



AP[®] Computer Science

Peeking at Programming with JavaScript

Teacher Guide

Martin Callaghan
John Leggott College
Scunthorpe, UK

connect to college success[™]
www.collegeboard.com

Introduction

Thinking where to start with teaching programming is always a difficult issue. Deciding which programming language to use is even more difficult.

This approach uses JavaScript. You might think “Why JavaScript? Why not something a bit more mainstream like Visual Basic or modern, like Java?”.

There are a number of good reasons for using JavaScript. Firstly, it is “free” in that it is built into modern Web browsers. If you have Internet Explorer, Firefox or Netscape Navigator installed then you have all you need to run JavaScript. Secondly, because we are not going to be using an overly complex IDE (like Visual Basic has), students won’t be too distracted with forms and widgets and will be able to concentrate on what their code actually does.

Because JavaScript is built into Web pages, it will allow you to introduce programming concepts through a scenario that many students are already familiar with. It seems that these days most kids have a basic idea of what HTML is all about, so you will be able to “extend” this knowledge with a bit of JavaScript.

This sequence of five lab exercises takes students through the basic ideas of “sequence”, “selection” and “iteration” and finishes off with two more complex exercises that bring together ideas about events and how they can be used to control the sequence of actions in a JavaScript program.

I use this set of exercises as part of an Introductory Day for high school students visiting the college as a taster of what they might do when they join the college. There is enough material here for about 2 one-hour sessions, but this can easily be made shorter or longer.

If you haven’t used much JavaScript before, there are loads of free resources on the Web. One of the better ones is the W3Schools Web site (<http://www.w3schools.com/js/default.asp>) which provides loads of additional examples of JavaScript coding.

Teacher Guide

Before you set students loose on this practical lab, you need to make sure that the student resource folder is put somewhere on your network where students can easily find it. They will need to copy this to their own resource area BEFORE they start to make changes and additions to the code.

A simple Powerpoint presentation is included in the Documentation folder. You can use this as a visual aid to introduce the lesson alongside the working files in the Teacher Files folder.

Introduction and Lab Task 1

In lab task 1, the idea is to introduce students to the structure of a Web page and a JavaScript program.

Firstly, get students to look at the page source for the index page to the lab (they can do this using VIEW> SOURCE in Internet Explorer, which will open up the page in Notepad. Remind students that to see any changes they have made, they will need to save the changes in Notepad (using FILE> SAVE) and then click the REFRESH button in their browser). Slides xxx of the Powerpoint will help you demonstrate this.

Point out that what they see on the page is enclosed by the

```
<BODY>  
</BODY>
```

tags towards the bottom of the page and that their code is towards the top of the page enclosed in the

```
<SCRIPT>  
</SCRIPT> tags.
```

They need to be made aware that clicking one of the buttons causes an EVENT which makes a JavaScript function run.

Buttons 1 and 2 cause something to happen, but they will need to look at the code and make changes to the functions called by the other 3 buttons to open up the correct page.

In the main part of the lab, students look more carefully at the function run by clicking button 1. They need to change it so that the message box shows their name as well as an appropriate message.

Lab Task 2

The "Click Me for lab task 2" button on the main page takes students to a new page. Again they need to be working on the page source.

This task introduces students to the concept of loops, first using the FOR loop and then using the WHILE loop. It also introduces the concept of variables as a temporary store for data.

The extension activity asks students to work out how to number the lines of the message that they print on screen.

Code Summary:

Code	Explanation
<pre>for (i = 1; i <= 10 ; i++) { Code to repeat goes here; }</pre>	<p>Sets the initial value of variable i to 1, on each repeat of the loop, increment i by 1 each time. The loop will terminate if i is greater than 10.</p>
<pre>var i = 1; while (i<10) { your code goes here; i++; }</pre>	<p>Sets the initial value of the counter variable i to 1</p> <p>Repeats the block of code inside { } as long as i < 10</p> <p>Increments i by 1 and...</p> <p>Go back to the start of the block.</p>

Lab Task 3

Here, students find out how to use conditional branching in JavaScript using the IF.. ELSE statements.

The extension activity asks students to use the SWITCH statement in place of IF..ELSE.

Code Summary:

Code	Explanation
<pre>if (word == "GOAT") window.open ("goat.htm"); else if (word == "DOG") window.open ("dog.htm"); else alert("I don't know that!");</pre>	<p>IF the word typed in (allocated to the variable word) is GOAT then open up a new browser window loaded with the file goat.htm</p> <p>IF the above condition isn't met, check IF word is DOG, if so open up a new browser window loaded with the file dog.htm</p> <p>If NONE of the above conditions are met, generate a pop-up message box containing the text "I don't know that!"</p>
<pre>switch (word) { case 1 : window.open ("goat.htm"); break; case 2 : window.open ("dog.htm"); break; default: alert("I don't know that!"); }</pre>	<p>Evaluate the content of the variable word and select the appropriate label in the block of code to execute.</p> <p>All label code needs to be terminated with a break statement (to ensure the label doesn't accidentally execute all the following code).</p>

Lab Task 4

Now we are beginning to draw some of the ideas together make a "Click And Go" navigation selection box. There is lots of scope here for students to make a navigation list for their own list of favourite sites.

The code I have given uses IF..ELSE to navigate to the appropriate Web site for the selection. However, with more than two or three choices, the code gets a bit unwieldy. Encourage students to experiment using the SWITCH statement here to make things easier.

This lab also introduces the concept of PARAMETERS.

This needs a bit of explanation! In the declaration line for the function:

```
function lab4(website)
```

a **parameter** is passed into the **function**. When the user chooses an option from the selection list, the function is called and passes a value into the function. The line:

```
mypage = website.options[website.selectedIndex].value
```

"finds" the correct Web site from the list of options and stores the Web site address in the variable **mypage**.

The line:

```
if (mypage != "") window.open (mypage)
```

opens up a new window at the correct Web site if (and only if) the variable **mypage** IS NOT empty (!= is JavaScript for **NOT EQUAL TO**). If you check the options for the selection list, you will see that the first option, "Choose a search engine" has an OPTION VALUE = "". This makes sure that if we select the first option (the instructions) then the function will not try to take us off to a Web site. In fact, it won't do anything (which is exactly what we want here).

Lab Task 5

Even more fun with this activity! The idea of this is to introduce a few more events:

```
onMouseover  
onDblick and  
onMouseout
```

When the user moves the mouse over one of the flags, the onMouseover event calls a function to display the name of the country in the text box. When they move the mouse off the flag, the textboxclear function is called to clear the text box.

If the user double clicks the flag, a new Web page should open with the Google search page for that country displayed in it. There are other, probably more interesting, options if you don't want to use Google.

Again, two of the functions use arguments.

The **lab5** function uses the argument **country** and the **newsite** function uses an argument **flagid**.

Moving the mouse over the first flag passes 1 as the argument, over the last flag passes 5 as the argument into the lab5 function. Students will use the code given as an example to :

- Write the correct country name in the textbox
- For students, the next stage is to write the rest of the **newsite** function to take them to the correct Google search page when the flag image is double clicked.
- Finally, a function needs to be written to clear the text box when the mouse is moved off the flag image.
- Most students will probably want to use IF statements, but a neater solution would be to use SWITCH.