Intro to JavaScript

Web Valley 2014 -- S. Lorenzo in Banale
Fondazione Bruno Kessler

June 29, 2014

About JavaScript

- JavaScript is NOT Java!
  - Java: static typing vs JS: dynamic typing
  - Java: bytecode vs JS: human-readable code
  - Java: class-based vs JS : prototype-base
- no concept of I/O
- JS runs in a host environment
  - in browsers
  - in mobile phones
  - in commercial sw (Photoshop, Adobe Acrobat, etc.)
  - even in Desktop Environments! (GNOME 3.x)

JavaScript and the web

- most popular client side scripting engine
- server side scripting

  → node.js

Set up and go

- browser's console
  - Chrome/Chromium: CTRL + ALT + j
  - FireFox: CTRL + ALT + k
- a good setup to start:
  - FireFox
  - Firebug add-on (https://addons.mozilla.org/en-US/firefox/addon/firebug/)
  - the scratchpad (Shift + F4)

operators

- arithmetic
  - + - * / %
  - ++ --
- assignment
  - =
  - += -= *= /= %=
- comparison and logic
  - == != === !== > < >= <=
  - && || !

assigning variables

- global scope vs local scope
- the global is the local scope if you are in it!
- the parent scope is reachable by the child scope
- so … always use “var” keyword to avoid strange behaviours

types

- Number, String,Boolean
- Object
  - Function
  - Array
  - Date
  - RegExp
- Null, Undefined

→ *typeof* operator

non-object types

- floats (double-precision 64bit)
  - NaN +Infinite -Infinite
  - the Math object
- “string”
  - methods
  - cast number to string
- *parseInt(), parseFloat(), + (unary op.), + “ “*

non-object types/2

- booleans
  - true, false
  - case sensitive!!
- undefined (not assigned)
- null (assigned with null value)

control structures

- if (test) {} else if (test) {} else {}
- while (test) {}
- do {} while (test)
- for (var i=0; …) and for (i in …)
  - kw: continue, break
- switch (expr) { case n: … default: ...}
  - kw: break

objects

- *key--value* hash tables (python's dicts)
  - var myObject = new Object();
  - var myObject = {};
- Arrays (python's lists)
  - access by index: arrayName[index]
  - push(item) pop() concat(item) join(string)
  - forEach(value, index, array) every(value, index, array)
- moreover: Date and RegExp

objects/2

- but also functions are objects!
  - declared: function doSomething() {}
  - orphan: var a = function Animal(name) {}
- keyword: *arguments*
- method: apply
  - funcName.apply(null, arrayOfArguments)

WAT video

https://www.destroyallsoftware.com/talks/wat

JavaScript and the DOM

- DOM: Document Object Model
  - object: …
  - attribute:
  - JS methods: document.createElement()
- finding elements
  - document.getElementById
  - document.getElementsByTagName
  - document.getElementsBy...

JavaScript and the DOM/2

- adding elements:
  - document.createElement()
  - document.appendChild()
  - document.insertBefore()
- deleting elements:
  - document.removeChild()
- editing elements:
  - document.replaceChild()
  - element.innerHTML = “...”

JavaScript and the DOM/3

- events:
  - <img src="..." onclick="function(){alert("Hey");}" …
  - onclick
  - onload
  - onmouseover onmouseout
  - onmousedown onmouseup
- *element*.addEventListner(*event*, *function*)
  - "onevent" becomes "event" (eg. "onclick" → "click")

JavaScript and the DOM/4

Browser objects:

- the *document* object
  - deals with the DOM
- the *window* object
  - deals with the current window
  - some useful methods: alert() setInterval()
  - useful to access the global scope

embedding JavaScript

- <script type=”text/javascript”>...</script>
- <script src=”path/to/script.js”>...</script>

N.B. : the <noscript> tag

Exercises

Learning the syntax

1. Generate a random integer, and then convert it to binary (write it from scratch!).
2. Using the letters “AGCT”, build a random string 40 chars long and a second of length 4. Now count how many times the second one is found in the first one.

Excerices/2

Learning how to interact with the DOM

1. set a background for the table rows with class “odd”
2. add some list items to the “unordered list”
3. set the value of the image “src” attribute in order to display an image
4. delete the ordered list
5. place a <hr /> object between each pairs of
6. attach an event to the image in order to make it bigger when the mouse hover on it
7. attach an event to the table in order to make it disappear when the user clicks on it, and appear when the button is released (use CSS visibility)
8. add a button to the DOM which make pop an alert window with the “hello world” message

Object Oriented Programming

```
function Animal(name) {
    var size = 5;  //local variable
    this.name = name; //instance property
     ...
}
```

Object Oriented Programming/2

```
function Animal(name) {
    ...
}
function.prototype.myMethod = function () {
    do something;
}
```

Object Oriented Programming/3

```
function Animal(name) {
    this.name = name
}

dog = new Animal(“Snoopy”);
dog.name
```

Object Oriented Programming/4

```
function Animal(name) {
    this.name = name
}
function Dog(name, color) {
    ...
}
Dog.prototype = Object.create(Animal.prototype);
```

Object Oriented Programming/5

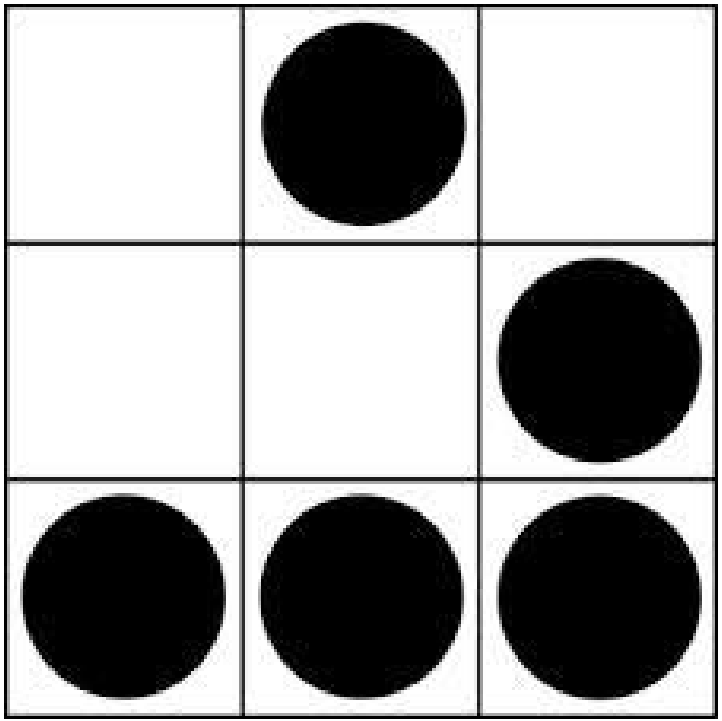
working with namespaces:

```
MYNS = {myFunc : function() { … },
        myClass : function() { … },
        myVar : 5,
        … }
```

a last few words ...

… about work methodology

What's this?



Hacking is ...

1. get something which was created for a purpose
2. disassemble it and study how it works
3. reassemble it in with own changes to reach a different goal
4. while getting FUN in doing this!

hacker or cracker?

actually are both **C**rackers !!!

How to become hackers

1. read some documentation
2. read some more documentation ...
3. then read a lot more documentation !
4. search and study some examples
5. select the best one
6. select even the worst if you can learn something
7. hack them, and when you can’t figure out why it damn doesn’t work …

How to become hackers/2

go back to reading documentation !!!

Guides & references

JavaScript - Mozilla Developer Network

https://developer.mozilla.org/en-US/docs/Web/JavaScript

Hacking culture

http://www.catb.org/hacker-emblem/