

Модуль - JavaScript

6/15 занятие

События

Событие это реакция браузера на взаимодействие пользователя с DOM-элементом, либо на заданные условия по времени или состоянию DOM. Например, когда DOM полностью готов, происходит событие

DOMContentLoaded

Самые распространенные события:

click – клик мышью(кнопка нажата и отпущена) или тап по DOM элементу

contextmenu – клик правой кнопкой мыши по элемент

mouseover / mouseout – курсор пересекает границу элемента

mousedown / mouseup – только нажатие кнопки мышь или отпускание

mousemove – движения курсора в пределах границ элемента, раз в N миллисекунд

focus – когда устанавливается фокус для ввода текста в `<input>`.

keydown и **keyup** - нажата или отпущена клавиша

Обработка событий

Аттрибут onclick = ФункцияОбработчик

`<button id='btn' onclick='clickHandler(e)'`

Чаще используется отслеживание событий:

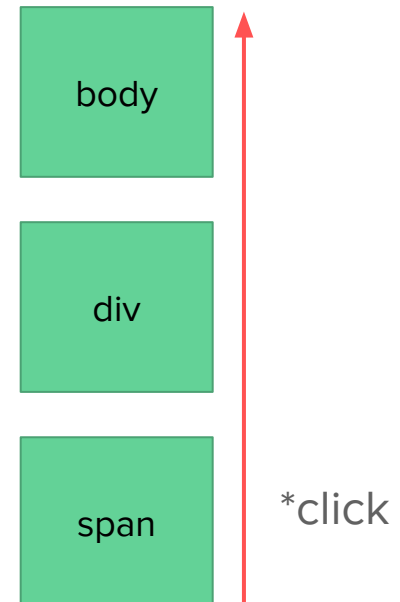
```
const handlerFunction = (e) => {  
    //исполняемый код  
}  
  
btn.addEventListener("click", handlerFunction, true)  
  
// событие[String] , Функция[function] , перехват[bool]
```

< JS >

ВСПЛЫТИЕ

Всплытие - это процесс передачи триггера события по узлам DOM дерева от самого последнего потомка, на котором инициировали событие и до корня дерева

```
span.addEventListener('click', ()=>{ alert('клик - span')} ) < JS >  
div.addEventListener('click', ()=>{ alert('клик - div')} )  
body.addEventListener('click', ()=>{ alert('клик - body')} )  
  
// 1 клик - span  
// 2 клик - div  
// 3 клик - body
```



Объект event

При возникновении любого события формируется объект **event**, который содержит всю возможную информацию о событии

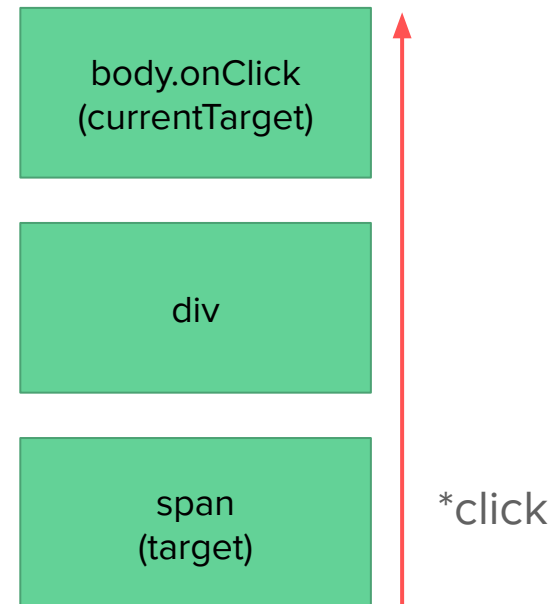
Например:

event.target – тот элемент с которым взаимодействовали (кликнули, зацепили границу, ...).

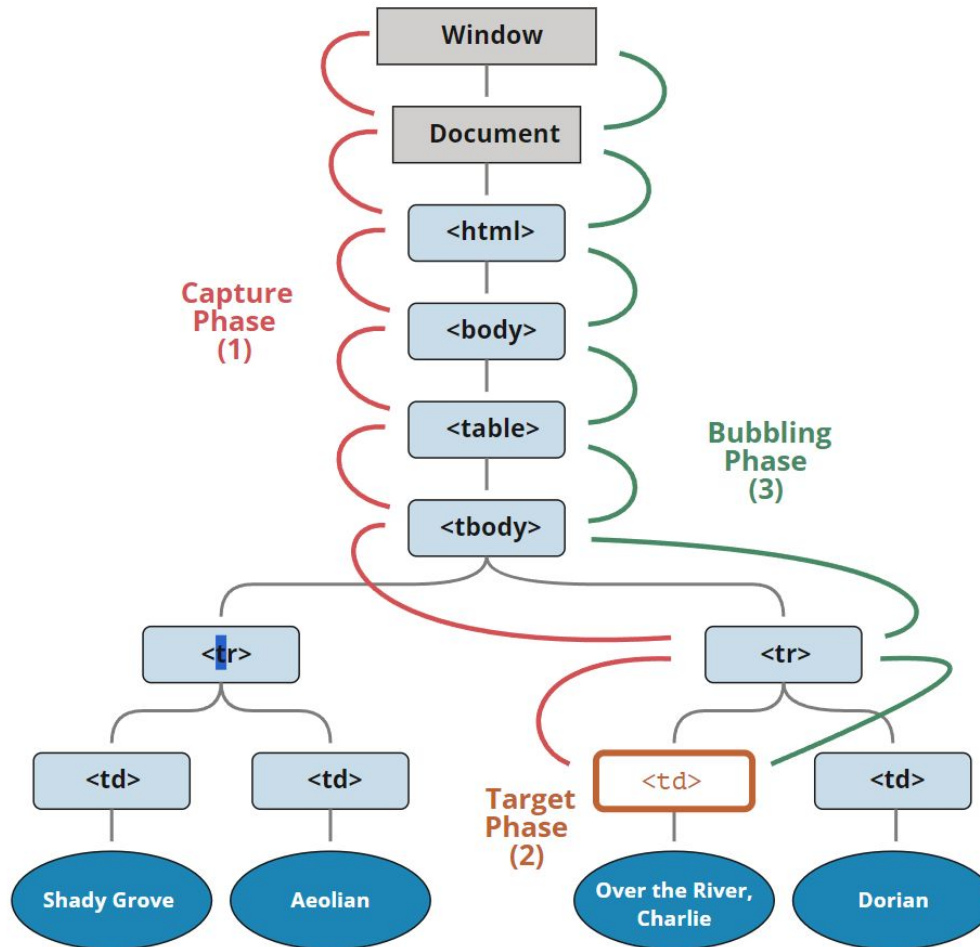
event.currentTarget – элемент, на котором отслеживается событие

event.stopPropagation() – Прекращение всплытия

event.preventDefault() – Блокирует стандартное действие, например переход по ссылке тега <a>



Погружение



Практика: Создание карусели

<https://getbootstrap.com/docs/5.3/components/carousel/>

перерыв

Методы взаимодействия с DOM

// Создание элементов HTML

< JS >

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

// Изменение текста и HTML

```
div.innerHTML = '<strong>Всем привет!</strong> Вы прочитали важное сообщение.'
```

```
div.innerText = '<strong>Всем привет!</strong> Вы прочитали важное сообщение.'
```

Методы вставки элемента в DOM

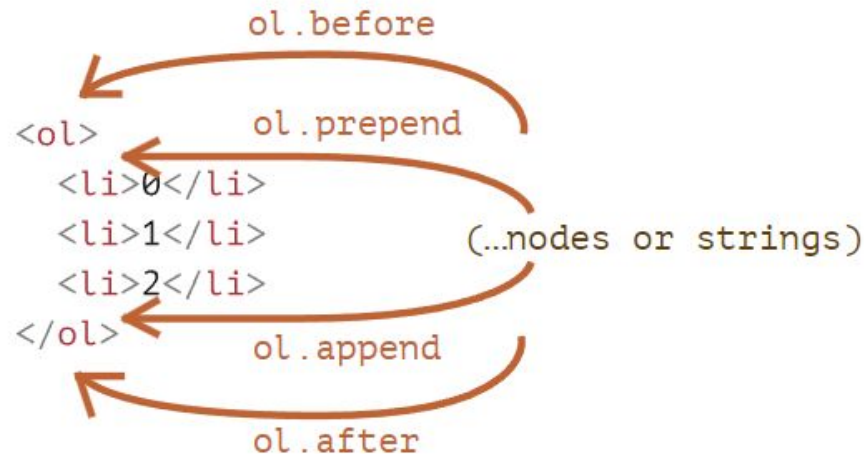
`node.append(...nodes or strings)` – добавляет узлы или строки в конец `node`

`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`,

`node.remove()` - Удаление элемента



Шаблоны HTML в js строках

Добавление нескольких элементов HTML одновременно

```
// Добавление нескольких элементов HTML одновременно
```

< JS >

```
let counter = 11
```

```
let template = `
```

```
  <div>
```

```
    <b>Файлов загружено: ${ counter > 10 ? 'больше 10' : counter } </b>
```

```
  </div>
```

```
`
```

```
document.body.innerHTML = template
```

Практика: Создание плагина карусели

<https://getbootstrap.com/docs/5.3/components/carousel/>

Конец занятия