HUMAN CAPITAL
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

Introduction to JavaScript

Maciej Grzenda & Michał Okulewicz
Warsaw University of Technology
Faculty of Mathematics and Information Science
M.Grzenda@mini.pw.edu.pl
M.Okulewicz@mini.pw.edu.pl

Motivation

- Static HTML documents do not allow any complex user interaction
- Still HTML forms need validation and extra processing
- Dynamic content like pop-up calendars, menus, etc. also is required for modern interface
- State-of-the-art user-friendly interface is fundamental for modern Web 2.0 applications
- AJAX requires JavaScript

JavaScript?

- JavaScript is a programming language primarily used to enrich the functionality of web sites by adding dynamic interaction with the user
- The language allows to modify the HTML document content, validate user's input and initiate the communication with the server
- In many cases it can be used instead of server-side scripts like JSP, ASP/ASP.NET or PHP. For instance, it can be used to check whether the user provided his name in an HTML form before actually sending the form data to the server

JavaScript and ECMAScript

- JavaScript evolved from its first versions available in Netscape browser. The success of the language was followed by Microsoft’s JScript
- Based on both languages, a formal standard has been developed named ECMAScript
- The name is not widely used, still the ECMAScript specification can be used to learn web scripting,
- However modern JavaScript is not strictly compliant with ECMAScript. It is rather considered as a dialect of ECMAScript

For these reasons:
1.http://www.w3schools.com/js/default.asp can be used for JavaScript tutorials
2.http://www.ecmascript.org/ contains formal standard and its current release at http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

Basic facts about JS in web programming

- The language syntax is similar to the syntax of C/C++/Java
- The JS code is downloaded to the web browser and **interpreted in the browser**.
- **There is no a priori compilation of the code in the standard**.
- Since the JS code is interpreted by the web browser and may be used to modify HTML document it may help reduce the volume of server-side processing, thus:
  - Decrease hardware requirements,
  - Decrease response time of web interface
- Many JS errors remain undetected until the browser attempts to execute a particular JS function e.g. until the browser tries to validate e-mail field of the form

JavaScript as a programming language

- The language is case sensitive
- The code may seem to be similar to Java/C++, but:
    - JS is a dynamic programming language –new code can be added on-the-fly,
    - JS does not allow to define classes (till ES 6),
    - Variable types are not declared (although ES 6 has typed arrays)
    - http://kangax.github.io/compat-table/es6/

Learning JS

- The JavaScript itself and the way it is applied to extend the functionality of HTML pages is a vast subject, with books on it exceeding 1000 pages.
- One of many web tutorials dealing with JS:

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

- JavaScript and HTML DOM reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

http://www.w3schools.com/jsref/default.asp

The relation between JavaScript and DOM (Document Object Model) is discussed below in this lecture.

JavaScript and HTML

- The JavaScript code can be combined with HTML in one of two basic ways:
  - Script code in the header section of the document or in the BODY section:

```
<head>
    <script type="text/javascript">
    …
    </script>
    …
```

  - A separate file referenced from the document:

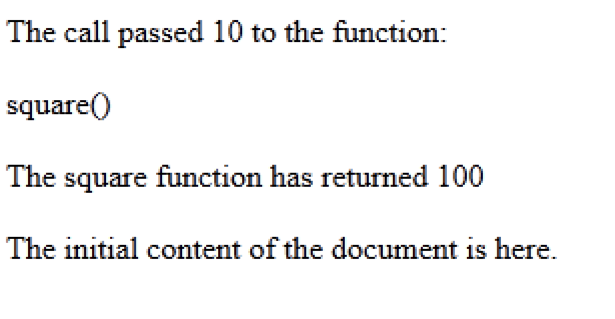
```
<head>
<script type="text/javascript" src="myscript.js">
</script>
…
```

Similarly to CSS, the preferred form is to put JavaScript code into a separate file/several files. One more reason is the fact the JavaScript code may be quite complex. Hence, by repeatedly embedding it HTML we largely increase network transfer and slow down the system.

Embedded JavaScript - example

js_example_1.html

```
<html>
<head>
   <title>The first JavaScript document</title>
   <script type="text/javascript">
   /** multiple-line comment **/
   // a simple function - only parsed when the HTML document is
   downloaded, but not executed yet
   function square(i)
   {   document.write("The call passed ",i," to the function:"
   ,"<P>square()<P>");
       return i*i;
   }
   //this code is  interpreted and executed when the header is parsed
   i.e. when the document is downloaded
   document.write("The square function has returned ",square(10));
</script>
</head>
<body>
<p>The initial content of the document is here.</p>
</body></html>
```



The call passed 10 to the function:

square()

The square function has returned 100

The initial content of the document is here.

JavaScript – external files revisited

```
…
<head>
...
// As in the example above the code that is not a part of any
   function will be executed immediately. The remaining code
   will be just parsed

<script type="text/javascript" src="test.js"></script>
<script type="text/javascript" src="myvalidation.js"></script>

</head>

…
```

1. In this case JavaScript code is placed in a separate file.
2. A number of JS files can be referenced from the same HTML document.
3. This method makes it possible to share the same typical JS code in a number of HTML documents.
4. JavaScript libraries (prototype, jQuery, json, ExtJS etc.) can be used in this way

General statements of the language

- Conditional statements

  -
    ```
    if (x < y) {z = y – x;} else {z = x – y;}
    ```
- Loops

  -
    ```
    for (var i = 0; i < array.length; i++) {
      if (i % 2 == 0) {continue;}
      process(array[i]); }
    ```

  -
    ```
    while (true) {
      process(array[i]); i++;
      if (i > array.length) { break; }}
    ```
- Exceptions processing

```
try { throw “Exception code”;}
catch (e) {console.error(e);}
```

- Function definition

```
function myNewFunction(arg1, arg2) {
    return arg1 * arg2;
}
```

JavaScript types

- The types in the JavaScript are not declared, but the objects have type
  - Object
  - Array
  - Number
  - String
  - Date
- Custom objects creation is also possible

```
function MyObject() {
    this.property1 = 'Value';
    this.concatenate = function(arg) {
        return this.property1 + arg;
    }
}
```