

Основы JavaScript

События. Обработка событий

Определение

Событие – это сигнал от браузера о том, что что-то произошло. Существует множество различных видов событий.

Некоторые виды событий:

- события мыши
- события клавиатуры
- события на элементах управления
- события документов
- события CSS

Обработчики событий

— — —

Событию можно назначить **обработчик**, то есть функцию, которая сработает, как только событие произошло.

Способы назначения обработчика:

- используя HTML-атрибут
- используя свойства DOM-объекта
- используя специальные методы

Добавление обработчика. HTML-атрибут

— — —

Обработчик может быть назначен прямо в атрибуте, который называется `on<событие>`.

Например, чтобы обрабатывать событие клика на элемент, к нему необходимо, присвоить обработчик `onclick`.

Примеры:

```
<button onclick="alert('The button is clicked!')">Click Me!</button>  
<button onclick="doStuff()">Do Stuff</button>
```

Добавление обработчика. Свойства DOM-объекта

— — —

Можно назначить обработчик, используя свойство DOM-элемента `on<событие>`.

Пример:

```
<button id="btn-purchase">Purchase</button>
<script>
  var btn = document.getElementById("btn-purchase");

  btn.onclick = function () {
    alert('Thank you!');
  };
</script>
```

Недостатки

— — —

Фундаментальный недостаток описанных способов назначения обработчика – невозможность **одновременного** добавления **нескольких обработчиков** на одно событие.



Добавление обработчика. Методы обработки событий

Методы **addEventListener** и **removeEventListener** являются современным способом назначить или удалить обработчик, и при этом позволяют использовать **сколько угодно любых обработчиков**.

Назначение обработчика осуществляется вызовом **addEventListener** с тремя аргументами:

```
element.addEventListener(event, handler[, options]);  
element.removeEventListener(event, handler[, options]);
```

Методы `addEventListener` и `removeEventListener`

— — —

event

Имя события, например `click`.

handler

Ссылка на функцию, которую надо поставить обработчиком.

options

Необязательный аргумент объекта опций. Этот аргумент применяется редко. В этом курсе он не подлежит рассмотрению.

Demo

Методы добавления и удаления
обработчиков событий



Доступ к элементу в обработчике

— — —

Внутри обработчика события `this` ссылается на текущий элемент, то есть на тот, на котором он сработал.

Это можно использовать, чтобы получить свойства или изменить элемент.

В коде ниже `button` выводит свое содержимое, используя `this.innerHTML`:

```
<button onclick="alert(this.innerHTML)">Click Me!</button>
```

Методы добавления обработчика событий в IE8-

— — —

В IE8- вместо `addEventListener/removeEventListener` используются свои методы.

Назначение обработчика осуществляется вызовом **`attachEvent`**:

```
element.attachEvent("on" + event, handler);
```

Удаление обработчика – вызовом **`detachEvent`**:

```
element.detachEvent("on" + event, handler);
```

Распространенные события. События мыши

— — —

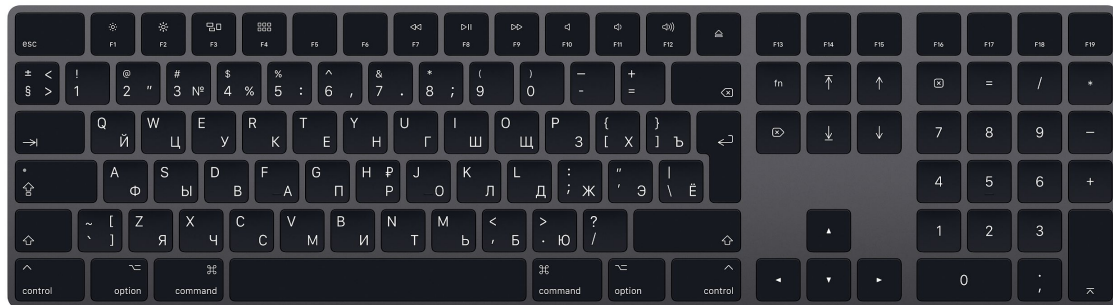
click	Клик на элемент левой кнопкой мыши
contextmenu	Клик на элемент правой кнопкой мыши
mouseover	На элемент наводится мышь
mousedown и mouseup	Кнопку мыши нажали или отжали
mousemove	Движении мыши



Распространенные события. События клавиатуры

— — —

keydown	Пользователь нажал клавишу
keyup	Пользователь отпустил клавишу



Распространенные события. Другие события

— — —

submit	Происходит, когда пользователь отправляет форму запроса
focus	Пользователь выделяет элемент управления
DOMContentLoaded	Документ полностью загружен. DOM-дерево построено
transitionend	CSS-анимация завершена