



POWERCODE
ACADEMY



Занятие на тему:

События JS

Горбачевский Валерий

Введение в события



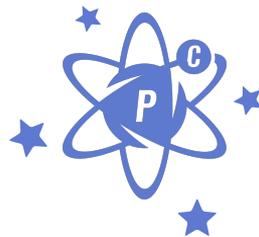
Практически все JavaScript-приложения выполняют те или иные действия, откликаясь на различные события.

Событие - это сигнал от браузера о том, что что-то произошло.

Есть множество самых различных событий.

- DOM-события, которые инициируются элементами DOM. Например, событие `click` происходит при клике на элементе, а событие `mouseover` - когда указатель мыши появляется над элементом,
- События окна. Например событие `resize` - при изменении размера окна браузера,
- Другие события, например `load`, `readystatechange`. Они используются, скажем, в технологии AJAX.

Основные события



События мыши:

- click – происходит, когда кликнули на элемент левой кнопкой мыши
- contextmenu – происходит, когда кликнули на элемент правой кнопкой мыши
- mouseover – возникает, когда на элемент наводится мышь
- mousedown и mouseup – когда кнопку мыши нажали или отжали
- mousemove – при движении мыши

События на элементах управления:

- submit – посетитель отправил форму `<form>`
- focus – посетитель фокусируется на элементе, например нажимает на `<input>`

Клавиатурные события:

- keydown – когда посетитель нажимает клавишу
- keyup – когда посетитель отпускает клавишу

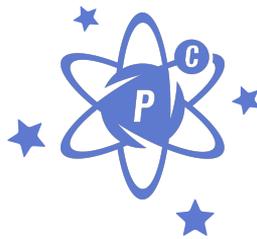
События документа:

- DOMContentLoaded – когда HTML загружен и обработан, DOM документа полностью построен и доступен.

События CSS:

- transitionend – когда CSS-анимация завершена.

Назначение обработчиков



Через свойство объекта:

```
<input id="myElement" type="button" value="Нажми меня"/>

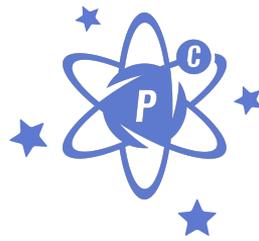
<script>
document.getElementById('myElement').onclick = function() {
    alert('Спасибо')
}
</script>
```

Можно назначить и функцию:

```
function doSomething() {
    alert('Спасибо')
}

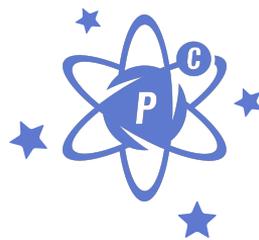
document.getElementById('button').onclick = doSomething
// Обратите внимание - свойству присваивается именно
// сама функция-обработчик doSomething, а не doSomething():
```

Доступ к элементу через this



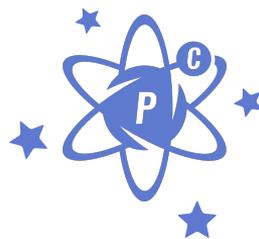
```
<button onclick="alert(this.innerHTML)">Нажми меня</button>
```

Когда использовать



```
input.onclick = function() { alert(1) }  
// ...  
input.onclick = function() { alert(2) } // заменит предыдущий
```

addEventListener и removeEventListener



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

event

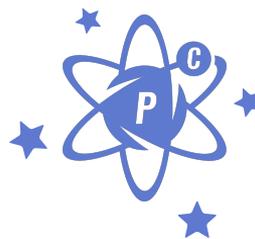
Имя события, например click

handler

Ссылка на функцию, которую надо поставить обработчиком.

```
// передать те же аргументы, что были у addEventListener  
element.removeEventListener(event, handler[, phase]);
```

Несколько обработчиков на одно событие одного элемента



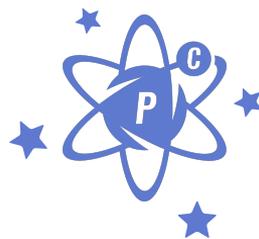
```
<input id="elem" type="button" value="Нажми меня"/>

<script>
  function handler1() {
    alert('Спасибо!');
  };

  function handler2() {
    alert('Спасибо ещё раз!');
  }

  elem.onclick = function() { alert("Привет"); };
  elem.addEventListener("click", handler1); // Спасибо!
  elem.addEventListener("click", handler2); // Спасибо ещё раз!
</script>
```

Объект события - event

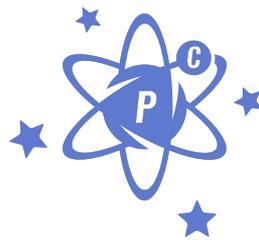


```
function doSomething(event) {  
    // event - будет содержать объект события  
}  
  
element.onclick = doSomething;
```

как предотвратить действие браузера по умолчанию

```
function showModal(event) {  
    // event - будет содержать объект события  
    event.preventDefault(); //предотвращение действия по умолчанию  
    // Код показа модального окна  
}  
  
var modalTrigger = document.querySelector("a[data-modal]");  
  
modalTrigger.onclick = showModal;
```

Основные события



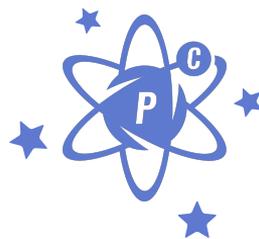
Простые события:

- `mousedown` - Кнопка мыши нажата над элементом.
- `mouseup` - Кнопка мыши отпущена над элементом.
- `mouseover` - Мышь появилась над элементом.
- `mouseout` - Мышь ушла с элемента.
- `mousemove` - Каждое движение мыши над элементом генерирует это событие.

Комплексные события:

- `click` - Вызывается при клике мышью, то есть при `mousedown`, а затем `mouseup` на одном элементе.
- `contextmenu` - Вызывается при клике правой кнопкой мыши на элементе.
- `dblclick` - Вызывается при двойном клике по элементу..

Координаты курсора



в окне: clientX/Y

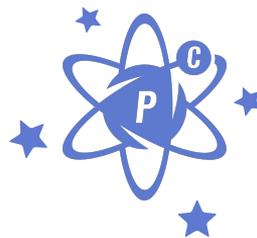
```
<input onmousemove="this.value = event.clientX+':'+event.clientY">
```

Относительно документа: pageX/Y

```
<input onmousemove="this.value = event.pageX+':'+event.pageY">
```

События мыши: движение

mouseover/mouseout, mouseenter/mouseleave



Событие `mouseover` происходит, когда мышь появляется над элементом, а `mouseout` – когда уходит из него.

При переходе на потомка срабатывает `mouseout` на родителе.

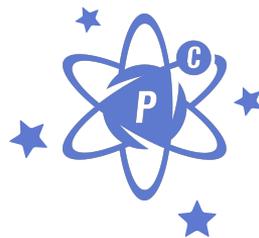
`mouseover/mouseout`

<https://codepen.io/alexcss/pen/iApWKo>

`mouseenter/mouseleave`

<https://codepen.io/alexcss/pen/ZOjQrG>

Мышь: колёсико, событие wheel



wheel

Свойство `deltaY` – количество прокрученных пикселей по горизонтали и вертикали. Существуют также свойства `deltaX` и `deltaZ` для других направлений прокрутки.

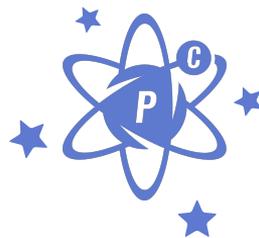
```
document.getElementById("myDIV").addEventListener("wheel", myFunction);

function myFunction(event) {
    this.style.fontSize = event.deltaY + "px";
}
```

Прокрутка: событие scroll

```
window.onscroll = function() {
    var scrolled = window.pageYOffset;
    document.getElementById('showScroll').innerHTML = scrolled + 'px';
}
```

Клавиатура: `keyup`, `keydown`, `keypress`



`keydown/keyup`

происходят при нажатии/отпускании клавиши и позволяют получить её скан-код в свойстве `keyCode`.

```
<input onkeydown="this.nextElementSibling.innerHTML = event.keyCode"><b></b>
```

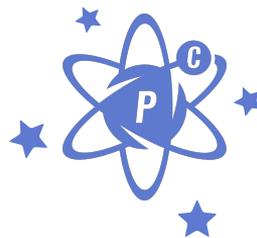
`keypress`

Возникает сразу после `keydown`, если нажата символьная клавиша, т.е. нажатие приводит к появлению символа.

```
<input onkeypress="this.nextElementSibling.innerHTML = event.charCode"><b></b>
```

<https://codepen.io/alexcss/pen/LkBNBJ>

Клавиатура: `keyup`, `keydown`, `keypress`



`keydown/keyup`

происходят при нажатии/отпускании клавиши и позволяют получить её скан-код в свойстве `keyCode`.

```
<input onkeydown="this.nextElementSibling.innerHTML = event.keyCode"><b></b>
```

`keypress`

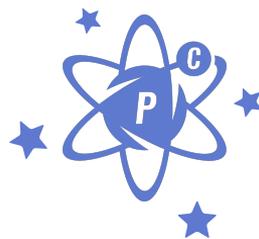
Возникает сразу после `keydown`, если нажата символьная клавиша, т.е. нажатие приводит к появлению символа.

```
<input onkeypress="this.nextElementSibling.innerHTML = event.charCode"><b></b>
```

<https://codepen.io/alexcss/pen/LkBNBJ>

Загрузка документа:

DOMContentLoaded, load, beforeunload, unload



- DOMContentLoaded – браузер полностью загрузил HTML, и построил DOM-дерево.
- load – браузер загрузил все ресурсы.
- beforeunload/unload – уход со страницы.

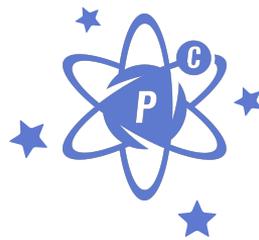
```
document.addEventListener("DOMContentLoaded", ready);
```

```
window.onload = function() {  
  //.....  
  $slider.init();  
};
```

```
window.onbeforeunload = function() {  
  return "Данные не сохранены. Точно перейти?";  
};
```

События формы и ее элементов:

навигация по формам



document.forms

- Форму можно получить как document.forms[name/index].

form.elements

- Элементы в форме: form.elements[name/index]. Каждый элемент имеет ссылку на форму в свойстве form. Свойство elements также есть у <fieldset>.
- Значение элементов читается/ставится через value или checked.

```
<form name="form">
  <select name="genre" multiple>
    <option value="blues" selected>Мягкий блюз</option>
    <option value="rock" selected>Жёсткий рок</option>
    <option value="classic">Классика</option>
  </select>
</form>
```

```
<script>
var form = document.forms[0];
var select = form.elements.genre;

for (var i = 0; i < select.options.length; i++) {
  var option = select.options[i];
  if(option.selected) {
    alert( option.value );
  }
}
</script>
```

Фокусировка: focus/blur



Событие `focus` вызывается тогда, когда пользователь фокусируется на элементе, а `blur` – когда фокус исчезает, например посетитель кликает на другом месте экрана.

```
input.onblur = function() {  
  if (isNaN(this.value)) { // введено не число  
    // показать ошибку  
    this.className = "error";  
    error.innerHTML = 'Вы ввели не число. Исправьте, пожалуйста.'  
  }  
};
```

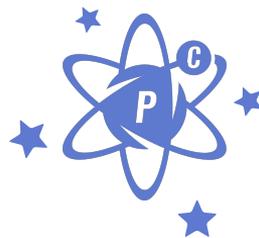
Метод `focus()`

```
age.onblur = function() {  
  if (isNaN(this.value)) { // введено не число  
    // показать ошибку  
    this.classList.add("error");  
    //... и вернуть фокус обратно  
    age.focus();  
  } else {  
    this.classList.remove("error");  
  }  
};
```

Изменение: change, input, cut, copy, paste

Событие	Описание	Особенности
change	Изменение значения любого элемента формы. Для текстовых элементов срабатывает при потере фокуса.	В IE8- на чекбоксах ждет потери фокуса, поэтому для мгновенной реакции ставят также onclick-обработчик или onpropertychange.
input	Событие срабатывает только на текстовых элементах. Оно не ждет потери фокуса, в отличие от change.	В IE8- не поддерживается, в IE9 не работает при удалении символов.
propertychange	Только для IE10-. Универсальное событие для отслеживания изменения свойств элементов. Имя изменённого свойства содержится в event.propertyName. Используют для мгновенной реакции на изменение значения в старых IE.	В IE9 не срабатывает при удалении символов.
cut/copy/paste	Срабатывают при вставке/копировании/удалении текста. Если в их обработчиках отменить действие браузера, то вставки/копирования/удаления не произойдёт.	Вставляемое значение получить нельзя: на момент срабатывания события в элементе всё ещё <i>старое</i> значение, а новое недоступно.

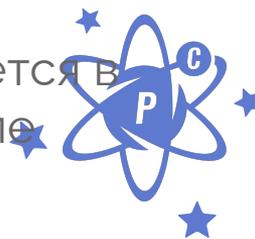
Изменение: change, input



```
<input type="text" onchange="alert(this.value)">  
<input type="button" value="Кнопка">
```

```
<input type="text"> oninput: <span id="result"></span>  
<script>  
  var input = document.body.children[0];  
  input.oninput = function() {  
    document.getElementById('result').innerHTML = input.value;  
  };  
</script>
```

Создать рабочий TODO список. При клике на “Add” пункт добавляется в секцию TODO, при нажатии на чекбокс, пункт перемещается в поле “Completed”. Delete удаляет пункт.



Использовать следующие методы в работе

- document.createElement
- elem.innerHTML
- parent.appendChild(elem)
- .cloneNode(true)

```
$('#cut').click(function(){ content = $('#div').detach(); });
```

```
$('#paste').click(function(){ content.appendTo('body'); })
```



Спасибо!

Есть вопросы?

Вы можете связаться со мной

mail: valeriy.gorbachevskiy@gmail.com

Telegram/Viber: +380954382408

Skype: [valerij.gorbachevskij](https://www.skype.com/people/valerij.gorbachevskij)