СОБЫТИЯ В БРАУЗЕРЕ

JavaScript

Модуль 2. Урок 2.

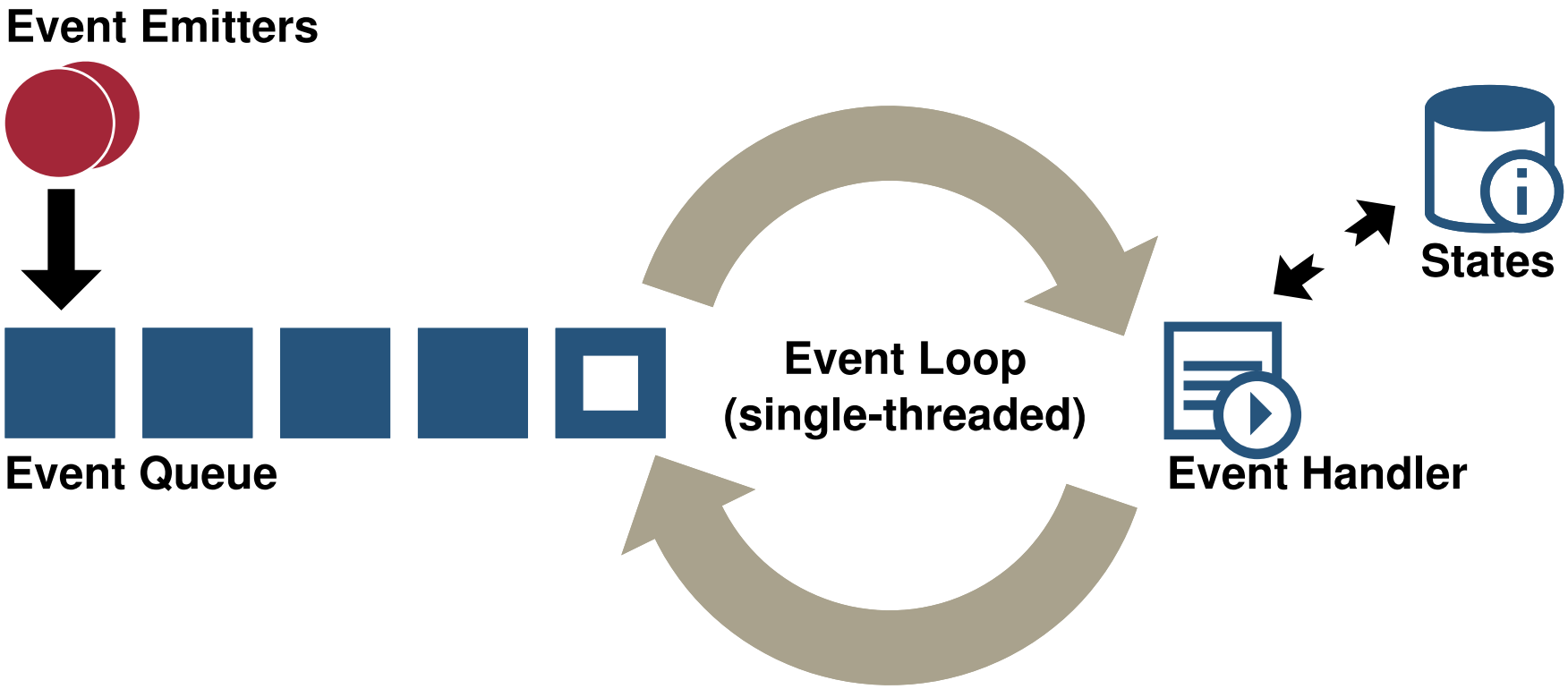
LUXOFT

LXFT LISTED NYSE

СОБЫТИЯ

- Событийно-ориентированное программирование
- Присоединение обработчиков событий
- События браузера
- События загрузки документа
- События формы
- События клавиатуры
- События мыши

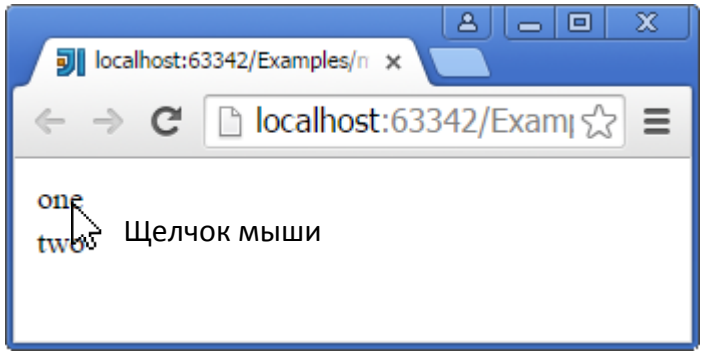
СОБЫТИЙНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

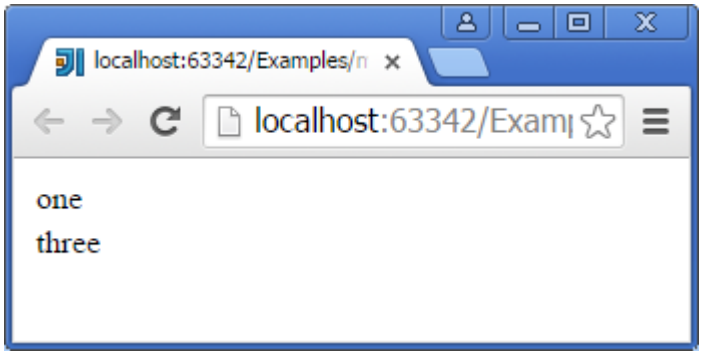


СОБЫТИЙНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

```
<table id="outside">
    <tr><td id="t1">one</td></tr>
    <tr><td id="t2">two</td></tr>
</table>

// Function to change the content of t2
function modifyText() {
  var t2 = document.getElementById("t2");
  if (t2.firstChild.nodeValue == "three") {
    t2.firstChild.nodeValue = "two";
  } else {
    t2.firstChild.nodeValue = "three";
  }
}

// add event listener to table
var el = document.getElementById("outside");
el.addEventListener("click", modifyText, false);
```

СОБЫТИЙНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ





СОБЫТИЙНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

```
<div id="test"></div>
```

```
document.getElementById("test").addEventListener("click", function( event ) {
  // display the current click count inside the clicked div
  event.target.innerHTML = "click count: " + event.detail;
}, false);
```

ОБЪЕКТ-СОБЫТИЕ

Свойство	Описание
bubbles	Булево значение, показывающее передается ли событие вверх по иерархии в модели DOM или нет
cancelable	Булево значение, показывающее, может ли событие быть отменено
defaultPrevented	Показывает, была ли вызвана функция event.preventDefault() на событие или нет
currentTarget	Ссылка на секущую зарегистрированную цель для события
target	Ссылка на цель, для которой первоначально было выдано событие
type	Имя события (не чувствительно к регистру)

ОБЪЕКТ-СОБЫТИЕ

Метод	Описание
preventDefault()	Отменяет событие (если оно может быть отменено)
stopImmediatePropagation()	Для этого конкретного события не будут вызываться никакой другой слушатель событий. Ни прикрепленные к тому же элементу, ни прикрепленные к элементам, которые будут пройдены позже (например, на этапе захвата).
stopPropagation()	Прекращает дальнейшее распространение событий в модели DOM.

РЕГИСТРАЦИЯ ОБРАБОТЧИКОВ СОБЫТИЙ

```
// Pass a function reference – do not add '()' after it,
// which would call the function!
el.onclick = modifyText;

// Using a function expression
element.onclick = function() {
  // ... function logic ...
};

<p onclick="paragraphClicked(this);">Paragraph</p>

<script>
  function paragraphClicked(that) {
    // accessing to event object using global scope
    window.event.target.innerHTML = "Hello";
    that.innerHTML = "Hello";
  }
</script>
```

ПРИСОЕДИНЕНИЕ ОБРАБОТЧИКОВ СОБЫТИЙ

- По соглашению свойства обработчика событий имеют имена, которые состоят из слова “on”, за которым идет имя события: onclick, onchange, onload, onmouseover и т. д.

```
// Set the onload property of the Window object to a function.
// The function is the event handler: it is invoked when the document loads.
window.onload = function() {
  // Look up a <form> element
  var elt = document.getElementById("shipping_address");
  // Register an event handler function that will be invoked right
    // before the form is submitted.
  elt.onsubmit = function() { return validate(this); }
}
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ ОБРАБОТЧИКА

- Для обработчиков событий, зарегистрированных как свойства, возвращаемое значение указывает браузеру, что он должен или не должен выполнить связанное с данным событием действие по умолчанию.

```
<a href="http://google.com" id="link">Don't go to Google</a>

<script>
  document.getElementById("link").onclick = cancelHandler;

  function cancelHandler(event) {
    return false; // cancel the default action associated with the event
  }
</script>
```

ПРИСОЕДИНЕНИЕ ОБРАБОТЧИКОВ СОБЫТИЙ

Новый способ регистрации слушателей событий

```
document.getElementById("myBtn").addEventListener("click", myFunction);

function myFunction() {
  document.getElementById("demo").innerHTML = "Hello World";
}

// using anonymous function
document.getElementById("myBtn").addEventListener("click", function() {
  myFunction();
});
```

ПРИСОЕДИНЕНИЕ ОБРАБОТЧИКОВ СОБЫТИЙ

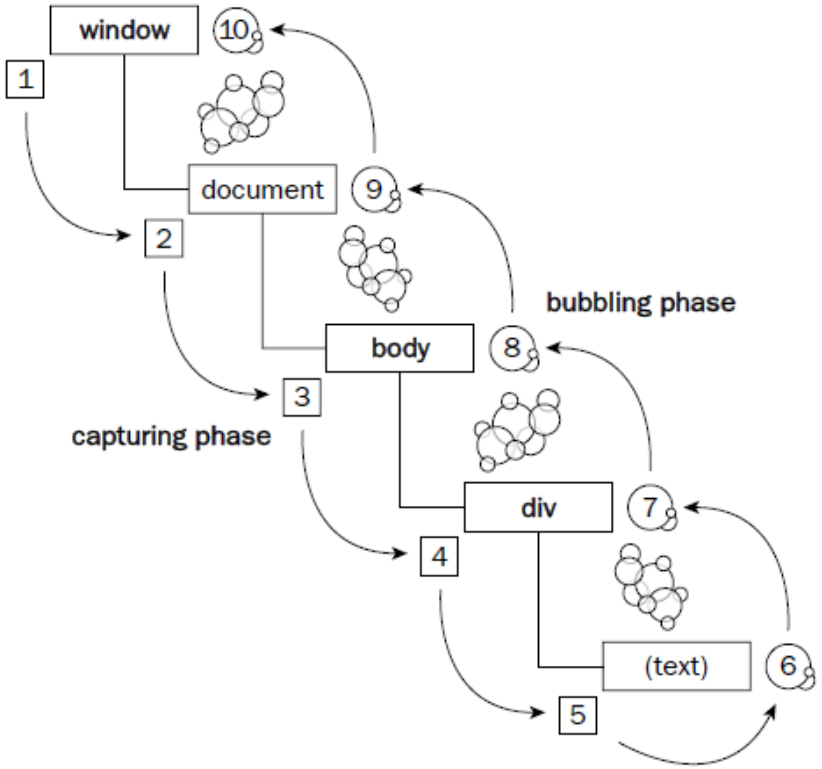
```
<button id="mybutton">Click me</button>
<script>
  var b = document.getElementById("mybutton");
    // onclick
  b.onclick = function () {
    alert("Thanks for clicking me!");
  };
    // but click
  b.addEventListener("click", function () {
    alert("Thanks again!");
  }, false);
</script>
```

УДАЛЕНИЕ ОБРАБОТЧИКОВ СОБЫТИЙ

```
// Attach an event handler to <div>
document.getElementById("myDIV").addEventListener("click", myFunction);

// Remove the event handler from <div>
document.getElementById("myDIV").removeEventListener("clic", myFunction);
```

BUBBLING И CAPTURING

```
<body>
 <div>
     Text
 </div>
</body>
```

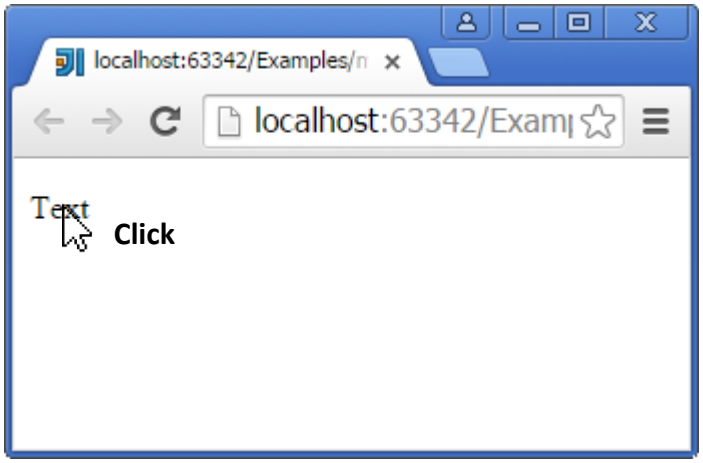


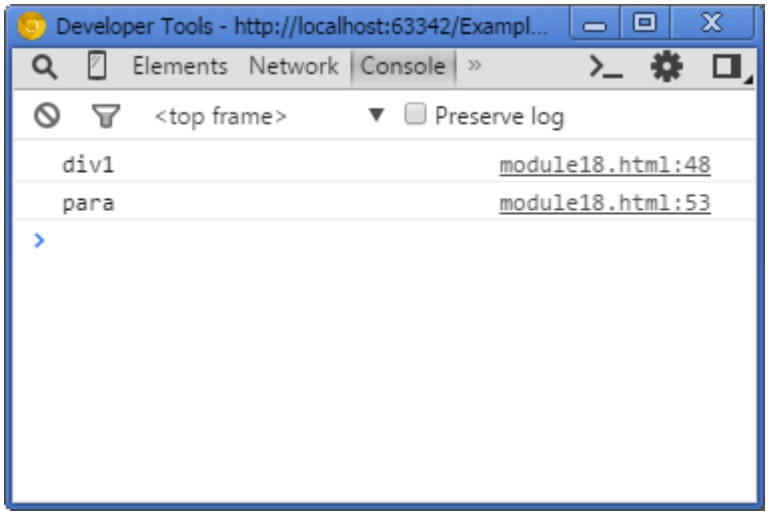
- Щелчок мыши по пустому пространс
  это по сути щелчок по каждому элем
- События передаются из нижней част
  дерева DOM вверх.

BUBBLING И CAPTURING

```
<div id="div1">
    <div id="div2">
        <p id="para">Text</p>
    </div>
</div>

<script>
  function sayDiv1(event) { console.log("div1"); }

  function sayDiv2(event) { console.log("div2"); }

  function sayPara(event) {
    console.log("para");
    event.stopPropagation();
  }

  document.getElementById("div1").addEventListener("click", sayDiv1, true);
  document.getElementById("div2").addEventListener("click", sayDiv2);
  document.getElementById("para").addEventListener("click", sayPara);
</script>
```

BUBBLING И CAPTURING





СОБЫТИЯ БРАУЗЕРА

- Некоторые типы событий ориентированы на браузер в целом, а не на какой-то определенный элемент документа.
- В JavaScript обработчики таких событий регистрируются в объекте Window. В HTML они размещаются в тэге <body>, но браузер регистрирует их в Window.

СОБЫТИЯ БРАУЗЕРА

Обработчик событий	Описание
onload	Браузер завершил загрузку документа.
onunload	Документ выгружается.
beforeunload	Окно, документ и его ресурсы сейчас будут выгружены.
onpageshow	Выполняется переход к записи истории сеанса.
onpagehide	Выполняется переход от записи истории сеанса.
onpopstate	Активная запись истории меняется между двумя записями истории для одного документа.
onhashchange	Хэш окна меняется.
onafterprint	Пользователь выполняет печать или выходит из диалогового окна печати.
onbeforeprint	Документ сейчас будет распечатан или просмотрен перед выводом на печать.
ononline	Браузер получил доступ к сети.
onoffline	Браузер потерял доступ к сети.

СОБЫТИЯ БРАУЗЕРА

Обработчик событий	Описание
onfocus	Документ получил фокус.
onblur	Документ потерял фокус.
onresize	Изменен размер вида документа.
onmessage	Получено сообщение.
onerror	Произошла ошибка при загрузке документа или изображения.
onstorage	Область Веб-хранилища обновлена.
onundo	Документ выполняет операцию undo.
onredo	Документ выполняет операцию redo.

СОБЫТИЯ БРАУЗЕРА

```
<body onload="checkCookies()">

<script>
  function checkCookies() {
    var text = "";

    if (navigator.cookieEnabled == true) {
      text = "Cookies are enabled.";
    } else {
      text = "Cookies are not enabled.";
    }

    document.getElementById("demo").innerHTML = text;
  }
</script>
```

СОБЫТИЯ БРАУЗЕРА

```
<button onclick="changePart()">Try it</button>

<p id="demo"></p>

<script>
  // Using the location.hash property to change the anchor part
  function changePart() {
    location.hash = "hash";
    var x = location.hash;
    document.getElementById("demo").innerHTML = "The hash is now: " + x;
  }

  window.onhashchange = logHash;

  function logHash() {
    console.log("The hash part has changed!");
  }
</script>
```

СОБЫТИЯ ФОРМЫ

Элемент формы	Обработчик событий	Описание и события
Buttons, Submit, Reset	onclick	Пользователь щелкает элемент.
Checkbox, Radio buttons, Input, List box, Textarea	onchange	Содержимое элемента формы, выбор или состояние выбора изменились.
Все элементы	onfocus	Элемент получает фокус.
Все элементы	onblur	Элемент теряет фокус.
form	onsubmit	Форма отправлена.
form	onreset	Выполнен сброс формы.

СОБЫТИЯ ФОРМЫ

```
<form name="myForm3" onsubmit="return validateForm()">
  First name: <input type="text" name="fname">
    <input type="submit" value="Submit">
</form>

<script>
  function validateForm() {
    var x = document.forms["myForm3"]["fname"].value;
    if (x == null || x == "") {
      alert("First name must be filled out");
      return false;
    }
    }
</script>
```

СОБЫТИЯ ФОРМЫ

```
Server name: <input type="text" name="server" id="server">
<script>
  document.getElementById("server").addEventListener("change", function(event) {
    get(event.target.value);
  });

  function get(server) {
    // request something from server
    console.log(server);
  }
</script>
```

СОБЫТИЯ КЛАВИАТУРЫ

Событие	Описание
keydown	Клавиша нажата.
keyup	Клавиша отпущена.
keypress	Клавиша нажата и, как правило, производит значение символа.

СОБЫТИЯ КЛАВИАТУРЫ

Свойство	Описание
char	Буквенное значение клавиши
key	Значение ключа клавиши, представленной событием
charCode	Справочный номер клавиши по стандарту Unicode; этот атрибут используется только для события нажатия клавиши (считается устаревшим).
keyCode	Числовой код, зависящий от системы и реализации, который указывает немодифицированное значение нажатой клавиши (считается устаревшим).
ctrlKey	true, если клавиша control была нажата при выдаче события.
shiftKey	true, если клавиша shift была нажата при выдаче события.
altKey	true, если клавиша alt была нажата при выдаче события.
metaKey	true, если клавиша meta была нажата при выдаче события.

СОБЫТИЯ КЛАВИАТУРЫ

```
<input type="text" id="demo" onkeydown="keydownFunc()" onkeyup="keyupFunc()">

<script>
  function keydownFunc() {
    document.getElementById("demo").style.backgroundColor = "red";
  }

  function keyupFunc() {
    document.getElementById("demo").style.backgroundColor = "green";
  }
</script>
```

СОБЫТИЯ КЛАВИАТУРЫ

```
<input type="text" id="demo3">
<script>
  document.getElementById("demo3").addEventListener("keypress", myFunction);
  function myFunction(event) {
    var color = "white";
    switch (event.keyCode) {
      case 114:
        color = "red";
        break;
      case 103:
        color = "green";
        break;
      case 98:
        color = "blue";
        break;
      default:
        color = "white";
    }
    document.getElementById("demo3").style.backgroundColor = color;
  }
</script>
```

СОБЫТИЯ МЫШИ

Тип	Описание
click	Событие более высокого уровня, выдаваемое, когда пользователь нажимает и отпускает кнопку мыши или иным образом «активирует» элемент.
dblclick	Выдается, когда пользователь делает двойной щелчок мышью.
contextmenu	Отменяемое событие, которое выдается перед тем, как появится contextmenu. Современные браузеры отображают контекстные меню при нажатии правой кнопки мыши, поэтому это событие также может использоваться как событие щелчка мыши.
mousedown	Выдается, когда пользователь нажимает кнопку мыши.
mouseup	Выдается, когда пользователь отпускает кнопку мыши.
mousemove	Выдается, когда пользователь двигает мышью.

СОБЫТИЯ МЫШИ

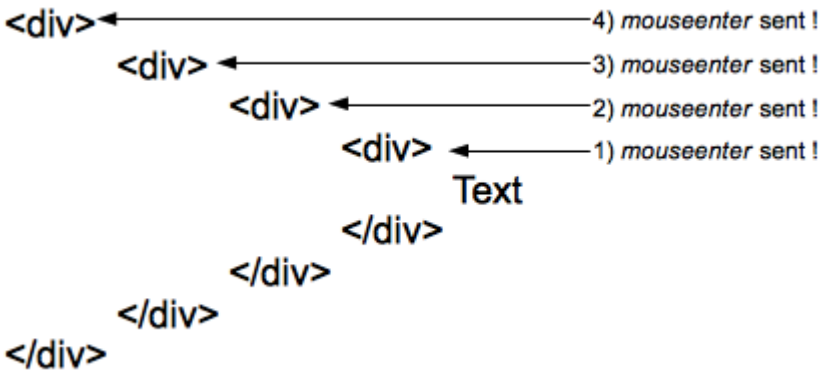
Тип	Описание
mouseover	Выдается, когда указатель мыши наводится на элемент.
mouseout	Выдается, когда указатель мыши выходит за пределы элемента.
mouseenter	То же, что и “mouseover”, но без перехода вверх
mouseleave	То же, что и “mouseout”, но с переходом вверх

MOUSEEVENT

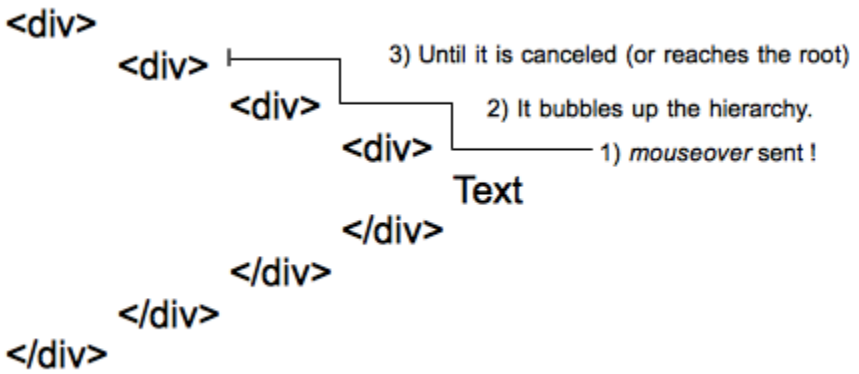
Свойство	Описание
screenX	Координата указателя мыши по оси X в глобальной (экранной) системе координат
screenY	Координата указателя мыши по оси Y в глобальной (экранной) системе координат
clientX	Координата указателя мыши по оси X в локальной системе координат (в контенте модели DOM)
clientY	Координата указателя мыши по оси Y в локальной системе координат (в контенте модели DOM)
button	Номер кнопки, которая была нажата при выдаче события мыши: Левая кнопка=0, средняя кнопка=1 (если есть), правая кнопка=2
ctrlKey, shiftKey, altKey, metaKey	То же, что и в KeyboardEvent

MOUSEENTER & MOUSEOVER

mouseenter

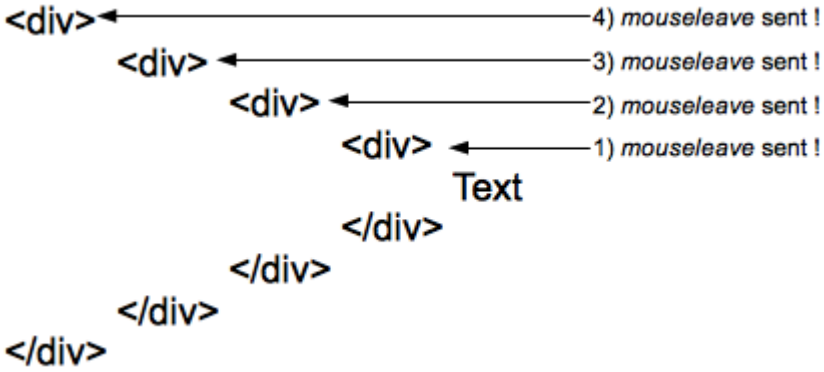


mouseover

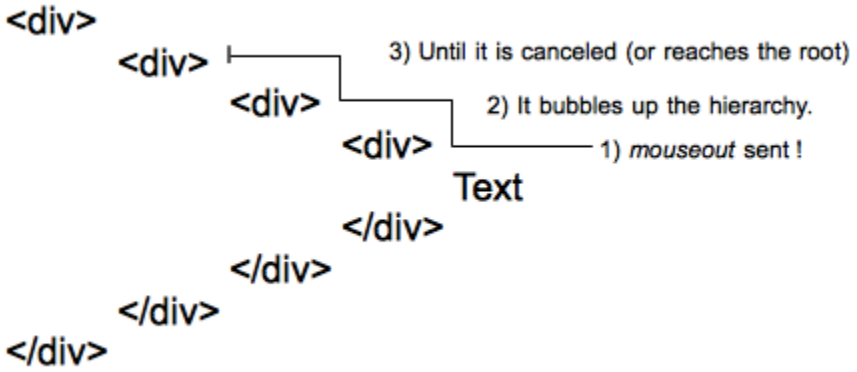


MOUSELEAVE & MOUSEOUT

mouseleave



mouseout



СОБЫТИЯ МЫШИ

```
<div id="enter" style="border: solid">
    <h3 id="demo4">mouseenter</h3>
    <h3>Mouse over me</h3>
</div>

<div id="over" style="border: solid">
    <h3 id="demo5">mouseover</h3>
    <h3>Mouse over me</h3>
</div>
```

СОБЫТИЯ МЫШИ

```
demo4 = document.getElementById("demo4");
demo5 = document.getElementById("demo5");
document.getElementById("enter").addEventListener("mouseenter", mouseEnter);
document.getElementById("enter").addEventListener("mouseleave", mouseLeave);
document.getElementById("over").addEventListener("mouseover", mouseOver);
document.getElementById("over").addEventListener("mouseout", mouseOut);

function mouseEnter() { blink(demo4, "red", "black"); }
function mouseLeave() { blink(demo4, "yellow", "black"); }
function mouseOver() { blink(demo5, "red", "black"); }
function mouseOut() { blink(demo5, "yellow", "black"); }
var working = false;
function blink(el, color1, color2) {
  if(!working) {
    working = true;
    el.style.color = color1;
    setTimeout(function () {
      el.style.color = color2;
      working = false;
    }, 500)
  }
```