COS 426: Precept 1

Huiwen Chang

Outline

- Outline
  - Programming tips for Javascript
  - Python server
  - GUI
  - Assignment 1

Javascript

- JavaScript is
  - an interpreted language.
  - object-based.
  - case sensitive.
  - widely used and supported.
  - accessible to the beginner.

Variables

- A variable can be:

```
var a = 0;
console.log(typeof a);      // → number

var a = "Hello world!";
console.log(typeof a);      // → string

var a =  ["Hello", "COS", 426];
console.log(typeof a);      // → object
```

== vs ===

=== will return false for them all, however == will:

```
''  == '0'            // → false
''  == 0              // → true
0  == '0'             // → true
false == 'false'      // → false
false == '0'          // → true
false == undefined    // → false
false == null         // → false
null == undefined     // → true
' \t\r\n ' == 0       // → true
```

Variables

- can be an array of object:

```
var journal = [
  {events: ["work", "ice cream", "pizza",
            "running", "television"],
   squirrel: false},
  {events: ["weekend", "cycling", "break",
            "peanuts", "beer"],
   squirrel: true},
];
console.log(journal[0].events[1]); // → ice cream
for ( var prop in journal[0] ) { console.log(prop);}
// → events
// → squirrel
```

Variable scope

- In JavaScript, instead of braces, functions are the only things that create a new scope

```
var a = 1;
{
  var a = 2;
}
console.log(a); // → 2
-------------------------------
var a = "outside";
var f = function() {
  var a = "inside f";
};
f();
console.log(a); // → outside
```

Function variables

- Function variables act as names for a specific piece of the program

```
 var Sqr = function( x ) { return x * x; };
```

- Function Declaration

```
 function sqr( x ) {return x * x; }
```

+ not part of regular top-to-buttom flow of control

+ can be used by all the code

- putting this function in a loop/condition block will be dangerous!

Special function

– `alert()`	to display a message box
– `confirm()`	to display a confirmation box
– `prompt()`	to display a prompt box
– `open()`	to open a new window
– `close()`	to close a window
– `write()`	write a string to the Web page
– `console.log()`	outputs a message to the Web Console

Objects

- PROTOTYPE

```
Array.prototype.myUpperCase = function() {
 for (i = 0; i < this.length; i++) {
        this[i] = this[i].toUpperCase();
    }
 };
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.myUpperCase();
document.write(fruits);
```

```
// → BANANA,ORANGE,APPLE,MANGO
```

Objects

```
var testOne = function () {};
testOne.prototype = function () {
  var me = {}, privateVar = 2;
  me.aMethod = function () {
    return privateVariable;
  };
   me.publicVar = "foo bar";
   me.bMethod = function () {
    return this.publicVar;
  };
  return me;
};
for (var i = loopCount; i>0; i--) {
  new testOne();
}
```

```
var testTwo = function() {
   var me = {}, privateVar = 2;
   me.aMethod = function () {
       return privateVar;
    };
    me.publicVar = "foo bar";
   me.bMethod = function () {
       return this.publicVar;
   };
    return me;
};

for (var i = loopCount; i>0; i--) {
  new testTwo();
}
```

loopCount=1000000:

TestOne takes 17ms, while test Two test 43ms.   WHY?

Simple HTTP server

- Open up a terminal and type:
  - `$ cd /home/yourdir`
  - `$ python -m SimpleHTTPServer`
- That's it! Now your http server will start in port 8000. You will get the message:
  - `Serving HTTP on 0.0.0.0 port 8000`

  You can access it via http://127.0.0.1:8000/yourhtml.html

Dat.Gui

- A lightweight graphical user interface for changing variables in JavaScript.
- Link for tutorial (no need to learn how to use it)

  http://workshop.chromeexperiments.com/examples/gui