

JAVA SCRIPT

Введение

Язык программирования JavaScript был разработан фирмой Netscape в сотрудничестве с Sun Microsystems и анонсирован в 1995 году. JavaScript предназначен для создания интерактивных html-документов. Основные области использования JavaScript:

- Создание динамических страниц, т.е. страниц, содержимое которых может меняться после загрузки.
- Проверка правильности заполнения пользовательских форм. Решение "локальных" задач с помощью сценариев.
- JavaScript-код - основа большинства Ajax-приложений.

Что делает JavaScript !!

- **JavaScript дает HTML дизайнерам инструмент программирования** - HTML авторы, как правило, не программисты, но JavaScript это язык сценариев с очень простым синтаксисом! Почти каждый может положить небольшие "фрагменты" кода на своих страницах HTML
- **JavaScript может поставить динамический текст в HTML страницу** - А JavaScript заявление так:
`document.write("<h1>" + имя + "</ h1>");`
может написать переменный текст в HTML страницу
- **JavaScript может реагировать на события** - JavaScript может быть установлен, чтобы выполнить, когда что-то происходит, например, когда завершения загрузки страницы или когда пользователь нажимает на HTML элемент

- **JavaScript может читать и писать HTML элементы** - JavaScript может читать и изменять содержимое на HTML элемент
- **JavaScript может использоваться для проверки данных** - JavaScript может быть использован для проверки данных формы, до его передачи на сервер. Это экономит сервер от дополнительной обработки
- **JavaScript может быть использован для обнаружения браузер посетителя** - JavaScript может быть использован для обнаружения браузер посетителя, и - в зависимости от браузера - загрузку еще одной страницы, специально предназначенные для этого браузера
- **JavaScript может быть использован для создания cookie** - JavaScript может быть использован для хранения и извлечения информации о компьютере посетителя

Hello World

- `<html>`
- `<body>`
- `<script type="text/javascript">`
- `document.write("Hello World!");`
- `</script>`
- `</body>`
- `</html>`
- Комментарий:

Одна строчный комментарий начинаются с //

Многострочный комментарий /*-комментарий- */

- JavaScript в секции body будут выполнены во время загрузки страниц.
- JavaScript в header секции будут выполнены при вызове JavaScript представляет собой последовательность инструкций для выполнения в браузере.
- `<script type="text/javascript">`
- `document.write("<h1>This is a header</h1>");`
`document.write("<p>This is a paragraph</p>");`
`document.write("<p>This is another paragraph</p>");`
- `</script>`

- Заявления JavaScript могут быть сгруппированы в блоки.
 - `<script type="text/javascript">`
 - `{`
 - `document.write("<h1>This is a header</h1>");`
`document.write("<p>This is a paragraph</p>");`
`document.write("<p>This is another paragraph</p>");`
 - `}`
 - `</script>`
 - Использование внешнего файла
- ```
<script language="JavaScript" src="filename.js">
 </script>
```

- myHtml.html :

- <html>
- <head>
- <script src = "test.js"></script>
- </head>
- <body>
- Enjoy !!!!!
- </body>
- </html>

- test.js :

- document.write("hello i am from test");



# Объявление переменных

- `var x=5;`
  - `var name="abcd"`
  -
- 

- `<html>`
- `<body>`
- `<script type="text/javascript">`
- `{`
- `var x="the number is" , y=50;`
- `document.write(x+y);`
- `}`
- `</script>`
- `</body>`
- `</html>`

- `<html>`
- `<head>`
- `<script language = "javascript">`
- `var x = 12.5 + 7.5;`
- `document.write(x);`
- `</script>`
- `</head>`
- `</html>`
- Каков результат::
- ..... "12.5" .....

- What is the output ?
- <html>
- <head>
- <script language = "javascript">
- var x = "12.5";
- var y = 7.5
- var a = parseFloat(x);
- document.write(a+y);
- </script>
- </head>
- </html>

- <html>
- <head>
- <script language = "javascript">
- var x = "ab12.5";
- var a = parseFloat(x);
- document.write(a);
- </script>
- </head>
- </html>
- ..... NaN (not a number).....

# JavaScript всплывающих окон

- В JavaScript мы можем создать три вида всплывающих окон:
  - **Alert box (окно оповещения)**
  - **Confirm box (окно подтверждения)**
  - **Prompt box (контекстное окно).**

- **Alert box** : Окно оповещения часто используется, если вы хотите, чтобы убедиться, оповещения поступает через пользователь. Когда окно предупреждения всплывает, пользователь должен будет нажать кнопку "ОК", чтобы продолжить.
- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function disp_alert()`
- `{`
- `alert("I am an alert box!!");`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<input type="button" onclick="disp_alert()" value="Display alert box"`  
`/>`
- `</body>`
- `</html>`

- **Confirm box** : Когда окно подтверждения всплывает, пользователь должен будет нажать либо "ОК" или "Отмена", чтобы продолжить.

- `<html><head>`
- `<script type="text/javascript">`
- `function disp_confirm()`
- `{`
- `var r=confirm("Press a button");`
- `if (r==true)`
- `{ document.write("You pressed OK!"); }`
- `else`
- `{ document.write("You pressed Cancel!"); }`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<input type="button" onclick="disp_confirm()" value="Display a confirm box" />`
- `</body>`
- `</html>`

- **Prompt box**: Контекстное окно часто используется, если вы хотите, чтобы пользователь ввел значение перед входом на страницу.
- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function disp_prompt()`
- `{`
- `var name=prompt("Please enter your name", "Kufliddin Makhmazaitov");`
- `if (name!=null && name!="")`
- `{`
- `document.write("Hello " + name + "! How are you today?");`
- `}`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<input type="button" onclick="disp_prompt()" value="Display a prompt box" />`
- `</body>`
- `</html>`



- Арифметический оператор

+ , - , \* , / , %

- Оператор сравнения

< , > , <= , >= , ==

- Оператор присваивания

- Условный оператор

- `greeting=(visitor=="PRES")?"Dear President ":"Dear ";`

- Логический оператор

- Логические операции обычно используется для объединения нескольких сравнений в условном выражении

- <html>
- <head>
- <script language = "javascript">
- var x = 10;
- alert("the value of x is = " +x); 10
- alert("the value of x+x is = " +(x+x));
- alert("the value of x-x is = " +(x-x));
- alert("the value of x\*x is = " +(x\*x));
- alert("the value of x/x is = " +(x/x));
- alert("the value of x%3 is = " +(x%3));
- alert("the value of --x is = " +(--x));
- alert("the value of ++x is = " ++(++x));
- alert("the value of -x is = " +(-x));
- </script>
- </head>
- </html>

| Оператор               | Использование                   | Описание                                                                                                                                                                                                                                                                  |
|------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Логическое<br>И (&&)   | <i>expr1</i> &&<br><i>expr2</i> | Возвращает <i>expr1</i> , если оно может быть преобразовано к <i>false</i> ; в ином случае возвращает <i>expr2</i> . Так что, при вызове с булевыми значениями, && вернет <i>true</i> только если оба операнда истинны, иначе <i>false</i> .                              |
| Логическое<br>ИЛИ (  ) | <i>expr1</i>   <br><i>expr2</i> | Возвращает <i>expr1</i> , если оно может быть преобразовано к <i>true</i> ; в ином случае возвращает <i>expr2</i> . Так что, при вызове с булевыми значениями,    вернет <i>true</i> только в том случае, когда хотя бы один операнд истинен, иначе вернет <i>false</i> . |
| Логическое<br>НЕ (!)   | ! <i>expr</i>                   | Возвращает <i>false</i> , если операнд может быть преобразован к <i>true</i> , в ином случае возвращает <i>true</i> .                                                                                                                                                     |

```
<HTML>
<HEAD>
 <SCRIPT LANGUAGE="JavaScript">
 var x = 10;
 var y = 5;
 alert ("The value of x is "
 + x + "The value of y is " + y);
 alert("x AND y = " + (x && y));
 alert("x OR y = " + (x || y));
 alert("NOT x = " + (!x));
 </SCRIPT>
</HEAD>
</HTML>
```



- **Если:**

- `<script type="text/javascript">`
- `// Если время меньше 10,`
- `// вы получите приветствие "Доброе утро".`
- `// В противном случае вы получите приветствие "Добрый день".`
- `var d = new Date();`
- `var time = d.getHours();`
- `if (time < 10)`
- `{`
- `document.write("Good morning!");`
- `}`
- `else`
- `{`
- `document.write("Good day!");`
- `}`
- `</script>`

- `<script type="text/javascript">`
- `var d = new Date()`
- `var time = d.getHours()`
- `if (time<10)`
- `{`
- `document.write("<b>Good morning</b>"); }`
- `else if (time>10 && time<16)`
- `{`
- `document.write("<b>Good day</b>");`
- `}`
- `else`
- `{`
- `document.write("<b>Hello World!</b>");`
- `}`
- `</script>`

- Switch :
- <script type="text/javascript">
- {
- var x=2;
- switch (x)
- {
- case 1:
- document.write("hello");
- break;
- case 2:
- document.write("Hai-hello");
- break;
- default:
- document.write("Goob bye!!");
- }
- </script>

# Функции

- Функция содержит код, который будет выполняться по событию или с помощью вызова этой функции
- Функции могут быть определены как в `<head>` и в разделе `<body>` документа. Тем не менее, чтобы убедиться, что функция чтения / загрузки обозревателем, прежде чем она называется, она может быть мудрым, чтобы положить его в разделе `<head>`.



- <html>
- <head>
- <script language = "javascript">
- function hello()
- {
- document.write("calling hello function() <br>");
- return;
- }
- function add(x,y)
- {
- var result = x + y;
- return result;
- }
- function circlearea(r)
- {
- var result = 3.14 \* r \* r;
- return result;
- }
- hello();
- document.write("sum of two number is " +add(12,32)+ "<br>");
- document.write("area of circle is " +circlearea(5));
- </script>
- </head></html>

- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function myfunction()`
- `{`
- `alert("HELLO");`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<form>`
- `<input type="button" onclick="myfunction()" value="Call function">`
- `</form>`
- `<p> При нажатии на кнопку, функция будет вызываться. Функция будет оповещать сообщение.</p>`
- `</body>`
- `</html>`

- Создать JavaScript файл для:
- Передачи параметра в функцию
- Функция возвращает значение
- **Локальная переменная:** когда мы определяем переменную внутри функции, она может быть доступ только в функциях,
- **Глобальная переменная :** когда мы определяем переменную вне функции, она может быть доступна на всех функций

- Цикл:

- <html>
- <body>
- <script type="text/javascript">
- var i=0;
- for (i=0;i<=10;i++)
- {
- document.write("The number is " + i);
- document.write("<br />");
- }
- </script>
- </body>
- </html>

- <html>
- <head>
- <script language = "javascript">
- var x =19,a;
- for(a = 1;a <=10;a++)
- {
- var re = a \* x;
- document.write(x + " \* "+a+" = " +re + "<br>");
- }
- </script>
- </head>
- </html>

- <html>
- <head>
- <script language = "javascript">
- function table(form)
- {
- var x = form.num.value;
- var a;
- for(a = 1;a <=10;a++)
- {
- var re = a \* x;
- document.write(x + " \* "+a+" = " +re + "<br>");
- }
- }
- </script>
- </head>
- <body>
- <form>
- <h2>Enter number : </h2>
- <input type = "text" name = "num" />
- <input type = button value = "display table" onclick = "table(this.form)" />
- </form>
- </body>
- </html>

- Напишите программу, чтобы скопировать содержимое одного текстового поля к другому текстовому полю.
- Напишите программу для отображения площадь круга в текстовом поле. Радиус задается пользователем в соответствующее текстовое поле.
- Напишите программу для преобразования температуры от Фаренгейта к Цельсия

- **While :**
  - while (i<=10)
  - {
  - document.write("The number is " + i);
  - document.write("<br />");
  - i=i+1;
  - }
- **Do-while :**
  - do
  - {
  - document.write("The number is " + i);
  - document.write("<br />");
  - i=i+1;
  - }
  - while (i<0);
- **Break/continue --->**



- Arrays :

- <html>
- <head>
- <script language = "javascript">
- emp = new Array(3);
- emp[0] = "John";
- emp[1] = "TOM";
- emp[2] = "Mike";
- document.writeln(emp[0]+"<br>");
- document.writeln(emp[1]+"<br>");
- document.writeln(emp[2]);
- </script>
- </head>
- </html>
- ..... emp.sort().....

- var x;
- var mycars = new Array();
- mycars[0] = "Volvo";
- mycars[1] = "BMW";
- mycars[2] = "Toyata";
- mycars[5] = "Mercedes";
- for (x in mycars)
- {
- document.write(mycars[x] + "<br />");
- }

# Java script try/catch

- Try ... Catch позволяет проверить блок кода на наличие ошибок. Попытка блок содержит код, который будет работать, а блок улов содержит код, который будет выполнен, если произошла ошибка.
- `try`
- `{`
- `//Run some code here`
- `}`
- `catch(err)`
- `{`
- `//Handle errors here`
- `}`

- **Error demo :**

- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function message()`
- `{`
- `addalert("Welcome guest!");`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<input type="button" value="View message" onclick="message()" />`
- `</body>`
- `</html>`

- Handle the error (use of try/catch block)
- function message()
  - {
  - try
  - {
  - addalert("Welcome guest!");
  - }
  - catch(err)
  - {
  - txt="Ошибке на этой странице.\n\n";
  - txt+=" Описание ошибки : " + err.description + "\n\n";
  - txt+="Click OK to continue.\n\n";
  - alert(txt);
  - }
  - }

# Обработка событий

- Каждый элемент на веб-странице имеет определенные события, которые могут вызвать функции JavaScript. Например, мы можем использовать событие OnClick из элемента кнопки, чтобы указать, что функция будет работать, когда пользователь нажимает на кнопку. Определим события в HTML тегов

# Пример некоторых событий

- **OnClick** событие происходит, когда объект получает нажал
- **ondbclick** событие используется для двойной щелчок

- **Некоторые событие тела :**

- **onresize()** : если вы хотите изменить размер документа, это событие генерируется.

`<body onresize = "func()" >`

- **onload()** : Это событие генерируется, когда мы загружаем или откроем документ в браузере.

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

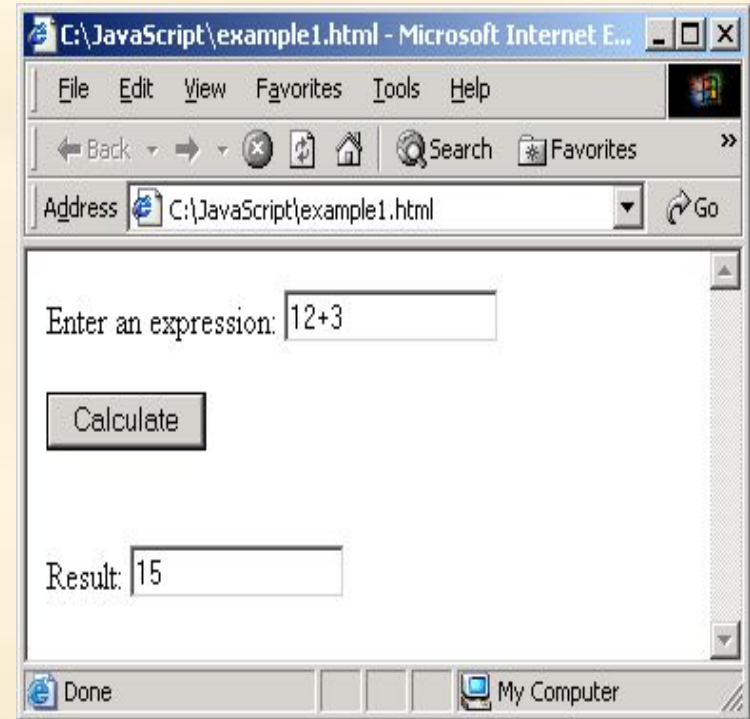
- **onunload()** : Это событие генерируется, когда мы закрываем документ в браузере.

`<body onunload = "func()" >`



- **onselect() Event** : Это событие относится на текстовое поле и области текста, когда пользователь хочет выделить.
- <form>
- <textarea cols="20" rows="5" **onselect="func()"**>
- Hello world! Please select me !!!
- </textarea>
- </form>
- </body>
- </html>
- <input type="text" value="Hello world!"  
**onselect="alert('You have selected some of the text.')**>

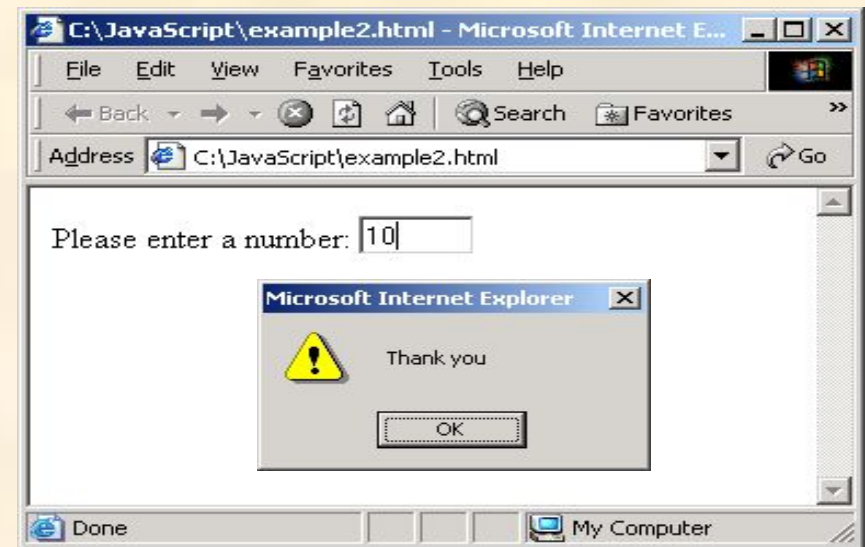
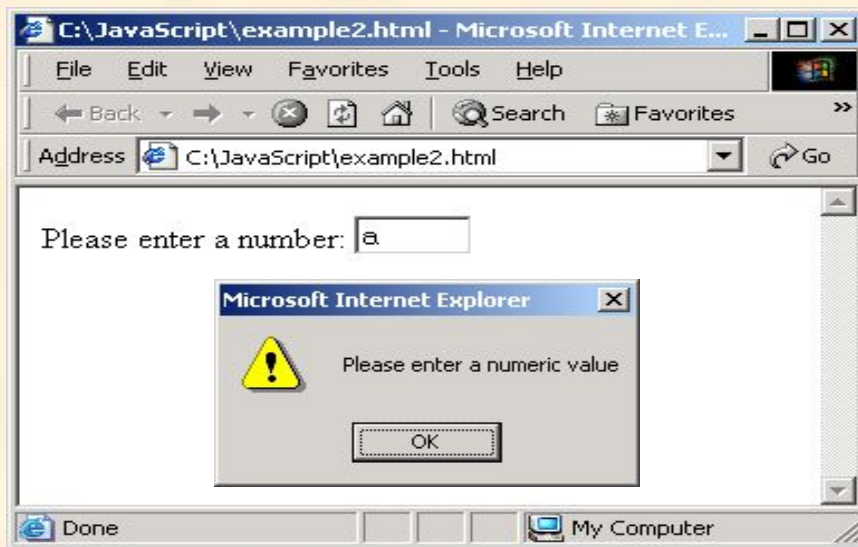
- **<HTML><HEAD>**
- **<SCRIPT LANGUAGE="JavaScript">**
- **function compute(form)**
- **{**
- **if (confirm("Are you sure?"))**
- **form.result.value = eval(form.expr.value)**
- **else**
- **alert("Please come back again.")**
- **}**
- **</SCRIPT> </HEAD>**
- **<BODY>**
- **<FORM>**
- **Enter an expression:**
- **<INPUT TYPE="text" NAME="expr" SIZE=15 ><BR><BR>**
- **<INPUT TYPE="button" VALUE="Calculate"**
- **ONCLICK="compute(this.form)">**
- **<BR><BR><BR>**
- **Result:**
- **<INPUT TYPE="text" NAME="result" SIZE=15 >**
- **</FORM></BODY></HTML>**



- **OnChange событие** : Мероприятие OnChange возникает всякий раз, когда изменяется элемент формы. Это может произойти, когда содержимое текстового элемента управления изменяется, или когда выбор в выборе изменений список.

- <HTML>
- <HEAD>
- <SCRIPT LANGUAGE="JavaScript">
- function checkNum(num)
- {
- if (num == "")
- {
- alert("Please enter a number");
- return false;
- }
- if (isNaN (num))
- {
- alert("Please enter a numeric value");
- return false;
- }
- }

- `else alert ("Thank you");`
- `}`
- `</SCRIPT>`
- `</HEAD>`
- `<BODY bgColor = white>`
- `<FORM>`
- Please enter a number:
- `<INPUT type = text size = 5 onChange="checkNum(this.value)">`
- `</FORM>`
- `</BODY>`
- `</HTML>`



**onFocus:** Мероприятие OnFocus отправляется всякий раз, когда элемент формы становится текущим элемент формы. Только тогда, когда элемент имеет фокус может это принимать ввод от пользователя.

- **onBlur:** Является противоположностью фокусе. Когда пользователь оставляет элемент формы, событие ONBLUR активируется.

- `<HTML><HEAD>`
- `<SCRIPT LANGUAGE="JavaScript">`
- `function fun1() {`
- `document.bgColor = "red";`
- `}`
- `function fun2() {`
- `document.bgColor = "blue";`
- `}`
- `</SCRIPT> </HEAD>`
- `<BODY>`
- `<FORM>`
- `<input type="text" name="t1" onblur="fun1()" onfocus= "fun2()" >`
- `</FORM></BODY></HTML>`
-

- **onMouseOver**

- Мероприятие OnMouseOver генерируется всякий раз, когда курсор мыши перемещается над элементом.

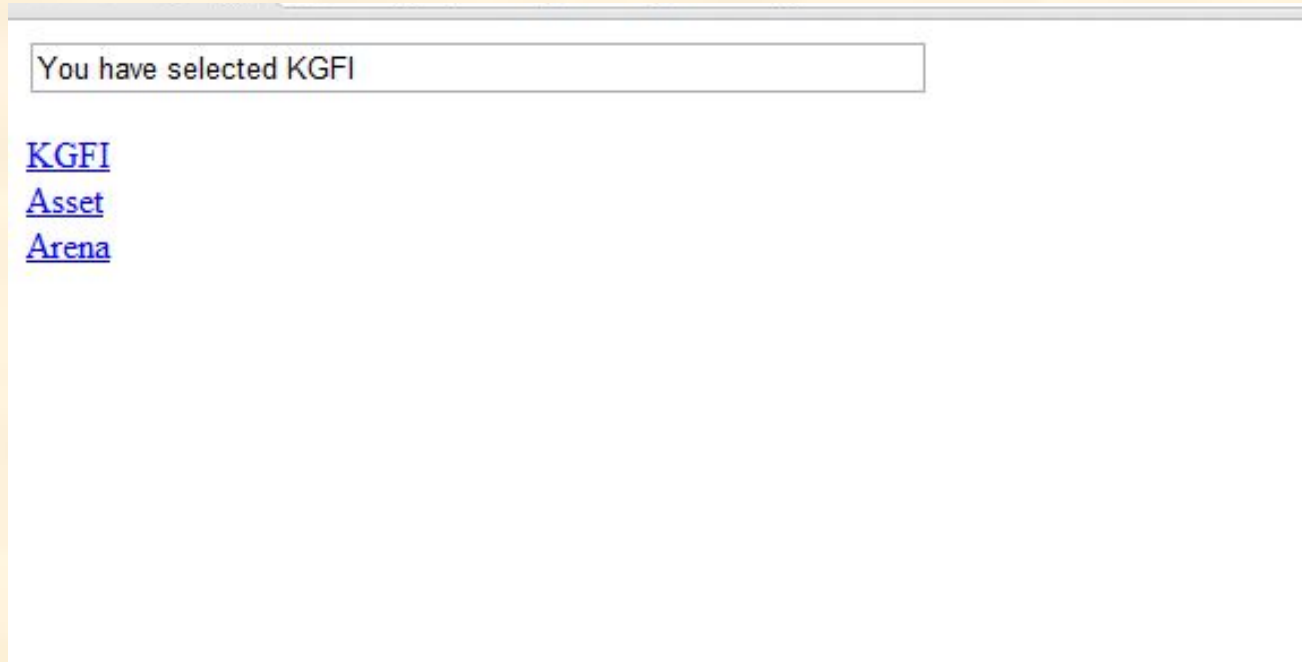
- **onMouseOut**

- Мероприятие onMouseOut генерируется всякий раз, когда курсор мыши перемещается с элемента

- <html>
- <head>
- <script language = "javascript">
- var num =0
- .....  
.....

- function showLink(num)
- {
- if (num==1)
- {
- document.f1.t1.value= "You have selected KGFI";
- }
- if (num==2)
- {
- document.f1.t1.value = "You have selected Asset";
- }
- if (num==3)
- {
- document.f1.t1.value = "You have selected Arena";
- }
- }
- </script>
- </head>

- `<body>`
- `<form name = "f1">`
- `<input type=text name = "t1" size=60 >`
- `</form>`
- `<a href="#" onMouseOver="showLink(1)">KGFI</a><br>`
- `<a href="#" onMouseOver="showLink(2)">Asset</a><br>`
- `<a href="#" onMouseOver="showLink(3)">Arena</a><br>`
- `</body></html>`





- <html>
- <head>
- <script type="text/javascript">
- function mouseOver()
- {
- document.b1.src ="b\_blue.gif";
- }
- function mouseOut()
- {
- document.b1.src ="b\_pink.gif";
- }
- </script>
- </head>
- <body>
- <a href="otherpage.html" >
- 
- </a>
- </body>
- </html>

# Javascript объект

- JavaScript является объектно-ориентированного программирования (ООП). Язык ООП позволяет определить свои собственные объекты и сделать свои собственные типы переменных.
- объект
- свойства
- метод
- **String** метод :
  - `document.write(str.length);`
  - `var str="Hello world!"; document.write(str.toUpperCase());`

- str.big()
- str.small()
- str.italics()
- str.bold()
- str.strike() : Hello World!
- Str.fontcolor("red")
- Str.fontsize(16)
- Substrics : str.sub()
- Superstrics : str.sup()

```

•var str="Hello world!";
•document.write(str.indexOf("Hello") + "
"); // 0
document.write(str.indexOf("World") + "
"); // -1
document.write(str.indexOf("world")); // 6
•document.write(str.match("world")); // world
•document.write(str.match("World")); // null

```

- **Объект Math :**

- Объект Math позволяет выполнять общие математические задачи.
- Объект Math включает в себя несколько математических значения и функции. Вам не нужно определять объект Math перед его использованием.

- **Math.E**
- **Math.PI**
- **Math.SQRT2**
- **Math.SQRT1\_2**
- **Math.LN2**
- **Math.LN10**
- **Math.LOG2E**
- **Math.LOG10E**

- Некоторые часто используемые методы:

**round()**  
**min()**  
**max()**  
**random()**

- Программа для отображения всей информации о браузере:
- `<script type="text/javascript">`
- `var x = navigator;`
- `document.write("CodeName=" + x.appCodeName); // mozilla`
- `document.write("MinorVersion=" + x.appMinorVersion); //SP2`
- `document.write("Name=" + x.appName); // microsoft IE`
- `document.write("Version=" + x.appVersion); //5.0`
- `document.write("CookieEnabled=" + x.cookieEnabled); //true`
- `document.write("CPUClass=" + x.cpuClass); //x86`
- `document.write("Platform=" + x.platform); // Windows`
- `document.write("UA=" + x.userAgent);`
- `document.write("BrowserLanguage=" + x.browserLanguage);`
- `document.write("SystemLanguage=" + x.systemLanguage);`
- `document.write("UserLanguage=" + x.userLanguage);`
- `</script>`

- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function detectBrowser()`
- `{`
- `var browser=navigator.appName;`
- `var b_version=navigator.appVersion;`
- `var version=parseFloat(b_version);`
- `if ((browser=="Netscape"||browser=="Microsoft Internet Explorer") &&`  
`(version>=4))`
- `{`
- `alert("Your browser is good enough!");`
- `}`
- `else`
- `{`
- `alert("It's time to upgrade your browser!");`
- `}`
- `}`
- `</script>`
- `</head>`
- `<body onload="detectBrowser()">`
- `</body>`
- `</html>`

- **объект Окно** : представляет окно браузера и может быть использован для получения информации о состоянии окна

#### СВОЙСТВА :

- `document`
- `event`
- `history`
- `location`
- `name`
- `navigator`
- `screen`

#### МЕТОД :

- `alert`
- `blur`
- `close`
- `focus`
- `navigate`
- `open`

- `<html>`
  - `<head>`
  - `<script language = "javascript">`
  - `document.write("opening new Window!!!!!! ");`
  - `document.write("<br>");`
  - `window.open();`
  - `</script>`
  - `</head>`
  - `</html>`
- 
- `window.open("table.html");`
  - `window.open("http://www.google.com");`



- Написать программу, в которой пользователь введет имя сайта в текстовом поле, нажав кнопку "открыт сайт" сайт будет открываться.
- `<html> <head>`
- `<script language = "javascript">`
- `function call()`
- `{`
- `window.open(fm.site.value);`
- `}`
- `</script>`
- `</head>`
- `<body>`
- `<h3> please Enter the complete URL : For Example <br> http://www.yahoo.com </h3>`
- `<form name = "fm" >`
- `<input type = text name = "site" size = "30" >`
- `<input type = "button" value = "open site" onclick = "call()">`
- `</form></body></html>`

- **History Object** : This object provides a list of the URL's most recently visited by the client  
Example, history's "back()" method cause the window to again display the immediately previous document:  
`history.back();`
- **Location Object** : This object maintains information about the current URL.  
It provides a method that causes the window's current URL to be reloaded.  
`location.replace()....`

# • JavaScript Объект Date

- **Объект Date** используется для работы с датами и временем
- `var myDate=new Date()`
  - Объект Date будет автоматически удерживать текущую дату и время в качестве начального значения
- Манипуляция с датами:
- `var d = new Date();`
- `d.setFullYear(1992,10,3);`
- `document.write(d); // Tue Nov 3 02:06:39 PST 1992`

# JavaScript Timing Events

- С помощью JavaScript можно выполнять некоторые коды не сразу после вызова функции, но по истечении определенного интервала времени. Это называется события времени.
- Два основных метода, которые используются:
- `setTimeout()` - выполняет через некоторое время
  - `clearTimeout()` - отмена `setTimeout()`
- Синтаксис:
- `var t=setTimeout("javascript statement",milliseconds);`
- `clearTimeout(setTimeout_variable)`

- <html>
- <head>
- <script type="text/javascript">
- function timedMsg()
- {
- var t=setTimeout("alert('5 seconds!')",5000);
- }
- </script>
- </head>
- <body>
- <form>
- <input type="button" value="Display timed alertbox!" onClick = "timedMsg()">
- </form>
- <p>Click on the button above. An alert box will be displayed after 5 seconds.</p>
- </body>
- </html>

- <html>
- <head>
- <script type="text/javascript">
- var c=0;
- var t;
- function timedCount()
- {
- document.getElementById('txt').value=c;
- c=c+1;
- t=setTimeout("timedCount()",1000);
- }
- </script>
- </head>
- <body>
- <form>
- <input type="button" value="Start count!" onClick="timedCount()">
- <input type="text" id="txt">
- </form>
- <p>Click on the button above. The input field will count for ever, starting at 0.</p>
- </body>

- **Показать часы:**

- `<html>`
- `<head>`
- `<script type="text/javascript">`
- `function startTime()`
- `{`
- `var today=new Date();`
- `var h=today.getHours();`
- `var m=today.getMinutes();`
- `var s=today.getSeconds();`
- `// add a zero in front of numbers<10`
- `m=checkTime(m);`
- `s=checkTime(s);`
- `document.getElementById('txt').innerHTML=h+":"+m+":"+s;`
- `t=setTimeout('startTime()',500);`
- `}`

- function checkTime(i)
- {
- if (i<10)
- {
- i="0" + i;
- }
- return i;
- }
- </script>
- </head>
  
- <body onload="startTime()">
- <div id="txt"></div>
- </body>
- </html>



# Handling Form Events

- Write a programme to compute addition, subtraction , multiplication and division by clicking corresponding buttons. Result will display in text field
- Write a programme to display name , age , address and phone number of user in pop-up window. Details will be given by the user.

- <html>
- <head>
- <script language = "javascript">
- function white()
- {
- document.bgColor = "white";
- }
- function black()
- {
- document.bgColor = "black";
- }
- </script>
- </head>
- <body>
- <form>
- Light On <input type = radio name = "r" value = "r1" onclick = "white()" >
- Light Off <input type = radio name = "r" value = "r2" onclick = "black()" >
- </form>
- </body>
- </html>

# Проверка с помощью JavaScript

JavaScript может быть использован для проверки входных данных в HTML форм перед отправкой содержание к серверу.

Данные формы, которые обычно проверяются в JavaScript могут быть:

- пользователь оставил необходимые поля пустыми?
- пользователь ввел действительный адрес электронной почты?
- пользователь ввел корректную дату?
- пользователь ввел текст в цифровой области?

- <html>
- <head>
- <script type="text/javascript">
- function validate\_form()
- {
- if (f1.t1.value == "")
- {
- alert ( "Please fill the login id" );
- return false;
- }
- if (f1.t2.value == "" )
- {
- alert ( "Please fill the password" );
- return false;
- }
- return true;
- }
- </script>

- </head>
- <body>
- <form name="f1" action="abc.jsp"
- onsubmit="return validate\_form ( );">
- Login ID : <input type=text name=t1 /><br>
- Password : <input type=password name=t2 /><br>
- <input type="submit" />
- </body>
- </html>

– If(f1.t1.value.indexOf("@") == -1) // for validating the e-mail id

- <html>
- <head>
- <script language = "javascript" >
- function validfn(frm)
- {
- var str = frm.fn.value;
- if(str.length == 0)
- {
- alert("first name field is empty ");
- } }
- function validln(frm)
- {
- var str = frm.ln.value;
- if(str.length == 0)
- {
- alert("Last name field is empty ");
- } }

- function validem(frm)
- {
- var str = frm.em.value;
- if(str.length == 0)
- {
- alert("E-mail id field is empty");
- }
- }
- function validco(frm)
- {
- var str = frm.comm.value;
- if(str.length == 0)
- {
- alert("comment field should not empty is empty");
- }
- }

- function process(frm)
- {
- var fname = frm.fn.value;
- var lname = frm.ln.value;
- var email = frm.em.value;
- var com = frm.comm.value;
- disp = open("", "result");
- disp.document.write("<center><h2>Thanks for signing</h2></center><hr>");
- disp.document.write("First Name : " +fname+ "<br>");
- disp.document.write("Last Name : " +lname+ "<br>");
- disp.document.write("E-mail ID : " +email+ "<br>");
- disp.document.write("your comments : " +com+ "<br>");
- if(disp.confirm("is this info correct"))
- disp.close();
- }
- </script>
- </head>

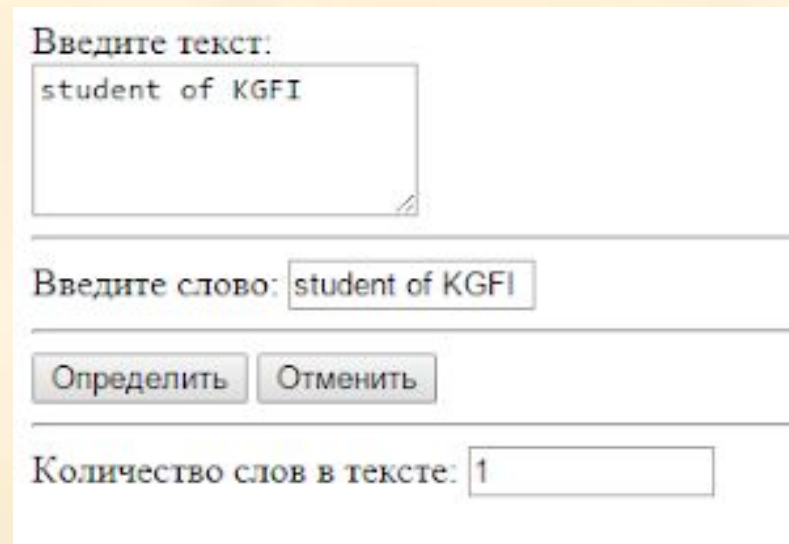


- <body>
- <form>
- <center><h2> Application Form </h2></center><hr>
- Enter your first name : <input type = text name = "fn" size = 20 onblur = "validfn(this.form)"><br><br>
- Enter your Last name : <input type = text name = "ln" size = 20 onblur = "validln(this.form)" ><br><br>
- Enter your E-mail ID : <input type = text name = "em" size = 20 onblur = "validem(this.form)"><br><br>
- Please enter your comments : <textarea name = "comm" rows = "5" cols = "30" onblur = "validco(this.form)"></textarea><br><br>
- <input type = "button" value = "submit form " onclick = "process(this.form)" >
- </form>
- </body>
- </html>

# JavaScript работа со строками

```
<html>
<head>
 <title>javascript строки</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <script type="text/javascript" src="script.js"></script>
</head>
<body>
 <form name="forma11">
 Введите текст:

 <textarea name="textin" rows="4" cols="20"></textarea><hr>
 Введите слово: <input type="text" size="10" name="slovo"><hr>
 <input type="button" value="Определить" onClick="numword(forma11);">
 <input type="reset" value="Отменить"><hr>
 Количество слов в тексте: <input type="text" size="10" name="res">
 </form>
</body>
</html>
```



Введите текст:

student of KGFI

---

Введите слово: student of KGFI

Определить Отменить

---

Количество слов в тексте: 1

```
function numword(obj) {
 var t=obj.textin.value;
 var s=obj.slovo.value;
 var m=s.length;
 var res=0;
 var i=0;
 while (i < t.length-1)
 {var ch=t.substr(i,m)
 if (ch==s){
 res+=1;
 i=i+m
 }
 else
 i++
 }
 obj.res.value=res
}
```

# Работа с окнами

```
<html> <head>
 <title>javascript окно</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <script type="text/javascript" src="script.js"></script>
</head> <body>
 <form name="forma14">
 <input type="button" value="Открыть окно" onClick="choiceOf();">
 </form> </body></html>
```

```
<html> <head>
 <title>javascript окно</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <script type="text/javascript" src="script.js"></script>
</head> <body>

 <input type="button" value="Закрыть" onClick="close_pict();">
</body></html>
```

```
function choiceOf(){
 if (confirm("Вы действительно хотите открыть окно?")) {
 volvo=window.open("volvo.html", "display_volvo",
 "width=400,height=300,status=no,toolbar=no,menubar=no");
 }
}
function close_pict() {
 window.close();
}
```

# Основы анимации

С точки зрения HTML/CSS, анимация — это постепенное изменение стиля DOM-элемента. Например, увеличение координаты `style.left` от 0px до 100px сдвигает элемент.

Код, который производит изменение, вызывается таймером. Интервал таймера очень мал и поэтому анимация выглядит плавной. Это тот же принцип, что и в кино: для непрерывной анимации достаточно 24 или больше вызовов таймера в секунду.

Псевдо-код для анимации выглядит так:

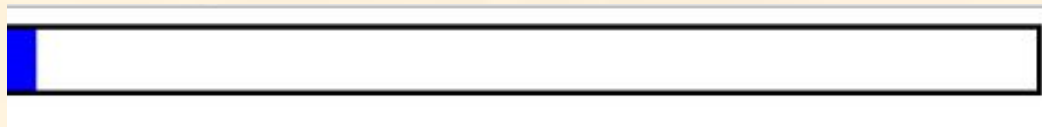
```
var timer = setInterval(function() {
 показать новый кадр
 if (время вышло)
 clearInterval(timer);
}, 10)
```

Задержка между кадрами в данном случае составляет 10 ms, что означает 100 кадров в секунду.

В большинстве фреймворков, задержка по умолчанию составляет 10-15 мс. Меньшая задержка делает анимацию более плавной, но только в том случае, если браузер достаточно быстр, чтобы анимировать каждый шаг вовремя.

# Пример

```
<script>function move(elem) {
 var left = 0; // начальное значение
 function frame() { // функция для отрисовки
 left++; elem.style.left = left + 'px'
 if (left == 100) {
 clearInterval(timer); // завершить анимацию
 }
 }
 var timer = setInterval(frame, 10) // рисовать каждые 10мс
}
</script>
<div onclick="move(this.children[0])" class="example_path"> <div
class="example_block"></div></div>
```



# Структура анимации

У анимации есть три основных параметра:

## **delay**

Время между кадрами (в миллисекундах, т.е. 1/1000 секунды).  
Например, 10мс.

## **duration**

Общее время, которое должна длиться анимация, в мс.  
Например, 1000мс.

## **step(progress)**

Функция **step(progress)** занимается отрисовкой состояния анимации, соответствующего времени progress. Каждый кадр выполняется, сколько времени прошло:  $progress = (now - start) / duration$ .  
Значение progress меняется от 0 в начале анимации до 1 в конце. Так как вычисления с дробными числами не всегда точны, то в конце оно может быть даже немного больше 1. В этом случае мы уменьшаем его до 1 и завершаем анимацию.



```
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<link type="text/css" rel="stylesheet" href=".animate.css">
<script src="width.js"></script>
</head>
<body>
<div onclick="stretch(this.children[0])" class="example_path">
<div class="example_block" style="width: 5%;">
</div></div>
</body></html>
```

```
function animate(opts) {
 var start = new Date; // сохранить время начала
 var timer = setInterval(function() { // вычислить сколько времени прошло
 var progress = (new Date - start) / opts.duration;
 if (progress > 1) progress = 1; // отрисовать анимацию
 opts.step(progress);
 if (progress == 1) clearInterval(timer); // конец
 }, opts.delay || 10);
}
```

```
function stretch(elem) {
 animate({
 duration: 1000, step: function(progress) {
 elem.style.width = progress*100 + "%";
 } });
}
```

Возьмите решение задачи [Карусель](#) и сделайте передвижение карусели плавным, анимированным.

Подключите для этого файл [animate.js](#).

Результат:



**Спасибо !!!!!**