

Anno Accademico 2007-2008

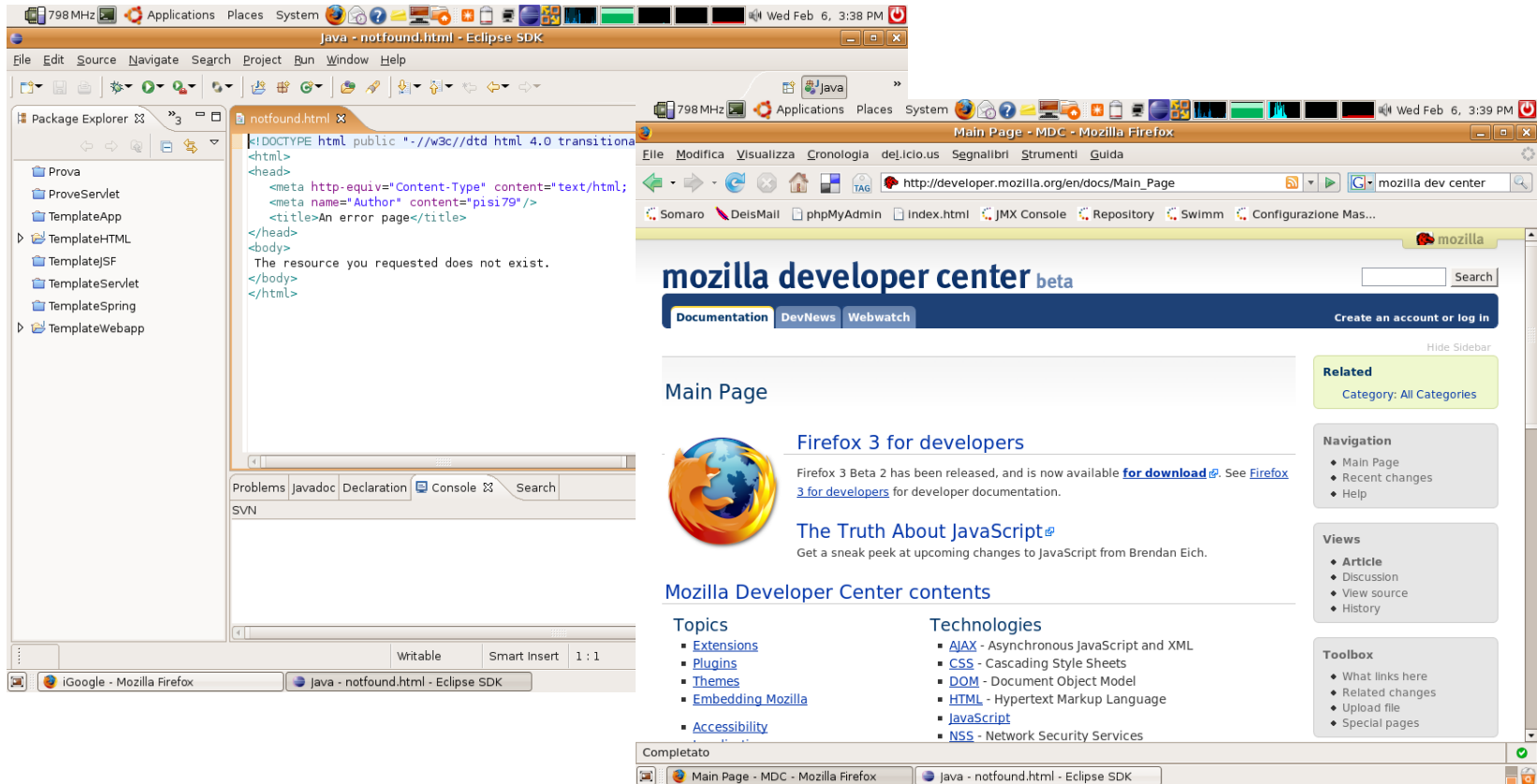
Laboratorio di Tecnologie Web

Sviluppo di applicazioni web
HTML, CSS e Javascript

<http://www-lia.deis.unibo.it/Courses/TecnologieWeb0708/>

Ever told you that... ?

> You will be using two major tools: *Eclipse* and the *browser*...



> ...and the most important of these is.. *OK, I guess you know the answer*

HTML, CSS and Javascript

- > All technologies meant to run client-side:
 - users access your web application through their browser application window
- > Understand the interaction paradigm
 - **HTTP** it's all about **request / response**
 - results conveyed via static resources (*HTML pages, CSS styles*)
 - presentation layer (client-side) is separate from the biz logic (server-side)
 - But **browsers can perform operations** too!
 - reacting to events (such as mouse movements) by dynamically changing pages (DOM and style modifications) without sending new requests
 - supporting scripting techniques (*Javascript*)
 - Up to breaking the synchronous request / response interaction paradigm
 - Advanced techniques (e.g., *Applet, AJAX*) make user intervention unnecessary: browsers autonomously perform '*background*' requests

styles and HTML: getting started

> Styles enable associating formatting properties and document elements

- definitions within the **style** attribute of HTML tags...

```
<h1 style="display:block"> .... </h1>
```

- ...within the document **head** element

```
<style type="text/css"> ..... </style>
```

- ...or in external stylesheet files

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

> Cascading: several layers of style definitions can apply to any document

- user-agent settings (the browser default behaviour) → linked style sheets → document head styles → tag hard-coded styles
- **inner layers override outer ones** in case of conflicts

CSS selectors

> Style definition format:

- head and external sheet:
- tag style attribute:

```
selector { property1: value1; property2: value2; }
```

```
<tag style="property1: value1; property2: value2;">...</tag>
```

> Selections:

- tag name
- selector list
- DOM pattern
- class attribute
- id attribute
- attribute presence
- attribute values
- pseudo-classes
- pseudo-elements
- wildcard usage

```
h1 { color: red; }
```

```
h1, h2, h3 { color: red; }
```

```
tr td p { color: red; }
```

```
p.titleclass { color: red; }
```

```
#contentid { color: red; }
```

```
table[border] { color: red; }
```

```
table[border="3"] { color: red; }
```

```
:link :visited :active :hover :focus :first-child
```

```
:first-line :first-letter
```

```
tr * p, *.title, ....
```

Which property applies to which element?

> You can do lots of thing with styles:

- text *style*, dimension, **color**, font, alignment
- background color, images
- spatial positioning, margins, borders, paddings
- layout flow

> Styles understand (among the rest) the following property value metrics:

- CSS keywords an specific properties (thin, thick, bolder, transparent, ..)
- Real-world measures (in, cm, mm, pt, ...)
- Screen measures (px, em, ex, %, ...)
- Color codes (#rrggbb, rgb(r,g,b), ...)

> Orienteering

- There are just so many things to remember
- Hey, do you remember? Search the web!
(e.g., http://developer.mozilla.org/en/docs/CSS_Reference#Properties)

Let's play with pages, forms and styles

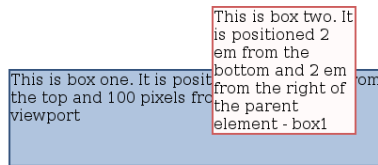


Some sample html and css pages

- [block versus inline](#)
- [div positioning #1](#)
- [div positioning #2](#)
- [centered div](#)
- [floating text around images](#)
- [item gallery](#)
- [navigation menu](#)
- [navigation tabs](#)

Some sample forms

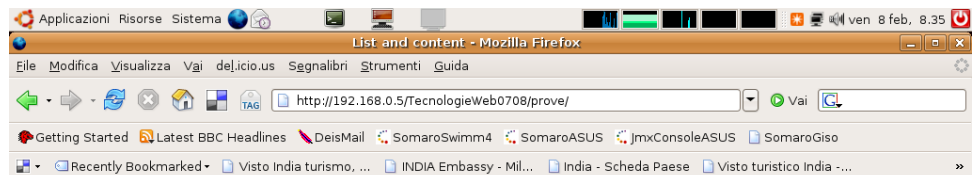
- [the simplest form ever](#)
- [button and field](#)
- [selection field](#)
- [several fields](#)
- [even more fields](#)
- [many kinds of questions](#)
- [spreadsheet-like form](#)



http://192.168.0.5/TecnologieWeb0708/prove/pages/02.position.html

List and content - Mozilla Firefox

*I'll be using **Firebug** to show you this, but it's just a tool (a **Firefox extension** anyone can install)! You can sniff code, styles, etc.. by just following links in the page source*



Some sample html and css pages

- [block versus inline](#)
- [div positioning #1](#)
- [div positioning #2](#)
- [centered div](#)
- [floating text around images](#)
- [item gallery](#)
- [navigation menu](#)
- [navigation tabs](#)

Some sample forms

- [the simplest form ever](#)
- [button and field](#)
- [selection field](#)
- [several fields](#)
- [even more fields](#)
- [many kinds of questions](#)
- [spreadsheet-like form](#)

What is your name?

Select your favorite color:

blue

Are you male or female?

Male

Female

Comments:

http://192.168.0.5/TecnologieWeb0708/prove/forms/form2.html

List and content - Mozilla Firefox

Some CSS deepening...

```
/* Formatting text; an example... */
```

```
p {  
  color: #ffffff;  
  font-family: Verdana, sans.serif;  
  font-size: 6px; o 80%;  
  font-weight: bold;  
  background-color: #ff6600;  
  text-align: center/right/left/center;  
  line-height: 2.0;  
}
```

```
/* Link; highlighting and colors */
```

```
a:link, a:visited {  
  text-decoration: none;  
  color: #ff6600;  
  background-color: transparent;  
}  
a:hover, a:active {  
  text-decoration: underline overline;  
  color: #191970;  
  background-color: #c9c3ed;  
}
```

```
/* Images... */
```

```
img {  
  border: 1px solid #000000;  
}  
body {  
  background-image: url(foto.gif);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-attachment: fixed;  
}
```

```
/* How to center page content... */
```

```
html, body {  
  margin:0;  
  padding:0;  
  text-align:center;  
}
```


Some more CSS deepening...

Blocks: *border / padding / margin*
margin governs distances among blocks
padding allows for indentation
border rules borders :)

Lists

```
list-style-type: none / disc / circle / square..  
list-style-image: url(foto.gif);  
padding-left:0;margin-left:0px; // no indent  
display: inline; // horizontal menu
```

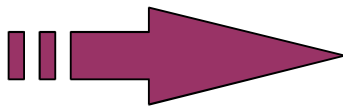
Div and layout

```
width, height  
top, bottom, left, right  
position: absolute / relative  
float: left / right  
...
```

...further information... ...and inspiration:
<http://www.csszengarden.com/tr/italiano/>
<http://css.html.it/>
<http://developer.mozilla.org/>

Javascript

- > **JavaScript** is an object-oriented scripting language
 - it is not Java (although syntax is somewhat similar)
 - **interpreted**, not compiled
 - nowadays, the vast majority of Web scripting relies on Javascript
- > Developers keep up with standards and recommendations at different rates
 - Javascript was introduced by Netscape
 - VBScript is an extension of Visual Basic created by Microsoft as a competitor
 - Internet Explorer provides support for JavaScript (calling it Jscript) too
 - Netscape but could not control Javascript features any longer as it became widely adopted
 - ECMA (European Computer Manufacturers Association) defined a standardized version of JavaScript, called ECMAScript.



*as long as there is more than one browser,
there will be more than one way of doing things*

What you can do with Javascript

- > **Manipulate variables and objects** in a document (i.e., in a web page)
 - **change the value of all the properties** of all the objects in the page
 - DOM (Document Object Model) requires **browsers to redraw pages** in response to events, **without further requests to the server**

- > For instance...
 - dynamic forms, displaying fields based on information already provided
 - *...if you said yes to this question than provide input for other fields...*
 - reward screen interactions by providing graphical effects
 - *...a congratulatory animation if all questions were answered right...*
 - *...animate buttons and links as the mouse moves over them...*
 - order page items based on user provided criteria without reloading server data
 - *...sort the results of a database table based on the requested sort order...*
 - etcetera...
 - *...dynamically changing the course web site stylesheet!*

What you cannot do with Javascript

- > JavaScript is limited to its own sandbox within the browser:
 - you cannot manipulate files on the client computer (i.e., creating, writing, or deleting them)
 - you cannot execute any operations outside of the browser (e.g., launching an installer, initiating a download, ...)
- > Java Applet and ActiveX controls can do more!
 - ...yes, but many visitors disable browsers support for them, as they fear malicious programs

Writing code

> Fairly basic syntax:

- code lines **should** end with a semicolon ';' (few exceptions, such as lines that end in a block delimiter '{' or '}')
- blocks of code (functions, if/for statements, ...) are enclosed in braces: '{', '}'
- explicit declaration of variables is **not necessary**, but it is a good idea
- variable names are case-sensitive and can contain alphabetic and numeric characters

> JavaScript supports a **wide range of variable types** (integer, float, string, ...) but provides **very loose variable type checking**

- you can change the value type stored in variable across your script

> Objects in the document are accessed through the **document's object collection** or the **DOM implementation and methods** that the browser provides

```
<form name="formname" id="formid"  
action="someaction" method="post">  
  <input id="addressid"  
  type="text" name="address"/>  
  ...  
</form>
```

→ `document.formname.address`

→ `document.getElementById('addressid')`

Have a look at...

http://developer.mozilla.org/it/docs/II_DOM_e_JavaScript

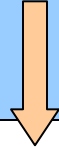
Writing useful code

> Including JavaScript into your HTML documents through the **<script>** element

```
page.html
<script language="javascript">
  <!-- Hide script from incompatible browsers
  ...script here...
  // finish hiding script -->
</script>
```

or

```
page.html
<script language="javascript" src="code.js">
</script>
```



```
code.js
...script here...
```

> **<script>** holds the code, but when does it execute? It depends on where it is and how it is written (definition of functions vs. sequence of instructions)...

- something that runs when a certain condition (event) is met on the page → it must appear before the point that will encounter the event and be invoked when it occurs
- something to run while loading the page → it must appear in the page code itself and provide instructions that browser will execute while rendering the page
- some effect on the initial display of the page → run before the page is loaded

Reacting to events

> An event is:

- an action taken by the visitor sitting at the browser
- something caused by the browser (e.g., the page finishes loading)

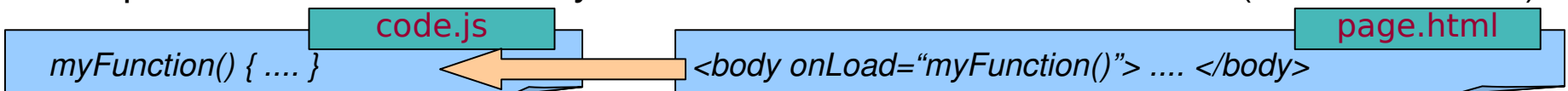
> Some major scriptable events:

Load	page is loaded
Unload	page is unloaded (usually when another page is called)
MouseOver	mouse goes over an object on the page
MouseOut	mouse is no longer over an object it was formerly over
Click	visitor both click and release an object
keyPress	a keyboard key is depressed and released
keyDown	a keyboard key is depressed
keyUp	a keyboard key is released

> Form-only events:

Focus	cursor moves to highlight a field
Blur	cursor moves away from a field that was formerly in focus
Submit	the object clicked is a BUTTON or INPUT element with a type of "submit"
Reset	the object clicked is a BUTTON or INPUT element with a type of "reset"
Change	the contents of the object in focus are changed and then the focus leaves this object.

> Scripts associate to events by means of 'event handler' attributes (***onEventName***)



Have fun with Javascript...

breadcrumbs

Here is some content. And a [link](#) to the index.html page in the subfolder dir of the current page location.

images rollover

The simplest way of doing it is with a "mouseover" image and a "mouseout" image.

A more reusable way, with Javascript (look at the source):

Another way, with Javascript (look at the source):

...and much much more (with patience..)

block positioning

Try to make this work in Internet Explorer!!

client-side form validation

Mailing list subscription

Please enter details below:

Nick name: (minimum 1 character)

Email address:

Please enter a valid email address in the form: yourname@yourdomain.com

Catch me if you can (ps: your browser window size is 1024 x 507)

Tecnolog



...but remember

> Cross-browser support is always an issue

- try downloading pages, styles and scripts from the course web site, then add `alert(chlds[name])` in the `for` loop of `hideshow.js`: now browse your local version of the site both in Internet Explorer and Firefox and see how different DOM implementations can be!!

> Anyway, there are (less and more elegant) ways to deal with it

Using **if / then statements**, you can provide the appropriate code for various browsers

```
browser=navigator.appName
if (browser.indexOf("Microsoft")!=-1)
{
    // MSIE browser specific code here
}
if (browser.indexOf("Netscape")!=-1)
{
    // vintage Netscape browser specific code here
}
if (browser.indexOf("Mozilla")!=-1)
{
    // Mozilla (= Firefox) browser specific code here
}
```

You can tell if a function, method, or property **exists by** using an **if** statement

```
if (window.focus)
{
    // window.focus() is supported, use it
}
else
{
    // window.focus() is not supported,
    // use alternate method
}
```