

Лекция # 4



ОСНОВЫ ИСПОЛЬЗОВАНИЯ JavaScript

Function



```
<script language="JavaScript">  
  // Функция для отсылки формы на сервер  
  function SendForm()  
  {  
    if( Check()) window.document.forms[0].submit;  
  }  
  // Функция для проверки содержимого формы  
  function Check()  
  {  
    var doc = window.document;
```

```
if( doc.forms[0].elements[0].value == " ||  
    doc.forms[0].elements[1].value == " ||  
    doc.forms[0].elements[2].value == ")  
{  
    alert( 'Поля не могут быть пустыми');  
    return false;  
}  
else  
    return true;  
}  
</script>
```

```
<form name="rqform" action="request.dll/doform"
method="post">
```

```
<font size=4 color="#000090">
```

```
<b>Пожалуйста, введите информацию:</b>
```

```
<br><br>
```

```
</font>
```

```
<table bgcolor="#fff8bd" border=0 cellpadding=0
cellpadding=1>
```

```
<tr>
```

```
<td><b>&nbsp;Фамилия</b></td>
```

```
<td><input type="text" size=35 value=""
name="famname">
```



```
< <td><TEXTAREA name="contact" rows="5"
cols="30"></TEXTAREA></td>
</tr>
<tr>
  <td width="50%" align="left">
    <input type="Submit" value="Послать"
onClick="SendForm()"></td>
  <td width=503 align right>
    <input type="Reset" value="Отменить"
onClick="history.back(-1)"></td>
</tr>
</table>
</form>
```

If..else



если результат выражения имеет значение true, выполняются инструкции, расположенные в блоке if;

если результат выражения имеет значение false, выполняются инструкции, расположенные в блоке else. Если блок else не указан, выполнение программы переходит на следующий элемент после всего блока if..else.

```
if( Check())  
    Send();  
else  
    Clear();
```



```
if( Check())  
{  
    alert('Отсылаем форму');  
    Send();  
}  
else  
{  
    alert('Очищаем форму');  
    Clear();  
}
```

```
function Check()  
{  
  var doc = window.document;  
  if( doc.forms[0].elements[0].value == "")  
  {  
    alert('Поля не могут быть пустыми');  
    return false;  
  }  
  else  
    return true;  
}
```

New



- для создания нового объекта Date, который является встроенным объектом языка JavaScript;
- для создания нового пользовательского объекта.

`varName = new objectName(params)`

- `varName` - имя переменной, в которой будет храниться новая копия объекта;
- `objectName` - имя объекта. При использовании встроенного объекта `Date` используется слово `Date` (с большой буквы!), при использовании пользовательских объектов - имена объектов;
- `params` - один или более параметров, передаваемых при создании копии объекта.

```
user = new someUser("Alex Fedorov");  
alert( user.name);  
function someUser(nameParam) {  
  this.name = nameParam;  
  return (this);  
}
```

Return



```
function Check()  
{  
  var doc = window.document;  
  if( doc.forms[0].elements[0].value == "")  
  {  
    alert('Поля не могут быть пустыми');  
    return false;  
  }  
  else  
    return true;  
}
```

This



- для ссылки на текущую форму или интерфейсный элемент в обработчике события (например, onClick или onSubmit);
- для задания нового свойства в пользовательском объекте.

```
<INPUT TYPE="button" NAME="test" VALUE="Test"  
onClick="test(this.form)"
```

```
<INPUT TYPE="button" NAME="test" VALUE="Test"  
onClick="test(this)">
```



```
<html>
<head><title>JS - CP1197</title>
<script language="JavaScript">
  function myTest(obj)
  {
    alert(obj.name);
    return;
  }
</script>
</head>
<body>
<p align="center">
```

Пример использования ключевого слова `this`

```
</p>
<center>
<form name="ThisTest">
<INPUT TYPE="button" NAME="test1" VALUE="Test1"
  onClick="myTest(this.form)">
<INPUT TYPE="button" NAME="test2" VALUE="Test2"
  onClick="myTest(this)">
</form>
</center>
</body>
</html>
```

Var



```
var VariableName;
```

```
var VariableName = value;
```

- VariableName - имя переменной;
- value - значение, присваиваемое переменной.

```
var someVar = 100;  
function showVar()  
{  
    alert( someVar);  
}
```

```
var someVar = 100;
function showLocalVar()
{
    var someVar = 256;
    alert( 'local var = ' + someVar);
    showGlobalVar();
}
function showGlobalVar()
{
    alert( 'global var = ' + someVar);
}
```

While



```
loopCount = 0;
while( loopCount < 10)
{
    document.write( "LoopCount = " + loopCount +
"<BR>");
    loopCount++;
}
```

With



```
function Show()  
{  
  with( document.forms[0].elements[0])  
  {  
    alert( name);  
    alert( value);  
  }  
}
```

Объект Array



```
<html>
<head><title>JavaScript. 12-97</title>
<script language="JavaScript">
// создание нового массива
  allStr = new Array(5);
// заполнение массива
  allStr[0] = "Message #1";
  allStr[1] = "Message #2";
  allStr[2] = "Message #3";
  allStr[3] = "Message #4";
  allStr[4] = "Message #5";
```



```
// функция для отображения элемента массива
```

```
function showMsg(ndx)
```

```
{
```

```
    alert(allStr[ndx]);
```

```
}
```

```
</script>
```

```
</head>
```

```
<!-- При загрузке документа показать сообщение N4 -->
```

```
<body onLoad="showMsg(3);">
```

```
</body>
```

```
</html>
```

```
<html>
<head><title>JavaScript. 12-97</title>
<script language="JavaScript">
// создание нового массива и его заполнение
  allStr = new Array("Message #1", "Message #2", "Message
#3",
  "Message #4", "Message #5");
// функция для отображения элемента массива
function showMsg(ndx)
{
  alert(allStr[ndx]);
}
</script>
```

```
</head>
```

```
<!-- При загрузке документа показать сообщение N4  
-->
```

```
<body onLoad="showMsg(3);">
```

```
</body>
```

```
</html>
```

Методы объекта Array



Метод	Описание
join	Объединяет все элементы массива в одну строку
reverse	Изменяет порядок элементов в массиве - первый элемент становится последним, последний - первым
Sort	Выполняет сортировку элементов массива

```
<html>
<head><title>JavaScript. 12-97</title>
<script language="JavaScript">
  myArray = new Array("Mother", "Father", "Sister", "Brother",
    "Uncle");
function showElement(ndx)
{
  alert(myArray[ndx]);
}
function showAll()
{
```

```
for( i = 0; i <= myArray.length-1; i++)  
  {  
    showElement(i);  
  }  
}  
</script>  
</head>  
<body onLoad="showAll();">  
</body>  
</html>
```

```
function test()  
{  
  alert(myArray.join());  
}
```

```
<body onLoad="test();">
```

```
function test()  
{  
  alert(myArray.join(" _|_ "));  
}
```



```
function test()  
{  
  myArray.reverse();  
  alert(myArray.join(";"));  
}
```

```
function test()  
{  
  myArray.sort();  
  alert(myArray.join(";"));  
}
```

Создание многомерных массивов



```
<html>
<head><title>JavaScript. 12-97</title></head>
<body>
<center>
<font size=5><b>Multidimensional Array</b></font><p>
<script language="JavaScript">
  a = new Array(4);
  for( i=0; i < 4; i++)
  {
    a[i] = new Array(4);
    for( j=0; j < 4; j++)
```

```
{  
    a[i][j] = "["+i+", "+j+"]";  
}  
}  
for( i=0; i < 4; i++)  
{  
    str = "Row "+i+":";  
    for( j=0; j < 4; j++)
```

```
{  
    str += a[i][j];  
}  
document.write( str, "<br>");  
}  
</script>  
</center>  
</body>  
</html>
```

Объект Boolean



```
bfalse = new Boolean(false);
```

```
btrue = new Boolean(true);
```

```
<html>
<head><title>JavaScript 12.97</title></head>
<body>
<script language="JavaScript">
  // создадим две булевых переменных
  bfalse = new Boolean(false);
  btrue = new Boolean(true);
  // выведем их значения (булевы значения)
  document.write(bfalse.valueOf()+"<br>");
  document.write(btrue.valueOf()+"<br>");
```

```
// выведем строчные значения
document.write(bfalse.toString()+"<br>");
document.write(btrue.toString()+"<br>");
</script>
</body>
</html>
```


Объект Date



`MyDate = new Date([параметры]);`

- без параметров - экземпляр будет содержать текущую дату и время. Например, `today = new Date();`
- строка, представляющая собой дату в следующем формате: "Месяц день, год часы:минуты:секунды". Например, `someDate = new Date("May 15, 1996")`. Если число часов, минут или секунд не указано, их значения равны 0;
- набор целочисленных значений для года, месяца и дня. Например, `otherDay = new Date(96, 4, 15);`
- набор целочисленных значений для года, месяца, дня, часов, минут и секунд. Например, `sameDay = new Date(96, 4, 15, 15, 30, 0);`

```
<html>
<head><title>JavaScript 12.97</title></head>
<body>
<center>
<script language="JavaScript">
today = new Date();
document.write("today="+today+"<br>");
someDate = new Date("May 16, 1996");
document.write("someDate="+someDate+"<br>");
```

```
otherDay = new Date( 96, 4, 15);  
document.write("otherDay="+otherDay+"<br>");  
sameDay = new Date( 96, 4, 16, 15, 30, 0);  
document.write("sameDay="+sameDay+"<br>");  
</script>  
</center>  
</body>  
</html>
```

Методы объекта Date



- *методы установки (set)* - методы для установки даты и времени у экземпляров объекта Date;
- *методы определения (get)* - методы для определения даты и времени у экземпляров объекта Date;
- *методы преобразования (to)* - методы для преобразования даты и времени в строки;
- *методы для обработки даты.*

Значения объекта Date



Значение	Диапазон
Число секунд и минут	0..59
Число часов	0..23
День недели	0..6
Дата	1..31
Месяц	0..11 (Январь..Декабрь)
Год	Число лет с 1900

```
<html>
<head><title>JavaScript 12.97</title></head>
<body>
<center>
<p>
<script language="JavaScript">
  someDate = new Date( "May 15, 1996");
  document.write("someDate="+someDate+"<br>");
  document.write("getDay
=" +someDate.getDay()+"<br>");
```

```
document.write("getMonth="+someDate.getMonth()+  
"<br>");  
    document.write("getYear  
="+someDate.getFullYear()+"<br>");  
</script>  
</center>  
</body>  
</html>
```

```
<html>
<head><title>JavaScript 12.97</title></head>
<body>
<center>
<br><br><br>
<script language="JavaScript">
  today = new Date();
  // задать дату
  endYear = new Date("December 31, 1990");
  // поменять год
  endYear.setYear(today.getYear());
  // вычислить число миллисекунд в дне
```



```
msPerDay = 24 * 60 * 60 * 1000;  
// получить число дней  
daysLeft = (endYear.getTime() - today.getTime()) / msPerDay;  
// округлить  
daysLeft = Math.round(daysLeft);  
// показать  
document.write("Number of days left in the year: "+daysLeft);  
</script>  
</center>  
</body>  
</html>
```