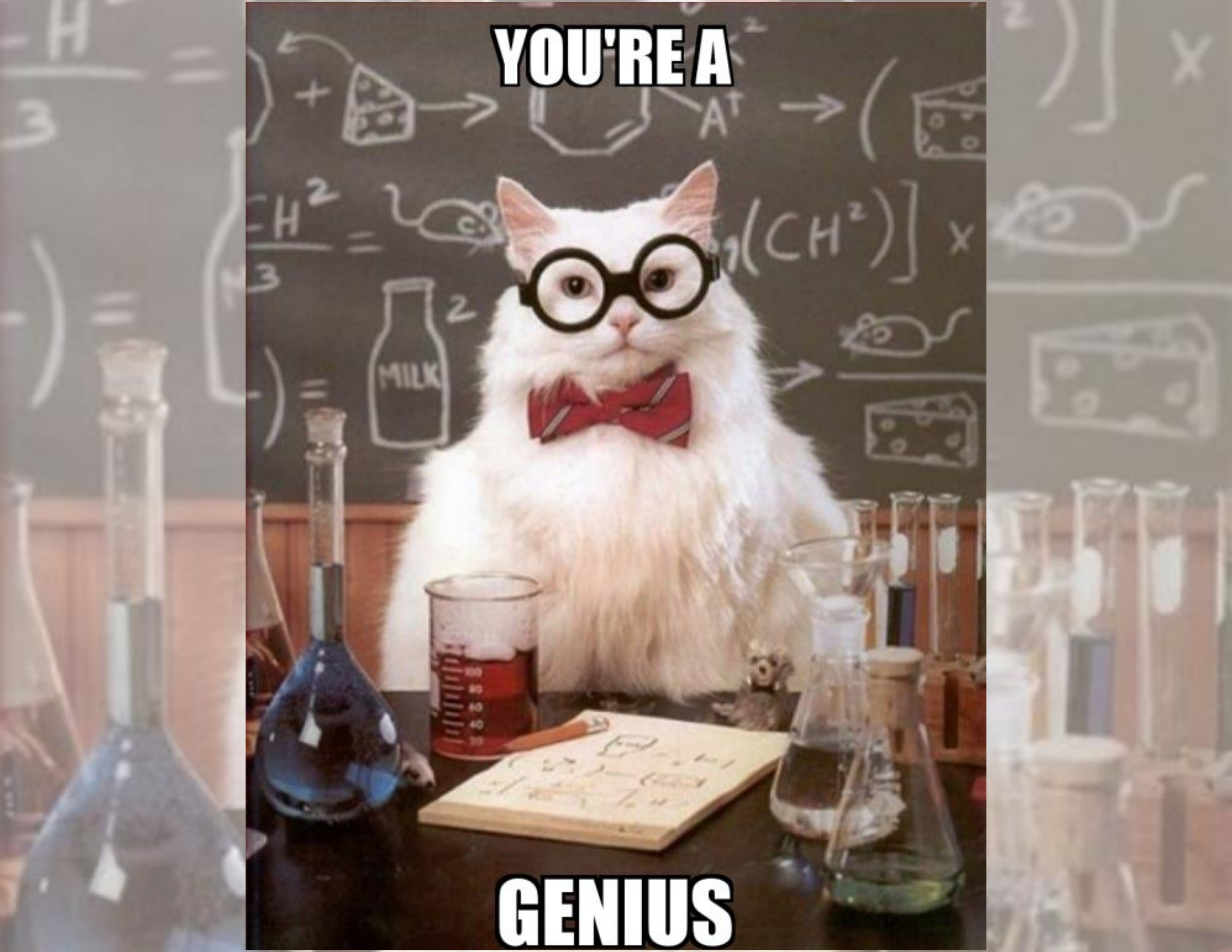
Решение:

```
let ul = document.createElement('ul'); // создание тела списка
document.body.append(ul);

for(let i=0; i<10; i++){
    let data = prompt('Введите текст для элемента списка'); // ввод содержимого элемента списка

    let li = document.createElement('li'); // создание элемента списка
    li.innerText = data;
    ul.append(li);
}
```


YOU'RE A



GENIUS