**ClassList и делигирование событий**

Свойство **classList** возвращает псевдомассив [DOMTokenList](https://developer.mozilla.org/ru/docs/Web/API/DOMTokenList), содержащий все классы элемента.

var elementClasses = elem.classList;

<https://developer.mozilla.org/ru/docs/Web/API/Element/classList>

**Методы**

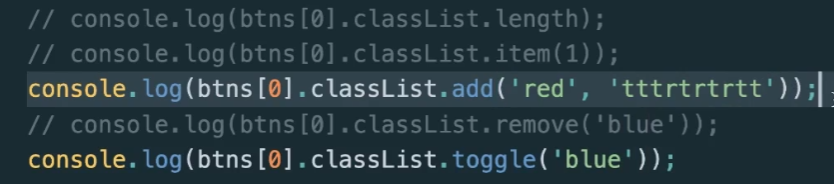
**\*** можно передавать несколько классов в аттрибуты

**Item()** – получает опредленный класс

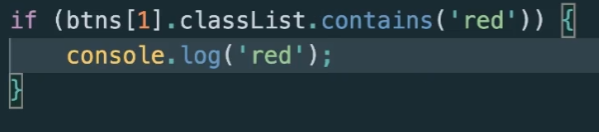
**Add()** – добавление класса

**Remove()** – удаление класса

**Toggle() –** если есть класс – то будет убран, если нету то будет добавлен



**Contains()**



**ClassName** Deprecated



Возвращает строку классав

**Делигирование событий**

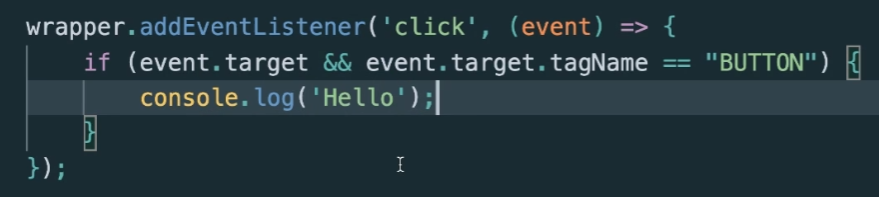
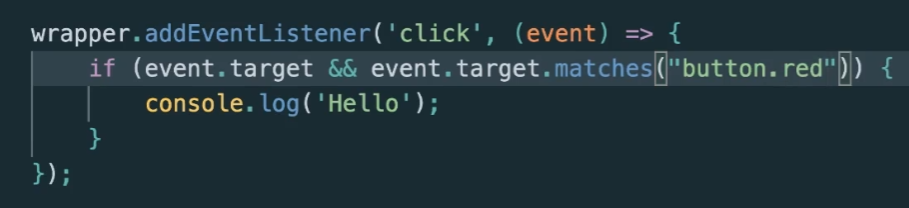
1. <https://learn.javascript.ru/event-delegation>
2. <https://medium.com/@stasonmars/%D0%B4%D0%B5%D0%BB%D0%B5%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D1%81%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B8%CC%86-%D0%B2-javascript-d91cbdd8916a>

По сути просто вешаем **eventListener()** на родителя, чтобы не вешать его на каждый элемент

Идея в том, что если у нас есть много элементов, события на которых нужно обрабатывать похожим образом, то вместо того, чтобы назначать обработчик каждому, мы ставим один обработчик на их общего предка.

Из него можно получить целевой элемент event.target, понять на каком именно потомке произошло событие и обработать его.

Классическое применение делигирование событий



Или

