

JavaScript

Prof. Cesare Pautasso

<http://www.pautasso.info>

cesare.pautasso@unisi.ch

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

1

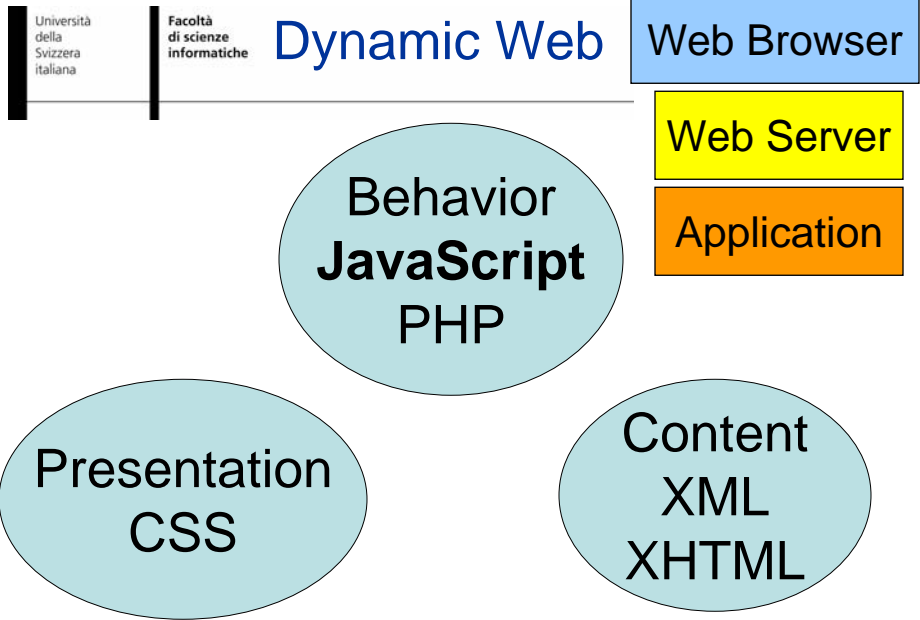
Contents

- Where are we?
- Hallo World JavaScript
- JavaScript vs. Java
- JavaScript:
 - as a programming language
 - as a dynamic scripting language
 - (as an object-oriented programming language)
- Useful JavaScript Functions and Tools

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

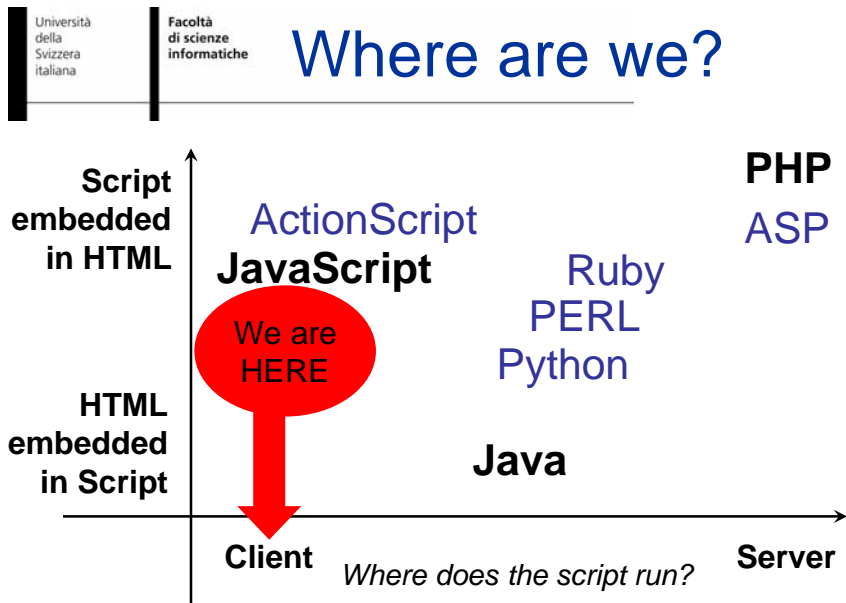
2



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

3



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

4

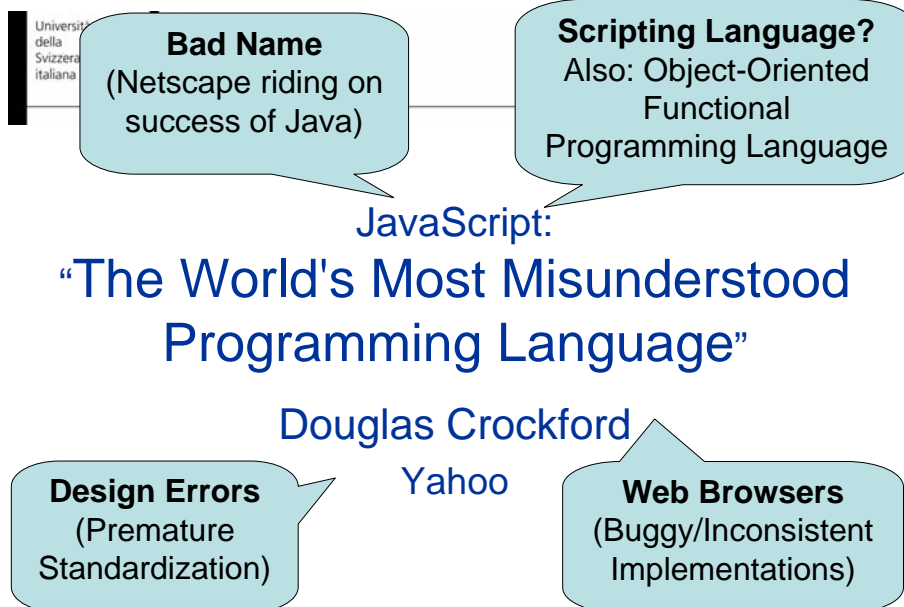
History

- Sun: Java
(1994 Applets target the Web browser)
- Netscape: Mocha/LiveScript/JavaScript
(Navigator 2.0B3 - December 1995)
- Microsoft: JScript (IE3.0 August 1996)
- ECMA Standardization: ECMAScript
(1.0 – 1998, 1.7 – 2006)
- JavaScript “serious” language since
IE6, Firefox1 (stable “enough” platform)

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

5



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

6

Scripting, What for?

- Add behavior to static content
 - (X)HTML + CSS declarative languages for page/text layout description only
 - Dynamically generate Web pages
 - Modify the content of a Web page on the fly
- Run the script on the browser
 - Minimize I/O latency with user interaction
 - Immediate feedback with *input form validation*
 - Immediate reaction to user events
 - Control the browser (history, window, popups, statusbar)

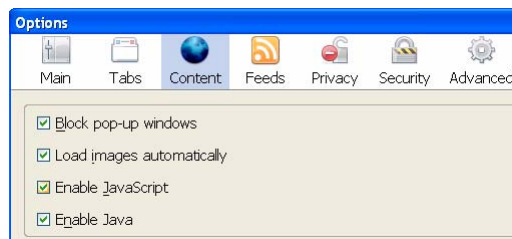
3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

7

Don't forget...

to enable JavaScript
in your Browser!



Use the `<noscript>` HTML element to warn the user about this

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

8

Main JavaScript Features

- Load and go delivery
 - JavaScript *source* code is downloaded and run by the Web browser
- Loose typing
 - Dynamic data types (like scripting languages)
- Objects as general containers
 - **No Classes**
 - Objects similar to “HashMap”
- “Prototypal” inheritance
 - Very different than “classical” object-oriented inheritance
- Lambda
 - Functions are also Objects (like functional languages)
- Linkage though global variables
 - No packages, very little information hiding

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

9



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

10

Hallo World

```
<script>  
document.write("Hello World!");  
</script>
```

1. When does this code run?
2. What is "document"?
3. Where do you see the output?

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

11

When does it run?

1. Immediate Execution
 - During page load (<body><script>)
2. Event Handler
 - Respond to user actions
 - Mouse, Keyboard, Browser Navigation
 - Timeout events

Note: document.write() will append/overwrite the page depending on when it is called

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

12

Event Handlers (on+)

Mouse Events

- click
- dblclick
- mousedown
- mousemove
- mouseout
- mouseover
- mouseup

Page Event

- load
- unload
- resize

Input Event

- blur
- change
- focus
- keydown
- keypress
- keyup
- reset
- submit

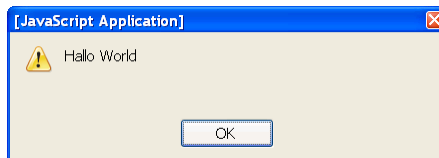
3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

13

Popup Message

```
<body>
<h1
  onmouseup="alert('Hallo World 2' )">
  Click Me</h1>
<form onsubmit="alert('Hallo World' )">
<input type="submit" value="Click Me"/>
</form>
</body>
```

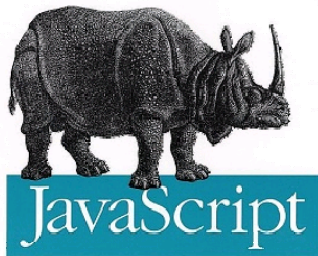


Note: do not use alert()
it will block the whole browser and annoy the user

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

14



JavaScript vs. Java



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

15

JavaScript vs. Java Summary

- JavaScript Language
 - Dynamic Loose Typing
 - Objects only
 - 1 Global scope only
 - No Threads
 - Object-based Exceptions
- JavaScript Runtime
 - Script controls entire Web page/window
 - Sourcecode interpreted by Browser VM
- Java Language
 - Strong Typing
 - Classes & Objects
 - Packages/Class loaders
 - Multi-threaded
 - Support for native code
 - Checked Exceptions
- Java Runtime
 - Applet runs inside a box (or in a new window)
 - Compiled to Bytecode, which is interpreted/JIT by VM

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

16

JavaScript Keywords vs. Java

abstract boolean **break** byte **case catch**
char class **const continue debugger**
default delete do double **else** enum
export extends **false** final **finally** float **for**
function goto **if** implements import **in**
instanceof int interface long native **new**
null package private protected public
return short static super **switch**
synchronized **this throw** throws transient
true try typeof var volatile **void while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

17

No Data Types

abstract boolean **break** byte **case catch**
char class **const continue debugger**
default delete do double **else** enum
export extends **false** final **finally** float **for**
function goto **if** implements import **in**
instanceof int interface long native **new**
null package private protected public
return short static super **switch**
synchronized **this throw** throws transient
true try typeof var volatile **void while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

18

No Classes No Inheritance/Packages

abstract boolean **break** byte **case catch**
char **class** **const** continue **debugger**
default delete do double **else** enum
export extends false final finally float **for**
function goto if implements import in
instanceof int **interface** long native **new**
null package private protected public
return short **static super switch**
synchronized **this throw** throws transient
true try typeof var volatile **void while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

19

No Threads/Native Code

abstract boolean **break** byte **case catch**
char **class** **const** continue **debugger**
default delete do double **else** enum
export extends false final finally float **for**
function goto if implements import in
instanceof int **interface** long **native** **new**
null package private protected public
return short **static super switch**
synchronized this throw throws **transient**
true try typeof var **volatile** **void while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

20

No GOTO

abstract boolean **break** byte **case** **catch**
char **class** **const** **continue** **debugger**
default **delete** **do** double **else** enum
export **extends** **false** **final** **finally** float **for**
function **goto** **if** **implements** **import** **in**
instanceof int **interface** long **native** **new**
null **package** **private** **protected** **public**
return short **static** **super** **switch**
synchronized **this** **throw** **throws** **transient**
true **try** **typeof** **var** **volatile** **void** **while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

21

No Checked Exceptions

abstract boolean **break** byte **case** **catch**
char **class** **const** **continue** **debugger**
default **delete** **do** double **else** enum
export **extends** **false** **final** **finally** float **for**
function **goto** **if** **implements** **import** **in**
instanceof int **interface** long **native** **new**
null **package** **private** **protected** **public**
return short **static** **super** **switch**
synchronized **this** **throw** **throws** **transient**
true **try** **typeof** **var** **volatile** **void** **while**
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

22

JavaScript keywords

break case catch
const continue debugger
default delete do else
function false finally for
instanceof if in
null new
return switch
this throw
true try typeof var void while
with

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

23

JavaScript Keywords

Variable Declaration
var

Constant Declaration
const

Control flow
break case continue
default do else for if
in switch while

Exception Handling
catch debugger
finally try throw

Function Declaration
function
return

Object Manipulation
new this delete

Predefined values
null undefined
false true

Dynamic Typing
instanceof typeof
void

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

24

JavaScript Keywords

Variable Declaration

var

Constant Declaration

const

Control flow

**break case continue
default do else for if
in switch while**

Exception Handling

**catch debugger
finally try throw**

Function Declaration

**function
return**

Object Manipulation

new this delete

Predefined values

**null undefined
false true**

Dynamic Typing

**instanceof typeof
void**

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

25

Comments

```
/*  
  
// slashslash          slashstar  
// line comment       block  
                        comment  
  
*/
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

26

JavaScript DataTypes

- Numbers
 - Only one number type
 - No integers!
- Strings
- Booleans
- Objects
- **null**
- **undefined**
- **NaN** is a number (representing Infinity and Math errors)

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

27

Strings

- **String(value)**
 - Converts any value to a string
 - **string.length**
 - returns the number of characters in a string
 - **“Hallo” ‘World’**
 - String literals can use single “ or double quotes ”
 - useful for putting JavaScript inside HTML attributes
 - **No character type**
 - use Strings of length 1
- String Methods:
- **charAt**
 - **concat**
 - **indexOf**
 - **lastIndexOf**
 - **match**
 - **replace**
 - **search**
 - **slice**
 - **split**
 - **substring**
 - **toLowerCase**
 - **toUpperCase**

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

28

Booleans

`false`

`true`

`null`

`undefined`

`""` (empty string)

`0`

`NaN`

*All other values
evaluate to true*

```
Boolean(value)
if(value) {...}
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

29

Predefined Values

`null`

`undefined`

- A value that isn't anything
- A value that isn't even that
- The **default value** for variables and parameters
- The value of **missing members** in objects

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

30

Operators

- Arithmetic
+ - * / % ++ --
- Comparison
== === != !==
< > <= >=
- Logical
&& || !
- Bitwise
& | ^ >> >>> <<
- Conditional
?:
- Assignment
=
- String Concatenation
+
- Implicit Type Conversion
"+10" = "10"
+"10" = 10
- Guard Operator
var a = b && b.member;
- Default Operator
var a = b || default;

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

31

Conditional Statement

```
var time = new Date().getHours();
if (time < 10)
{
    document.write("Good morning!");
}
else
{
    document.write("Good day!");
}
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

32

For-Loop Statement

```
for (i = 1; i <= 6; i++)  
{  
    document.write("<h" + i + ">");  
    document.write("Header Level " + i)  
    document.write("</h" + i + ">")  
}
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

33

While-Loop Statement

```
while (...)  
{  
    ...  
}  
  
do  
{  
    ...  
}  
while (...)
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

34

```
function name(parameters) {
...
return [expression];
}
```

```
return; = return undefined;
```

Function Scoping: Variables defined in a function are not visible outside of the function.

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

35

```
<html >
<head>
<scri pt>
function product(a, b)
{
return a*b;
}
</scri pt>
</head>
<body>
<scri pt>
document.wri te(product(4, 3));
</scri pt>
</body>
</html >
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

36

```
function map(f, a) {  
  var result=new Array;  
  for (var i = 0; i != a.length; i++)  
    result[i] = f(a[i]);  
  return result;  
}  
  
var square = function(x)  
  { return x * x; }  
map(square, [0, 1, 2, 3, 4, 5]);
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

37

Some Useful JavaScript Functions & Tools

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

38

innerHTML

A much better way to produce HTML from your JavaScript without breaking the existing page structure

```
<head>
<script>
function getTime()
{
  document.getElementById('clock').innerHTML =
    '<b>' + new Date() + '</b>';
}
</script>
</head>
<body>
<p>Time: <span id='clock'>?</span> </p>
<input type='button' onclick='getTime()'
  value='What time is it?' />
</body>
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

39

Accessing the Browser VM

`eval (string)`

- The `eval` function compiles and executes a string of JavaScript code and returns the result.
- It is what the browser uses to convert strings into actions.
- Useful for parsing JSON data
- Can open your code to security risks if you do not fully trust the source of the code string

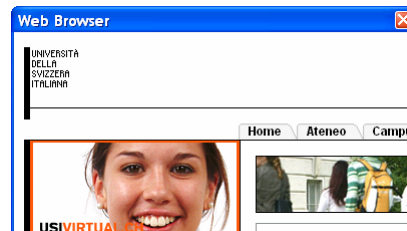
3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

40

Navigation Window Control

- `window.open(`
 `'http://www.unisi.ch',`
 `'usi',`
 `'width=400,height=200');`
- `window.close();`
- `history.back();`
- `history.forward();`



3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

41

Debugger Statement

- The `debugger` statement can be used as a programmable breakpoint.

```
if ( something === 'wrong' ) {  
    debugger;  
}
```

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

42



<http://www.getfirebug.com>

- You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page.

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

43

JSLint – Style Checker

<http://www.JSLint.com/>

- JSLint can help improve the robustness and portability of your JavaScript programs.
- It enforces style rules.
- It can spot some errors that are very difficult to find in debugging.
- It can help eliminate implied globals.

3.10.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

44

References

- Follow the links on Moodle for an in-depth video tutorial on JavaScript by Douglas Crockford
- Danny Goodman, Michael Morrison, **JavaScript Bible**, 6th Edition, Wiley, April 2007
- David Flanagan, **JavaScript: The Definitive Guide**, Fifth Edition, O'Reilly, August 2006
- Mark Pilgrim, **Greasemonkey Hacks**, O'Reilly