

## Лабораторная работа № 8. События. Динамические эффекты на JS

Цель: научиться создавать динамические эффекты средствами JavaScript.

### Теория

События – это функции, которые могут быть привязаны к элементам HTML страниц.

Код событий выполнится только после того, как произойдет их активирующее действие. Разные типы событий имеют разные активирующие действия.

Дополнительную информацию о событиях, которые произошли, содержит объект события **events**, то есть в объекте **events** хранятся атрибуты событий. Например, атрибут **screenX** позволяет узнать горизонтальные координаты указателя мыши во время вызова события относительно границ экрана, атрибут **screenY** – вертикальные координаты, **target** позволяет узнать элемент, который вызвал событие, **type** – позволяет узнать имя события и т. д.

В любой момент времени существует не более одного объекта события. Все инициированные события заносятся операционной системой в буфер и выполняются последовательно в том порядке, в каком они туда попали.

Примеры активирующих действий JavaScript:

- щелчок мыши (событие **onclick**);
- нажатие клавиши (**onkeypress**);
- отправление формы (**onsubmit**);
- наведение курсора мыши на элемент (**onmouseover**);
- выведение курсора мыши за пределы границ элемента (**onmouseout**);
- полная загрузка страницы или картинки (**onload**);
- изменение содержимого элемента, например содержимого

текстового поля формы (**onchange**).

Метод **getElementById(значение\_ID)** позволяет обратиться к любому элементу по значению его идентификатора – значению атрибута **ID**. Если несколько элементов документа имеют одинаковый **ID**, метод возвращает первый элемент с указанным значением **ID**.

Свойство **srcElement** возвращает ссылку на объект элемента HTML-документа, который инициировал событие. При получении такой ссылки можно узнать или изменить значения свойств этого объекта и применить к нему любой из его методов.

Следующий код на JS позволяет при щелчке кнопкой мыши на картинке увеличивать изображение, а затем при щелчке на увеличенном изображении его уменьшать.

```
<script>
```

```
var pic_sm = new Array("mcat1.jpg", "mcat2.jpg", "mcat3.jpg")
```

```

var pic_bg = new Array("cat1.jpg", "cat2.jpg", "cat3.jpg")
var pic_flag = new Array(pic_sm.length) // массив флагов

/* Формирование строки тегов изображений */
var xstr = ""
for (i=0; i<pic_sm.length; i++)
    xstr += ''
    document.write(xstr)

function imgchange() {
    var xid=event.srcElement.id //id изображения, на котором был
щелчок
    var n=parseInt(xid.substr(3)) // выделяем номер элемента
    if (pic_flag[n]) document.all[xid].src=pic_sm[n]
    else document.all[xid].src=pic_bg[n]
    pic_flag[n] = !pic_flag[n]
}
</script>

```

## Задания к лабораторной работе № 8

**Задание 1.** Создать программу на JS, изменяющую цвет слова в тексте при наведении курсора мыши (событие Onmouseover, свойство style.color).

**Задание 2.** Создать программу на JS, изменяющую размер шрифта слова щелчком мыши (событие Onclick, свойство style.fontSize).

**Задание 3.** Создать программу на JS замены картинки на другую при щелчке мышью с помощью метода getElementById и свойства src.

**Задание 4.** Создать программу на JS, заменяющую текст изображением, с использованием метода getElementById и свойства innerHTML.

**Задание 5.** Создать эффекты увеличения размера рисунка (свойство width) при наведении на него курсора мыши и возврата к первоначальному размеру при уходе курсора с рисунка (событие onmouseout)

**Задание 6.** Нарисовать цветную рамку произвольного стиля вокруг любого абзаца при двойном щелчке (событие dblclick) на абзац.