

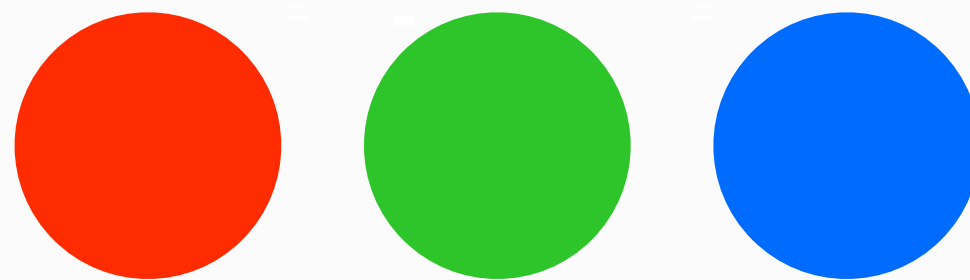
IO Lab: Javascript & jQuery

September 16, 2013

INFO 290TA (Information Organization Lab)
Kate Rushton & Raymon Sutedjo-The

JavaScript

(BEHAVIOR)



Javascript 101

Javascript is a programming language used most often in web browsers.

You can write comments in your code with `//` or `/* */`.

A semi-colon goes at the end of every statement.

Variables

Variables

Variables are how we store data.

A variable has a name and a value.

```
var grade = 94;
```

Data Types

Type	Example
Number	13
String	"Shoe"
Boolean	FALSE
Array	<code>["i202", "i206", "i290ta"]</code>
Object	<code>{"name": "Tina", "age": 57}</code>

String

A sequence of characters.

Use single- or double-quotes to indicate a string.

```
var myName = "Steve"
```

```
myName.length
```

```
5
```

```
myName.toUpperCase()
```

```
"STEVE"
```

```
myName.indexOf("v")
```

```
3
```

Array

An ordered collection of elements.

Use square brackets to indicate an array.

```
var myArray = [  
  "cherry", "banana",  
  "apple"  
]
```

```
myArray.length
```

```
3
```

```
myArray[1]
```

```
"banana"
```

```
myArray.push("watermelon")
```

```
["cherry", "banana", "apple", "watermelon"]
```

```
myArray.sort()
```

```
["apple", "banana", "cherry", "watermelon"]
```


Object

A collection of key-value pairs or named properties.

Use curly braces to indicate an object.

```
var student = {  
  "name": "Robert",  
  "age": 22,  
  "degree": "masters"  
}
```

```
student.name
```

```
"Robert"
```

```
student.degree
```

```
"masters"
```

```
student.lastName = "Smith"
```

```
student.lastName
```

```
student["lastName"]
```

```
"Smith"
```

Conditionals

`if`

`if... else`

`if... else if... else`

`switch`

Loops

for

for/in

while

do... while

=, ==, ===

```
x = 3;
```

```
y = "3";
```

```
x == y; // TRUE
```

```
x === y; // FALSE
```

“=” sets a variable’s value

“==” compares the variables’ value

“===” compares the variables’ value **and** type

Functions

Functions

A function is a set of statements that performs a task or calculates a value.

2 ways to write a function

Function expression

Function declaration

Expression VS Declaration

Function Expression	Function Declaration
<pre>var sum = function(a,b) { return a+b; }</pre>	<pre>function sum(a,b) { return a+b; }</pre>

Expression VS Declaration

```
test();
```

```
var test = function() {  
    alert("test");  
}
```

Nope. (Uncaught TypeError)

```
test();
```

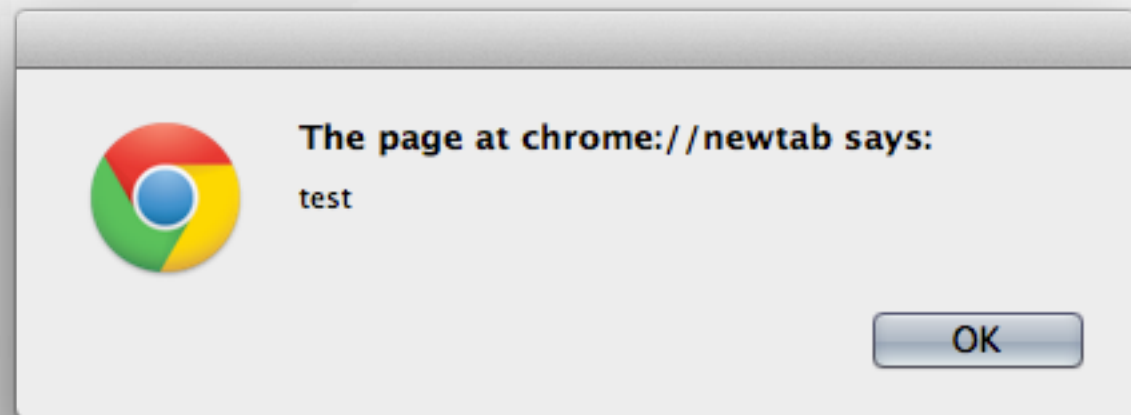
```
function test() {  
    alert("test");  
}
```

Yep.

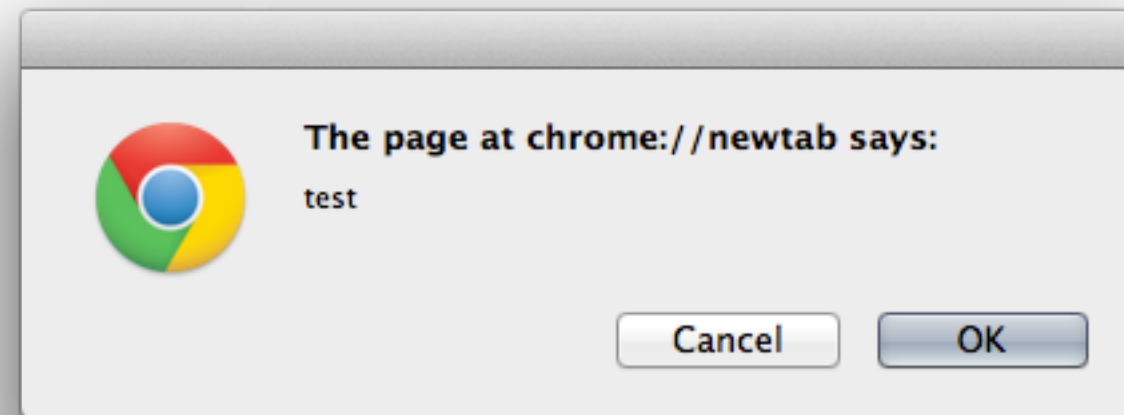
“JavaScript knows about declaration functions and can parse them **before the program executes...** JavaScript has hoisted the function to the top of the current scope.”

Browser Functions

```
alert("...")
```

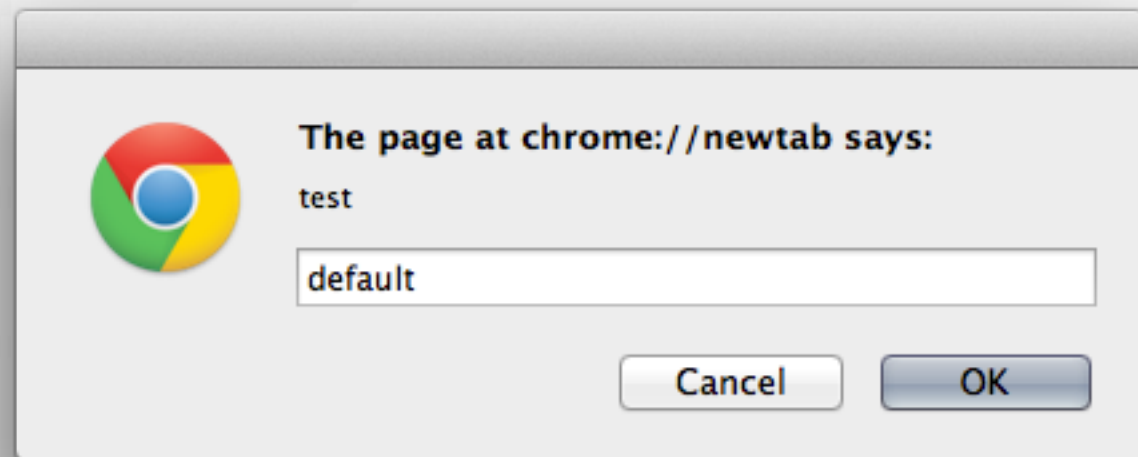


```
confirm("...")
```

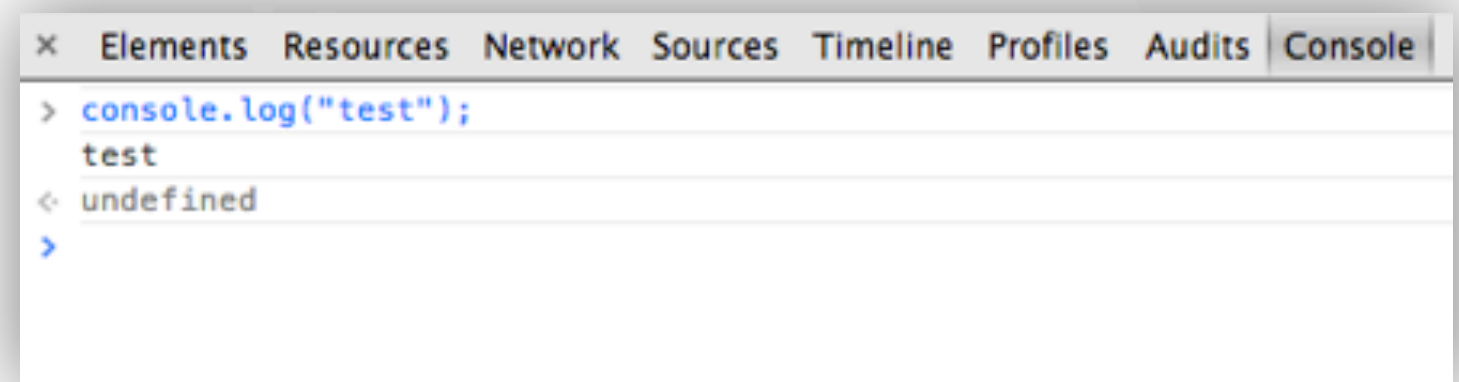


Browser Functions

```
prompt( "...")
```



```
console.log( "...")
```



Global & Local Variables

Global variables exist throughout the script, and it can be accessed by any function.

Local variables only exist in a particular function.

Minimize your use of global variable as much as possible.

Global & Local Variables

```
var x = 123;  
  
check_x();  
  
function check_x() {  
    var x = 456;  
    alert(x);    // 456  
}  
  
alert(x);    // 123
```

Global & Local Variables

```
var x = 123;  
  
check_x();  
  
function check_x() {  
    x = 456;  
    alert(x);    // 456  
}  
  
alert(x);    // 456
```

jQuery

jQuery 101

“Javascript + CSS,” making your life a lot easier.

In this course, we'll use v1.10.2.

Javascript	jQuery
<pre>el = document.getElementById("item"); el.style.color = "#ff0000";</pre>	<pre>\$("#item").css("color", "#ff0000");</pre>

Common jQuery Operations

Events

Attributes

Manipulating

Traversing

Events

Attaching functions to events in the browser.

```
$(".link").on("click", function() {...});
```

Attributes

Getting and setting DOM attributes of elements.

```
var imgTitle = $("#main-photo").attr("title");  
$(".link").css("background-color", "red");
```

Manipulating

Getting and setting properties and values.

Inserting, copying, or removing elements.

```
var firstName = $("form #first-name").val();  
$(".link").remove();
```

Traversing

Moving from selected element(s) in the DOM to other(s).

```
$( "#link" ).parent( );
```

Some Tips

jQuery Chaining

```
// no chaining
$("#menu").fadeIn("fast");
$("#menu").addClass("active");
$("#menu").parent().addClass("open");
```

```
// chaining
$("#menu").fadeIn("fast")
    .addClass("active")
    .parent().addClass("open");
```

Faster to execute.

Easier to maintain.

Code Runtime

```
$(window).on('load', function() {  
    ...  
});
```

```
$(document).on('ready', function() {  
    ...  
});  
  
$(function() {  
    ...  
});
```

Code runs after DOM is loaded by the browser.

Tends to be quicker, since it doesn't wait until **everything** is loaded (HTML, images, scripts, etc).

Getting & Setting

```
<a id="link-home" href="http://berkeley.edu">  
UC Berkeley</a>
```

```
// GET
```

```
$("#link-home").text()  
"UC Berkeley"  
  
$("#link-home").attr("href")  
"http://berkeley.edu"
```

```
// SET
```

```
$("#link-home").text("Berk")  
<a id="link" href="http://berkeley.edu">  
Berk</a>  
  
$("#link-home").attr("href", "http://  
google.com")  
<a id="link" href="http://google.com">  
UC Berkeley</a>
```


.each()

```
<ul id="list">
  <li>one</li>
  <li>two</li>
  <li>three</li>
</ul>
```

```
$("#list li").each(function() {
  $(this).addClass("test");
});
```

```
<ul id="list">
  <li class="test">one</li>
  <li class="test">two</li>
  <li class="test">three</li>
</ul>
```

Some Useful Tools

jsfiddle.net

Test out your Javascript quickly.

jshint.com

Check your script for bugs.

jsccompress.com

Minify your Javascript files for production.

Homework 1

Next Class

Next Class

HTML, CSS, & jQuery lab

Reading

[html5dog HTML tutorial \(beginning & intermediate\)](#)

[html5dog CSS tutorial \(beginning & intermediate\)](#)