



S. S Jain Subodh P.G. (Autonomous) College

SUBJECT - JavaScript

TITLE - Basic Concepts

JavaScript - Basic Concepts

Prepared and Presented by
Dr. Yogendar Kumar Verma

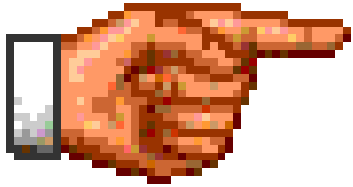


Table of Content

- Brief Introduction
- Embedding JavaScript in HTML
- Statements/Expressions
- Functions - Defining - Calling
- Event Handlers



Introduction

- JavaScript is Netscape's cross-platform, object-based scripting language for client and server applications
- JavaScript is not Java. They are similar in some ways but fundamentally different in others.



JavaScript and Java

- The JavaScript resembles Java but does not have Java's static typing and strong type checking.
- JavaScript supports most Java expression syntax and basic control-flow constructs.
- JavaScript has a simple, instance-based object model that still provides significant capabilities.



JavaScript Types

There're 2 types:

- * Navigator's JavaScript, also called client-side JavaScript
- * LiveWire JavaScript, also called server-side JavaScript



Embedding JavaScript in HTML

- By using the SCRIPT tag
- By specifying a file of JavaScript code
- By specifying a JavaScript expression as the value for an HTML attribute
- By using event handlers within certain other HTML tags



SCRIPT Tag

The `<SCRIPT>` tag is an extension to HTML that can enclose any number of JavaScript statements as shown here:

```
<SCRIPT>  
    JavaScript statements...  
</SCRIPT>
```

A document can have multiple SCRIPT tags, and each can enclose any number of JavaScript statements.



Hiding scripts in comment tags

```
<SCRIPT>
```

```
<!-- Begin to hide script contents from old  
browsers.
```

```
JavaScript statements...
```

```
// End the hiding here. -->
```

```
</SCRIPT>
```




Famous “Hello World” Program

```
<html>  
<body>  
  <script language="JavaScript">  
    document.write("Hello, World!")  
  </script>  
</body>  
</html>
```



JavaScript code in a file

- The SRC attribute of the <SCRIPT> tag lets you specify a file as the JavaScript source (rather than embedding the JavaScript in the HTML).
- This attribute is especially useful for sharing functions among many different pages.



Example

```
<HEAD>
```

```
<TITLE>My Page</TITLE>
```

```
<SCRIPT SRC="common.js">
```

```
.....
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
```

```
.....
```



Statements

- Conditional Statement: if...else

```
if (condition) {  
    statements1  
} else {  
    statements2  
}
```



Loop Statements

- for statement:

```
for ([initial-expression]; [condition]; [increment-expression]) {  
  statements  
}
```

- while statement:

```
while (condition) {  
  statements  
}
```



Expressions

- An *expression* is any valid set of literals, variables, operators, and expressions that evaluates to a single value; the value can be a number, a string, or a logical value.



Expressions (cont'd)

- JavaScript has the following types of expressions:
 - * Arithmetic: evaluates to a number, for example 3.14159
 - * String: evaluates to a character string, for example, "Fred" or "234"
 - * Logical: evaluates to true or false



Data type conversion

- JavaScript is a loosely typed language. That means you do not have to specify the data type of a variable when you declare it, and data types are converted automatically as needed during script execution. So, for example, you could define a variable as follows:

```
var answer = 42
```
- And later, you could assign the same variable a string value, for example,

```
answer = "Thank you"
```




Data type conversion (cont'd)

In expressions involving numeric and string values, JavaScript converts the numeric values to strings. For example, consider the following statements:

```
x = "The answer is " + 42
```

```
y = 42 + " is the answer."
```



Defining and calling Functions

- Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure--a set of statements that performs a specific task. A function definition has these basic parts:
 - * The **function** keyword.
 - * A function name.
 - * A comma-separated list of arguments to the function in parentheses.
 - * The statements in the function in curly braces.



Functions example

```
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    function square(number) {
      return number * number
    }
  </SCRIPT>
</HEAD>
<BODY>
  <SCRIPT>
    document.write("The function returned ", square(5), ".")
  </SCRIPT>
  <P> All done.
</BODY>
```



Event handlers

JavaScript applications in the Navigator are largely event-driven. *Events* are actions that occur usually as a result of something the user does. For example, clicking a button is an event, as is changing a text field or moving the mouse over a hyperlink. You can define *event handlers*, such as **onChange** and **onClick**, to make your script react to events.



Event Handler (cont'd)

Here're a few event handler function

onAbort: user aborts the loading

onClick: user clicks on the link

onChange: user changes value of an element

onFocus: user gives input focus to window

onLoad: user loads page in Navigator



An example of event handler

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



Thank you.....

