

Lab 6: Functions

University of Central Florida
COP2500

Sarah Loewy

February 29, 2012

Functions

- ▶ Functions are used to generalize code.
- ▶ When you start repeating code with only minor modifications, you should probably make a function.
- ▶ A **function** is a bit of code that can be “**called**” upon, executed, and the program continues executing from wherever it was called.

An Example of When to Create a Function: Initializing Numbers

```
var x = Math.round(Math.random() * 100);  
var y = Math.round(Math.random() * 5);  
var z = Math.round(Math.random() * 80);
```

- ▶ This code creates random integers in various ranges.
- ▶ Let's make a function that creates a random number within a specified range.

A Function to Create a Random Number in a Specified Range

```
function randomIntegerInRange(range)
{
return Math.round(Math.random() * range);
}
```

- ▶ This function, `randomIntegerInRange`, creates a random integer in the specified range.

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

- ▶ We call the function `randomIntegerInRange` with a specified range to initialize `x`, `y`, and `z`.

Trace of Random Number Generation (1)

```
function randomIntegerInRange(range)
{
return Math.round(Math.random() * range);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Trace of Random Number Generation (2)

```
function randomIntegerInRange(range)
{
  return Math.round(Math.random() * 100);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Trace of Random Number Generation (3)

```
function randomIntegerInRange(range)
{
return Math.round(Math.random() * 5);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Trace of Random Number Generation (4)

```
function randomIntegerInRange(range)
{
  return Math.round(Math.random() * 5);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Trace of Random Number Generation (5)

```
function randomIntegerInRange(range)
{
return Math.round(Math.random() * range);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Trace of Random Number Generation (6)

```
function randomIntegerInRange(range)
{
  return Math.round(Math.random() * 80);
}
```

```
var x = randomIntegerInRange(100);
var y = randomIntegerInRange(5);
var z = randomIntegerInRange(80);
```

Declaring a Function

```
function name(arg1 , arg2 , ...)  
{  
return ...;  
}
```

- ▶ A function accepts 0 or more inputs, or **arguments** (arg1, arg2, ...), such as range in the previous example.
- ▶ A function may output, or **return**, a value, but this is not required.

Calling a Function

```
myFunction( arg1 , arg2 , ... );
```

- ▶ Calling, or executing, a function directs the browser to run the function's code with your arguments and, when complete, continue executing at the line where you called the function.
- ▶ If a function has no arguments, you may call it with

```
myFunction ( );
```

- ▶ Declaring a function does not call the function. They must be explicitly called.

Some Functions You're Familiar With

- ▶ `document.write(...)`
- ▶ `alert(...)`
- ▶ `value = prompt(...)`
- ▶ `Math.random()`
- ▶ `new Array()`

Calling JavaScript Functions from HTML Buttons

- ▶ HTML buttons have a special attribute, **onClick**, which calls a Javascript function you specify when the user clicks the button.

```
<button onclick=" add ()">+</button>
```

- ▶ This button would call your JavaScript function `add()` when the user clicks the button.

Accessing HTML with JavaScript

- ▶ Say your HTML has a tag named SomelD

```
<input type = "text" id = "SomelD" >
```

- ▶ You can access tags in the HTML document from JavaScript with **getElementById**

```
var SomelD = document.getElementById( 'SomelD' );
```

- ▶ Your variable SomelD is set to whatever the text box contains.
- ▶ You can **set** the HTML with **getElementById.value**

```
document.getElementById( 'SomelD' ).value = "My new text";
```