

# JavaScript. Уровень 2

## Расширенные ВОЗМОЖНОСТИ

Игорь Борисов  
<http://igor-borisov.ru>

# Темы курса

- Объектная модель браузера
- Использование элементов HTML
- Объектная модель документа: DOM
- События и их обработка

# Объектная модель браузера

Игорь Борисов

<http://igor-borisov.ru>

# Темы модуля

- Объектная модель браузера
- Объект Window
- Объект Navigator
- Объект Screen
- Объект History
- Объект Location
- Объект Document
- HTML-коллекции

# Псевдопротокол javascript

- javascript:
- `<a href="javascript:someFunction()">`

# Объект Window

- Объект Window
  - Объект Navigator (свойство navigator)
  - Объект Screen (свойство screen)
  - Объект History (свойство history)
  - Объект Location (свойство location)
  - Объект Document (свойство document)
  - другие свойства
  - методы

# Объект History

- СВОЙСТВО
  - length
- Методы
  - back()
  - forward()
  - go(**1**)

# Объект Navigator

- Свойства
  - `appName`
  - `appVersion`
  - `userAgent`
  - `cookieEnabled`
  - ...
- Метод
  - `javaEnabled()`

# Объект Screen: свойства

- width
- height
- availWidth
- availHeight
- colorDepth

# Объект Location: свойства

- protocol
- host
- hostname
- port
- pathname
- search
- hash
- href

# Объект Location: методы

- reload (`false`)
- assign (`'http://site.ru/'`)
- replace (`'http://site.ru/'`)
- toString()
  - location.href

# Объект Document: обзор

- Свойства
  - lastModified
  - title
  - domain
  - ...
- Методы
  - open()
  - write(string, [string[, ...]])
  - close()

# Объект Window

- Свойства
  - frames
  - status
  - defaultStatus
- Методы
  - Диалоговые окна
  - Таймеры
  - Манипуляции с окном
  - print()

# Диалоговые окна

- `alert ("Привет")`
- `confirm ("Вы уверены?")`
  - `true`
  - `false`
- `prompt ("Введите ваш возраст", "25")`
  - `string`
  - `null`

# Таймеры

- `x = setTimeout(code, 1000)`
- `clearTimeout(x)`
  
- `x = setInterval(code, 1000)`
- `clearInterval(x)`

# Создание нового окна

- СВОЙСТВО
  - closed
  - opener
- Методы
  - `open('url', 'name', 'params', history)`
  - `close()`
  - `focus()`
  - `blur()`

# Параметры открываемого окна

- width (число)
- height (число)
- left (число)
- top (число)
- location (1|0 или yes|no)
- menubar (1|0 или yes|no)
- scrollbars (1|0 или yes|no)
- toolbar (1|0 или yes|no)
- status (1|0 или yes|no)
- resizable (1|0 или yes|no)

# Методы манипуляции окном

- `moveTo(x, y)`
- `moveBy(x, y)`
  
- `resizeTo(x, y)`
- `resizeBy(x, y)`
  
- `scrollTo(x, y)`
- `scrollBy(x, y)`

# Лабораторная работа – 1

- Использование свойств и методов объектов браузера

# HTML-коллекция frames

- `<iframe name="top" src="top.html"></iframe>`  
`<iframe name="bottom" src="bottom.html"></iframe>`
- `frames[0]` // как к элементу массива  
`frames['top']` // как к свойству объекта  
`frames.top` // как к свойству объекта

# Отношения между окнами

- window
- self
- parent
- top
- name

# Выводы

- Объектная модель браузера
- Объект Window
- Объект Navigator
- Объект Screen
- Объект History
- Объект Location
- Объект Document
- HTML-коллекции

# Использование элементов HTML

Игорь Борисов  
<http://igor-borisov.ru>

# Темы модуля

- Свойства и методы элемента Form
- Свойства и методы элемента Input
- Свойства и методы элемента TextArea
- Свойства и методы элемента Select
- Свойства и методы элемента Option
- Свойства и методы элемента Image
- Свойства и методы элемента Table

# HTML-коллекции

- forms
- images
- links
- applets
- embeds
- `<form name="search">`
  - `document.forms[0]`
  - `document.forms['search']`
  - `document.forms.search`
- `document.links[0].style.color = "red"`

# Правила именования свойств

- `element.style.padding = '10px';`
- Правила именования
  - `border-top-width => borderTopWidth`
- Разрешение конфликта имен атрибутов
  - `for => htmlFor`
  - `float => cssFloat`
  - Исключение
    - `class => className`

# Элементы HTML

- HTMLFormElement
  - HTMLInputElement
  - HTMLTextAreaElement
  - HTMLSelectElement
  - HTMLOptionElement
- HTMLImageElement
- HTMLTableElement
  - HTMLTableSectionElement
  - HTMLTableRowElement
  - HTMLTableCellElement

# HTMLFormElement

## ■ Свойства

- elements
- length
- name
- action
- enctype
- method
- target

## ■ Методы

- submit()
- reset()

# HTMLInputElement

## ■ Свойства

- **form** (JS)
- **defaultValue** (JS)
- **defaultChecked**
- **checked**
- **maxLength**
- **type**
- **value**
- **size**
- **name**

## ■ Методы

- **select()**
- **click()**
- **focus()**
- **blur()**

# HTMLTextAreaElement

## ■ Свойства

- **defaultValue** (JS)
- **form** (JS)
- **value** (JS)
- **type** (JS)
- cols
- rows
- readOnly
- tabIndex
- disabled

## ■ Методы

- select()
- focus()
- blur()

# Лабораторная работа – 2-1

- Проверка заполнения полей формы перед отправкой

# HTMLSelectElement

## ■ Свойства

- **options** (JS)
- **length** (JS)
- **type** (JS)
- **selectedIndex** (JS)
- **value** (JS)
- **form** (JS)
- **disabled** и **multiple**
- **name**
- **size** и **tabIndex**

## ■ Методы

- **add()**
- **remove()**
- **focus()**
- **blur()**

# HTMLOptionElement

- СВОЙСТВА
  - **form** (JS)
  - **text** (JS)
  - **defaultSelected** (JS)
  - **index** (JS)
  - **value**
  - **selected**
  - **disabled**
  - **label**

# Конструктор Option

- Option(text, value, defaultSelected, selected)
- `var o = new Option('PHP5', 'php');`
- `select.add(newOpt, curOpt);`  
`select.add(newOpt, curOptIdx); // IE`
- `select.remove(index);`

# Лабораторная работа – 2-2

- Добавление и удаление элементов списка

# HTMLImageElement

- `var img = new Image(300, 200);`
- СВОЙСТВА
  - src
  - alt
  - width
  - height
  - border

# HTMLTableElement

## ■ Свойства

- tHead
- tFoot
- tBodies[]
- rows[]
- caption
- width
- border
- cellPadding
- cellSpacing

## ■ Методы

- createCaption()
- deleteCaption()
- createTHead()
- deleteTHead()
- createTFoot()
- deleteTFoot()
- insertRow(index)
- deleteRow(index)

# HTMLTableSectionElement

## ■ Свойства

- rows
- align
- vAlign

## ■ Методы

- insertRow(index)
- deleteRow(index)

# HTMLTableRowElement

## ■ Свойства

- cells
- rowIndex
- sectionRowIndex
- vAlign
- bgColor

## ■ Методы

- insertCell(index)
- deleteCell(index)

# HTMLTableCellElement

- Свойства
  - cellIndex
  - colSpan
  - rowSpan
  - align
  - vAlign
  - width
  - height
  - textContent
  - **innerHTML**

# Выводы

- Свойства и методы элемента Form
- Свойства и методы элемента Input
- Свойства и методы элемента TextArea
- Свойства и методы элемента Select
- Свойства и методы элемента Option
- Свойства и методы элемента Image
- Свойства и методы элемента Table

# Объектная модель документа: DOM

Игорь Борисов

<http://igor-borisov.ru>

# Темы модуля

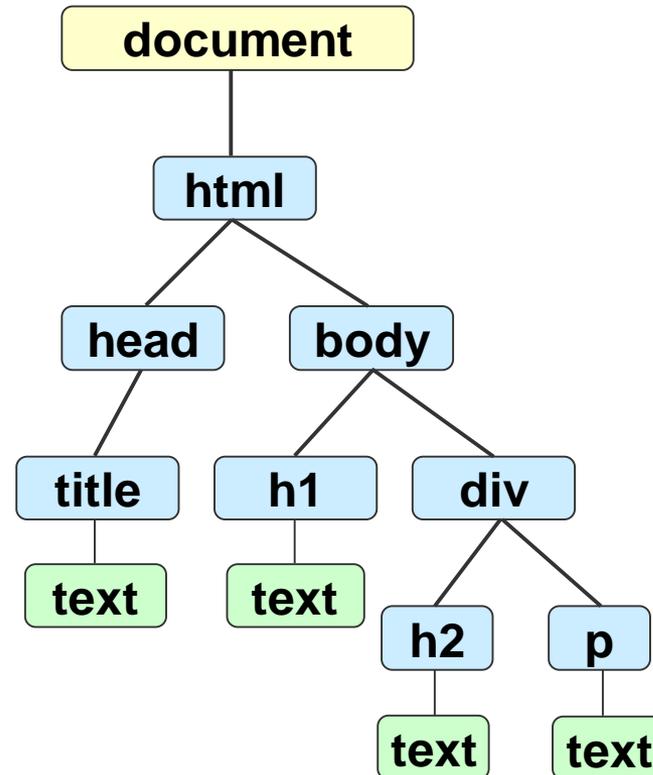
- Document Object Model (DOM)
- Типы узлов
- Связи между объектами
- Свойства и методы интерфейса Node
- Свойства и методы интерфейса Element
- Свойства и методы интерфейса Document
- Нестандартные свойства и методы HTML-элементов
- Свойства и методы коллекции styleSheets

# Уровни DOM

<b>DOM 0</b>	Традиционная модель JavaScript. Эквивалент модели, предложенной в Netscape 3.0 и IE 3.0. Поддерживает общие коллекции объектов документа: forms[], images[], anchors[], links[], applets[]
<b>DOM 1</b>	Обеспечивает возможность работы со всеми элементами документа посредством стандартного набора функций
<b>DOM 2</b>	Обеспечивает более совершенные возможности доступа к элементам страницы, а также возможности доступа к таблицам стилей.
<b>DOM 3</b>	Улучшение поддержки XML (на основе XPath). Обеспечение возможностей для обмена содержимым файлов.

# Document Object Model

```
<html>  
  
  <head>  
    <title>PAGE</title>  
  </head>  
  
  <body>  
    <h1>PAGE</h1>  
    <div>  
      <h2>Раздел</h2>  
      <p>Параграф</p>  
    </div>  
  </body>  
  
</html>
```

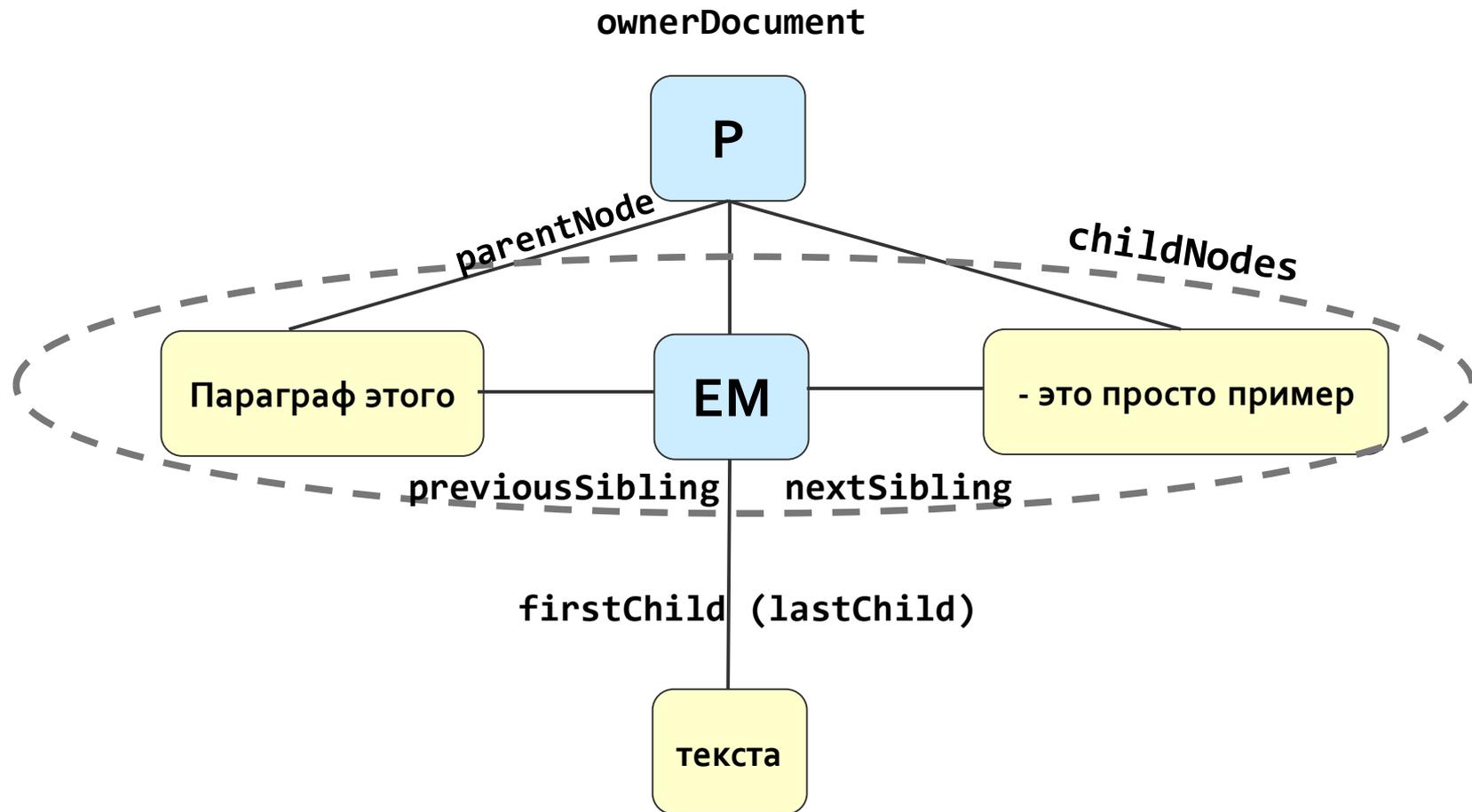


# Типы узлов

Код типа	Тип узла	Описание	Пример
1	ELEMENT	Элемент	<code>&lt;p&gt; . . . &lt;/p&gt;</code>
2	ATTRIBUTE	Атрибут элемента	<code>align="center"</code>
3	TEXT	Текстовый узел	Это текст
8	COMMENT	Комментарий	<code>&lt;!-- comment --&gt;</code>
9	DOCUMENT	Узел документа	document
10	DocumentType	Декларация типа документа	<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;</code>

# Связи между объектами

`<p>Параграф этого <em>текста</em> - это просто пример</p>`



# Node – узел документа

## ■ Свойства

- parentNode
- childNodes
- firstChild
- lastChild
- nextSibling
- previousSibling
- ownerDocument
  
- nodeName
- nodeType
- nodeValue

## ■ Методы

- hasChildNodes()
- cloneNode(bool)
- appendChild(e)
- insertBefore(e, p)
- replaceChild(e, p)
- removeChild(e)

# Element – элемент документа

## ■ Свойства

- Свойства Node

+

- tagName

## ■ Методы

- Методы Node

+

- `getElementsByTagName(t)`
- `hasAttribute(a)`
- `getAttribute(a)`
- `setAttribute(a, v)`
- `removeAttribute(a)`

# Интерфейс Document

- Свойства Node

+

- **documentElement**

- **body**

- **styleSheets**

- **doctype**

- Методы Node

+

- `getElementsByTagName(e)`

- `getElementById(id)`

- другие

# Лабораторная работа 3-1

- Использование свойств и методов интерфейса Element

# Создание узлов

- Методы интерфейса Document
  - `createElement(element_name)`
  - `createTextNode(string)`
  - `createDocumentFragment()`
  - `createComment(string)`

# Лабораторная работа 3-2

- Создание элементов DOM

# Специфика элемента `iframe`

## ■ Свойства

- `contentDocument` (`contentWindow.document`)
- `contentDocument.defaultView` (`contentWindow`)

## ■ Использование

- `var f = document.getElementById('f');`
- `//доступ к элементам загруженной страницы`  
`f.contentDocument | contentDocument.document;`

# Выводы

- Document Object Model (DOM)
- Типы узлов
- Связи между объектами
- Свойства и методы интерфейса Node
- Свойства и методы интерфейса Element
- Свойства и методы интерфейса Document

# События и их обработка

Игорь Борисов

<http://igor-borisov.ru>

# Темы модуля

- Список событий
- Модели событий
- Назначение обработчиков событий
- Работа с обработчиками событий
- Получение ссылки на событие
- Отмена действий по умолчанию
- Кроссбраузерные свойства события
- Получение ссылки на элемент
- Использование фазы всплытия событий
- Отмена всплытия события

# Типы событий I

- Срабатывают на большинстве элементов
  - onclick (можно отменить)
  - onmousedown
  - onmouseup
  - onmousemove
  - onmouseover
  - onmouseout

# Типы событий II

- `<body>`, `<input>`, `<textarea>`
  - `onkeydown` (можно отменить)
  - `onkeypress` (можно отменить)
  - `onkeyup`
  
- `<body>`, `<a>`, `<area>`, `<button>`, `<input>`, `<label>`,  
`<select>`, `<textarea>`
  - `onfocus`
  - `onblur`

# Типы событий III

- `<body>`, `<frameset>`, `<iframe>`, `<img>`, `<object>`
  - `onload`
- `<body>`, `<frameset>`
  - `onunload`
- `<img>`, `<object>`
  - `onabort`
- `<body>`, `<frameset>`, `<img>`, `<object>`
  - `onerror`
- `<body>`, `<frameset>`, `<iframe>`
  - `onresize`

# Типы событий IV

- `<form>`
  - `onsubmit` (можно отменить)
  - `onreset` (можно отменить)
- `<input>`, `<textarea>`
  - `onselect`
- `<input>`, `<textarea>`, `<select>`
  - `onchange`
- `<body>`, элементы с прокруткой
  - `onscroll`

# Типы событий: DOM 2

- Window
  - onfocus
  - onblur
  - onload
  - onunload
  - onerror
  - onresize
  - onscroll
- Document
  - onkeydown
  - onkeypress
  - onkeyup

# Типы моделей событий

- Базовая (исходная)
- W3C DOM
- Internet Explorer

# Базовая (исходная) модель

- Использование

- `<a onclick="alert('Ура')">`

- Отмена события

- `<form onsubmit="alert('!');return false">`

- ```
function test(){
    alert('Не отправляется');
    return false;
}
```

- `<form onsubmit="return test();">`

# Передача ссылки на элемент

- ```
function test(e1){  
    alert(e1.nodeType);  
}
```
- ```
<p onclick="test(this);">Text</p>
```

# Обработчик как свойство

## ■ Вариант 1

- `<form name="search">`

- ```
function test(){  
    alert(this);  
}
```

- `document.forms.search.onsubmit = test;`

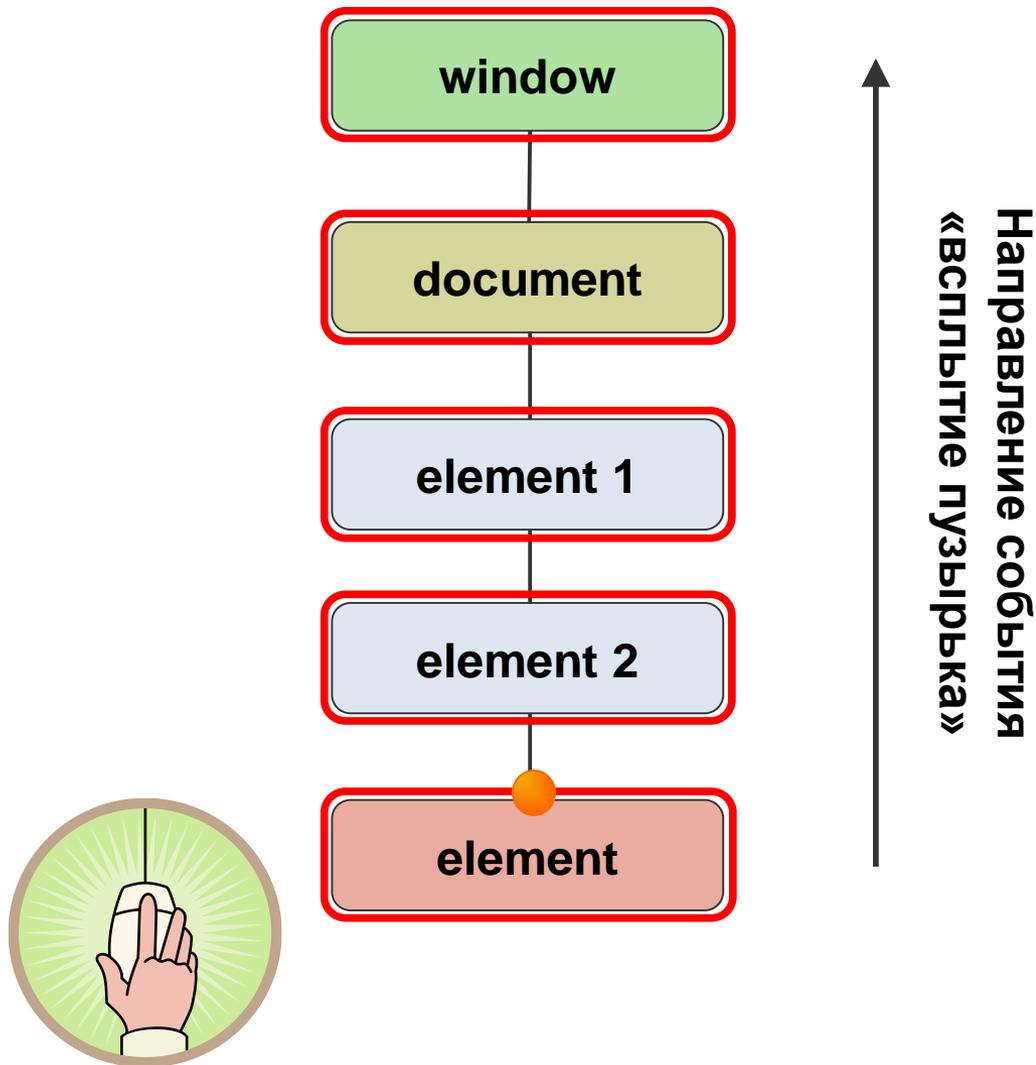
## ■ Вариант 2

- `new Function("test(this, 'param')")`

# Другие модели событий

- Модель W3C DOM
  - Фаза перехвата или распространения
  - Фаза регистрации
  - Фазы всплытия
- Модель IE
  - Фаза регистрации
  - Фазы всплытия

# Модель обработки событий



# Регистрация событий

- `var el = document.getElementById('x');`
- `function foo(){`  
    `//Здесь исполняемый код`  
`}`
- DOM 2
  - `el.addEventListener('click', foo);`
- IE
  - `el.attachEvent('onclick', foo);`

# Отмена регистрации событий

- `var el = document.getElementById('x');`
- `function foo(){`  
    //Здесь исполняемый код  
`}`
- W3C DOM
  - `el.removeEventListener('click', foo);`
- IE
  - `el.detachEvent('onclick', foo);`

# Получение ссылки на событие

- `function foo(e){`  
    `e = e || event;`  
`}`
- `event == window.event`

# Отмена действий по умолчанию

- Получение ссылки на событие

- ```
function foo(e){  
    e = e || event;  
}
```

- W3C DOM

- ```
e.preventDefault();
```

- IE

- ```
e.returnValue = false;
```

# Свойства события

- Любой источник
  - type
  - target (W3C DOM) / srcElement (IE)
  - currentTarget (W3C DOM) / srcElement (IE)
- Источник мышь
  - button
  - altKey, ctrlKey, shiftKey
  - clientX, clientY
- Источник клавиатура
  - keyCode

# Лабораторная работа 4-1

- Раскрывающееся меню в виде списка

# Отмена всплытия события

- Получение ссылки на событие

- ```
function foo(e){  
    e = e || event;  
}
```

- W3C DOM

- ```
e.stopPropagation();
```

- IE

- ```
e.cancelBubble = true;
```

# Кроссбраузерная функция

- ```
function addEvent(e1, e, handler){  
    if (e1.addEventListener) {  
        e1.addEventListener(e, handler);  
    } else if (element.attachEvent) {  
        e1.attachEvent('on' + e, handler);  
    } else {  
        e1['on' + e] = handler; }  
}
```
- ```
addEvent(element, 'click', function(e){...});
```

# Лабораторная работа 4-2

- Выпадающее меню

# Выводы

- Список событий
- Модели событий
- Назначение обработчиков событий
- Работа с обработчиками событий
- Получение ссылки на событие
- Отмена действий по умолчанию
- Кроссбраузерные свойства события
- Получение ссылки на элемент
- Использование фазы всплытия событий
- Отмена всплытия события