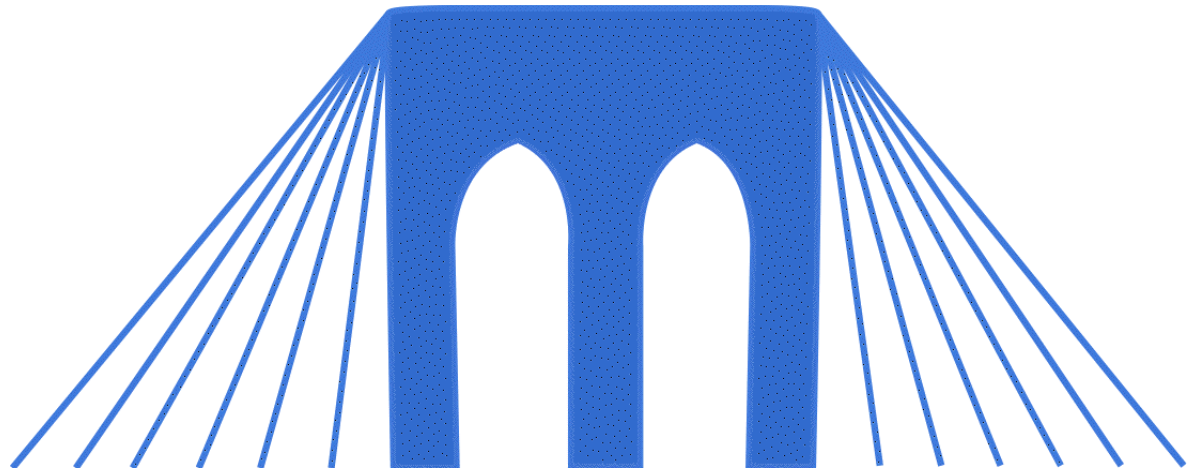
Bridges To Computing



General Information:

- This document was created for use in the "Bridges to Computing" project of Brooklyn College.
- You are invited and encouraged to use this presentation to promote computer science education in the U.S. and around the world.
- For more information about the Bridges Program, please visit our website at: http://bridges.brooklyn.cuny.edu/

Disclaimers:

- All images in this presentation were created by our Bridges to Computing staff or were found online through open access media sites and are used under the Creative Commons Attribution-Share Alike 3.0 License.
- If you believe an image in this presentation is in fact copyrighted material, never intended for creative commons use, please contact us at http://bridges.brooklyn.cuny.edu/ so that we can remove it from this presentation.
- This document may include links to sites and documents outside of the "Bridges to Computing" domain. The Bridges Program cannot be held responsible for the content of 3rd party sources and sites.

JavaScript

An Introduction

Contents

What is JavaScript?

1. JavaScript is NOT Java!
2. JavaScript was designed to add interactivity to HTML pages
3. JavaScript is a scripting language
4. JavaScript is usually embedded directly into HTML pages
5. JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
6. JavaScript is free and open-source

What can a JavaScript do?

1. JavaScript gives HTML designers a programming tool (very simple syntax)
2. JavaScript can put dynamic text into an HTML page
3. JavaScript can react to events
4. JavaScript can read and write HTML elements
5. JavaScript can be used to validate data (validate form data)
6. JavaScript can be used to detect the visitor's browser
7. JavaScript can be used to create cookies

DOM (Document Object Model)

- In object-oriented programming we look at a program as a collection of interacting objects.
  - Objects have properties (facts about them)
    - Length, width,
  - Objects have functions (things they can do)
    - write(), shoot()
- In JavaScript the document is considered to be an object.

"dot" syntax

- Using Javascript we can access the properties of a document and change a document using special functions.
- When you see a dot (.) that implies ownership
- Examples:
  - document.title <!-- the title of the document -->
  - document.images[1] <!-- the second image in page -->
  - document.write("Hi!") <!-- function that writes text -->

First JavaScript

```
<html>
   <head>
   </head>
   <body>
      <script type="text/javascript">
         <!--
            document.write("<h1>Hello World!</h1>");
         //-->
      /script>
   </body>
</html>

<!-- Note use of document object model -->
<!-- See scripts/1_FirstJavaScript.html -->
```

events & event-listeners

- Javascript allows for interactivity.
- Specifically it tracks events:
  - A mouse click
  - A web page or an image loading
  - Mousing over a hot spot on the web page
  - Submitting an HTML form
  - A keystroke
- And allows code to be attached to event-listeners:
  - onLoad
  - onUnload
  - onMouseOver
  - onMouseOut
  - onSubmit

functions

- As you may have noticed event-listeners are a special type of built in functions.
- You can also create your own functions using the "function" keyword.
- All JavaScript code (like functions) should be put inside <script> and </script> tags.
- It is best to put as much of your JavaScript code as possible, in the head section of your document.

Body vs. Head

```
<html>

    <head>
        <script type="text/javascript">
            function message()
            {
                alert("This alert box called by onload event");
            }
        </script>
    </head>

    <body onload="message()">
        <script type="text/javascript">
            document.write("Message written by JavaScript");
        </script>
    </body>

</html>

<!-- See 2_EventListeners-->
```

General Syntax

- Scripts need to be in <script> tags.
- JavaScript statements 'should' end in a semicolon.
- Large blocks of javascript 'should' be put in { }
- Comments can be // or <*

```
<script type="text/javascript">
{
    document.write("<h1>This is a heading</h1>");
    document.write("<p>This is a paragraph.</p>");
    document.write("<p>This is another paragraph.</p>");
    // Single line comment
    /*
        Multi-line
        comment
    */
}
</script>
```

Declaring (Creating) Variables

- Variables are placeholders. You can create (or 'declare') them in JavaScript by using the "var" keyword.
  var x;
  var carname;

- You can also assign values to the variables when you declare them:
  var x=5;
  var carname="Volvo"; // Text requires quotes

- If you assign values to variables that have not yet been declared, the variables will automatically be declared.
  x=5;
  carname="Volvo";

- See 3_Var_PromptBox

Math && Logic

JavaScript supports the following operators

Mathematical:
+, -, *, /, %, ++, --

Logical:
&&, ||, !, <, >, etc...

```
var x=10;
var y=2;
var z=x/y;
```

Strings can be concatenated with the '+' sign.

```
var txt1="What a very";
var txt2="nice day";
var txt3=txt1+" "+txt2;
```

Note: If you add a number and a string, the result will be a string!

Selection (the 'if' statement)

```
<script type="text/javascript">
        // Comments are VERY VERY important
        //  If the time < 10, "Good morning" greeting.
        //  Otherwise you will get a "Good day" greeting.

    var d = new Date();
    var time = d.getHours();

    if (time < 10){
        document.write("Good morning!");
    } else {
        document.write("Good day!");
    }
</script>
```

See file 4_Selection

Repetition (loops)

```
<script type="text/javascript">
    // This script shows the two common types of loops.
    // The for loop
    for (var counter = 1; counter <= 5; ++counter)
    {
                    document.write(counter + " <br />");
    }
    // The while loop
    var i = true
    while (i == true) {
          i = window.confirm("Redisplay this dialog box?");
    }
</script>
```

See file 5_Repetition

Popup boxes

- As you have seen some of JavaScripts interactive abilities come from the use of pop-up boxes.
- There are 3 general kinds:

```
alert("sometext");
confirm("sometext");
   // returns "true" or "false")
prompt("sometext","defaultvalue");
   //returns what user enters as answer
```

See 6_PopUps

Changing Page Properties

- Remember the DOM of JavaScript allows you to change the properties of a page… like the CSS sheet.

```
<head>
    <link rel="stylesheet" type="text/css" href="default.css" id="css_sheet"/>
    <script type="text/javascript" language="javascript">
        function swapCss()    {
            if(document.getElementById('css_sheet').href == 'default.css')    {
                document.getElementById('css_sheet').href = 'alt.css';
            }  else  {
                document.getElementById('css_sheet').href = 'default.css';
            }
        }
</script>
</head>
```

How Powerful is JavaScript?

- It is possible to do simple (and complex) animations (and even games) in JavaScript.
- You can create vector images in JavaScript (you will need to download a special library).
- You can load and swap bitmap (or jpeg, gif, etc.) images natively.
- Timing the image swaps can be done using the built in setTimeout method:

```
t=setTimeout("javascript statement",milliseconds);
```

Examples:

- SimpleAnimation1.html
- SimpleAnimation2.html
- SimpleAnimation3.html
- SimpleAnimation4.html

Getting Help

- W3C JavaScript Tutorial:
- http://www.w3schools.com/js/default.asp

- JavaScript Animations:
- http://www.schillmania.com/content/projects/javascript-animation-1/

- JavaScript Games:
- http://www.devx.com/webdev/10MinuteSolution/27134