COS 426: Precept 1

JavaScript

Linguang Zhang

Outline

- Outline
  - Programming tips for JavaScript
  - Python server
  - GUI

JavaScript

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

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

var a = true;
console.log(typeof a);     // → boolean
// can also be null or undefined
```

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);
  console.log(journal[0][prop])
}
// → events
// → ["work", "ice cream", "pizza", "running",
"television”]
// → squirrel
// → false
```

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-bottom flow of control

+ can be used by all the code

Special functions

- `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

Debugging

- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend
- `console.log()` is your friend

== 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`

Objects

- PROTOTYPE

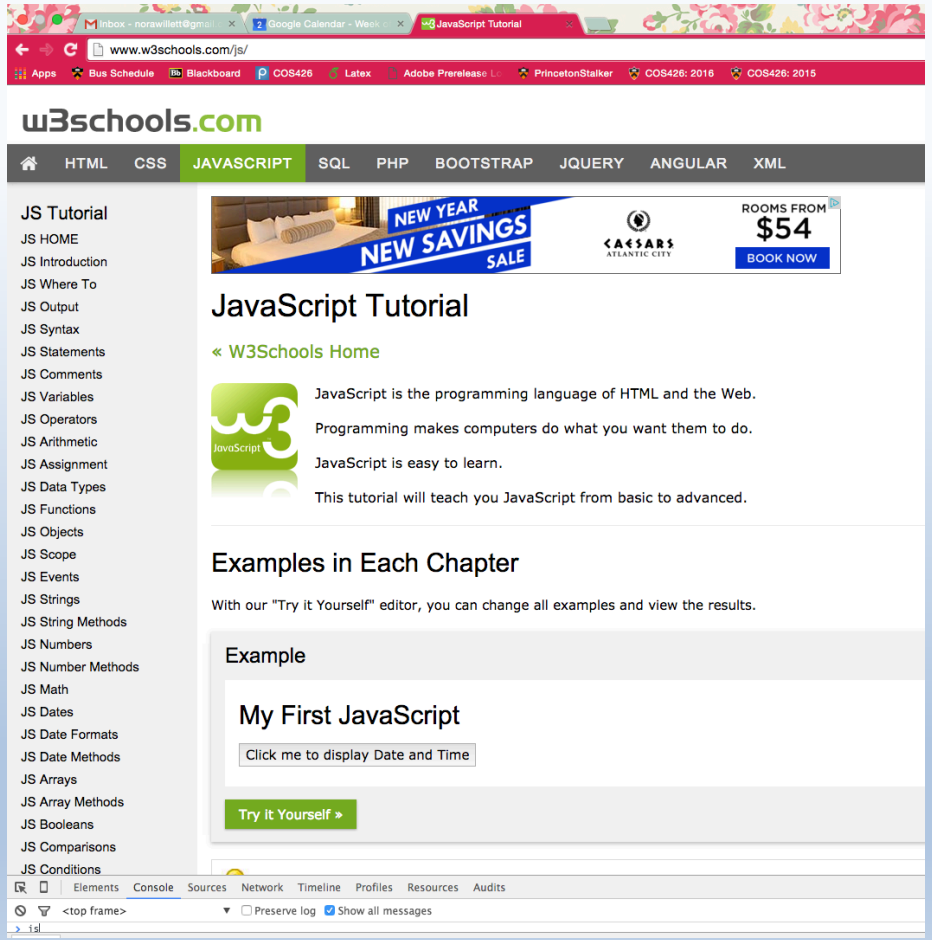
```
Array.prototype.myUpperCase = function() {
    for (i = 0; i < this.length; i++) {
        this[i] = this[i].toUpperCase();
    }
};
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.myUpperCase();
document.write(fruits);
// → BANANA,ORANGE,APPLE,MANGO
```

Objects

```
function Student(name, id) {
    this.name = name;
    this.id = id;
    this.printStudent = function() {
        console.log("student %s: id = %d", this.name, this.id);
    }
}
Student.prototype.printStudent2 = function(){
    console.log("student %s: id = %d", this.name, this.id);
}
var student = new Student("Java", 123);
student.printStudent3 = function() {
    console.log("student %s: id = %d", this.name, this.id);
}
student.printStudent2(); //->student Java: id = 123
student.printStudent3(); //->student Java: id = 123
var student2 = new Student("Script", 456);
student2.printStudent(); //->student Script: id = 456
student2.printStudent3();
//->Uncaught TypeError: student2.printStudent3 is not a function
```

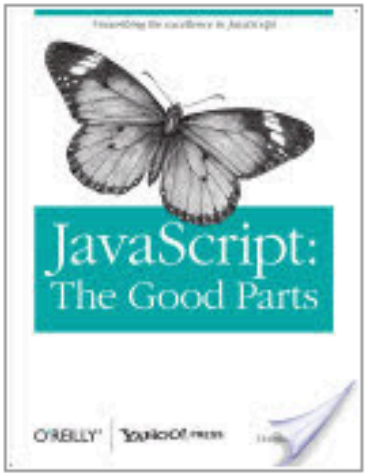
JavaScript Help

http://www.w3schools.com/js/



JavaScript Help

JavaScript: The Good Parts: The Good Parts



Douglas Crockford

"O'Reilly Media, Inc.", May 8, 2008 - Computers - 172 pages

★★★★☆

99 Reviews

G+1 23

Most programming languages contain good and bad parts, but JavaScript has more than its share of the bad, having been developed and released in a hurry before it could be refined. This authoritative book scrapes away these bad features to

More »

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

QUESTIONS?