JavaScript Functions

Functions (sometimes known as procedures or methods)
In programming, functions encapsulate a task. They let you write one line of code to call perform a task that executes several lines of code.

For example, the Math library functions: we can write `Math.min(n1, n2, n3)` because the Math library already contains the code necessary to compute the minimum of a list of numbers.

The values n1, n2 and n3 are called *parameters* to the function.

Writing `Math.min(…)` is called a *function call* to the min function. When your program runs into a function call, it:

1. Enters the function that was called, taking with it any parameters.
2. The code within the function is executed.
3. If there is a *return value*, that value “replaces” the original function call.

```
<script type = “text/javascript”>
function function_name(parameter1, parameter2, parameter3,…)
//Precondition: What is true of the parameters, etc. before
//              executing the function?
//Postcondition: What is true after the function is executed?
//              If there is a return value, what is it?
{
    code to be executed;

    return some_value; //
}
</script>
```

Local variables are declared with the keyword var.
What are *local* and *global* variables?

Example: A very basic function call.

```
<head>
<script type = “text/javascript”>
function blubblub()
{
    return 5;
}
</script>
</head>

<body>

<input type = “button” value = “click me!”
onclick = “document.getElementById(‘outputDiv’).innerHTML =
blubblub();”>

<div id = “outputDiv”> </div>

</body>
```

What will happen when we click the button?