

Universita' degli Studi di Bologna
Facolta' di Ingegneria

Anno Accademico 2008-2009

Laboratorio di Tecnologie Web

HTML e CSS

Javascript e il DOM

<http://www-lia.deis.unibo.it/Courses/TecnologieWeb0809>

Un'altra semplice applicazione web

- Per ottenerla...
 - ...scaricare lo zip dal sito del corso, senza estrarre nulla (*TemplateHTML.zip*)
 - ...importare come “existing project” in formato archivio
 - *File* → *Import* → *Existing project into workspace* → *Select archive file...*
- Per poterla utilizzare...
 - ...settare le proprietà relative all'ambiente locale in *environment.properties*
 - $\$ \{jdk-home\}$: directory di installazione di un JDK ≥ 1.5
 - (nota: all'interno di Eclipse, ANT ignora il nostro valore! Eclipse lo imposta prima di noi e in ANT le variabili sono **write once!**)*
 - $\$ \{appserver.home\}$: directory di installazione di Tomcat
 - Le proprietà possono essere definite in termini di altre proprietà!
 - ...lanciare i target di ANT
 - trascinare *build.xml* nella *Ant view*
 - *Double-click* sul target desiderato



Troubleshooting

- Tomcat non parte (non riesco ad accedere a <http://localhost:8080/>)
 - JRE_HOME è settato (ed esportato, in linux) correttamente?
 - la directory `$TOMCAT_HOME/logs` esiste?
 - cosa riporta il file `$TOMCAT_HOME/logs/catalina.out` ?
 - eventuali avvii precedenti sono stati terminati (porte TCP già in uso) ?
- Su <http://localhost:8080/TemplateHTML> non compare nulla (error 404)
 - `environemnt.properties` è stato modificato e salvato correttamente?
 - è stato eseguito il **deploy**?
 - il descrittore `web.xml` contiene errori? (in questo caso no, ma in futuro..)
- Ho modificato qualcosa ma non vedo cambiamenti
 - è stato eseguito **undeploy / deploy** ?
 - stiamo vedendo la risorsa remota o la cache del browser?
(capita specialmente nel caso di pagine HTML statiche... anche nel sito del corso!)



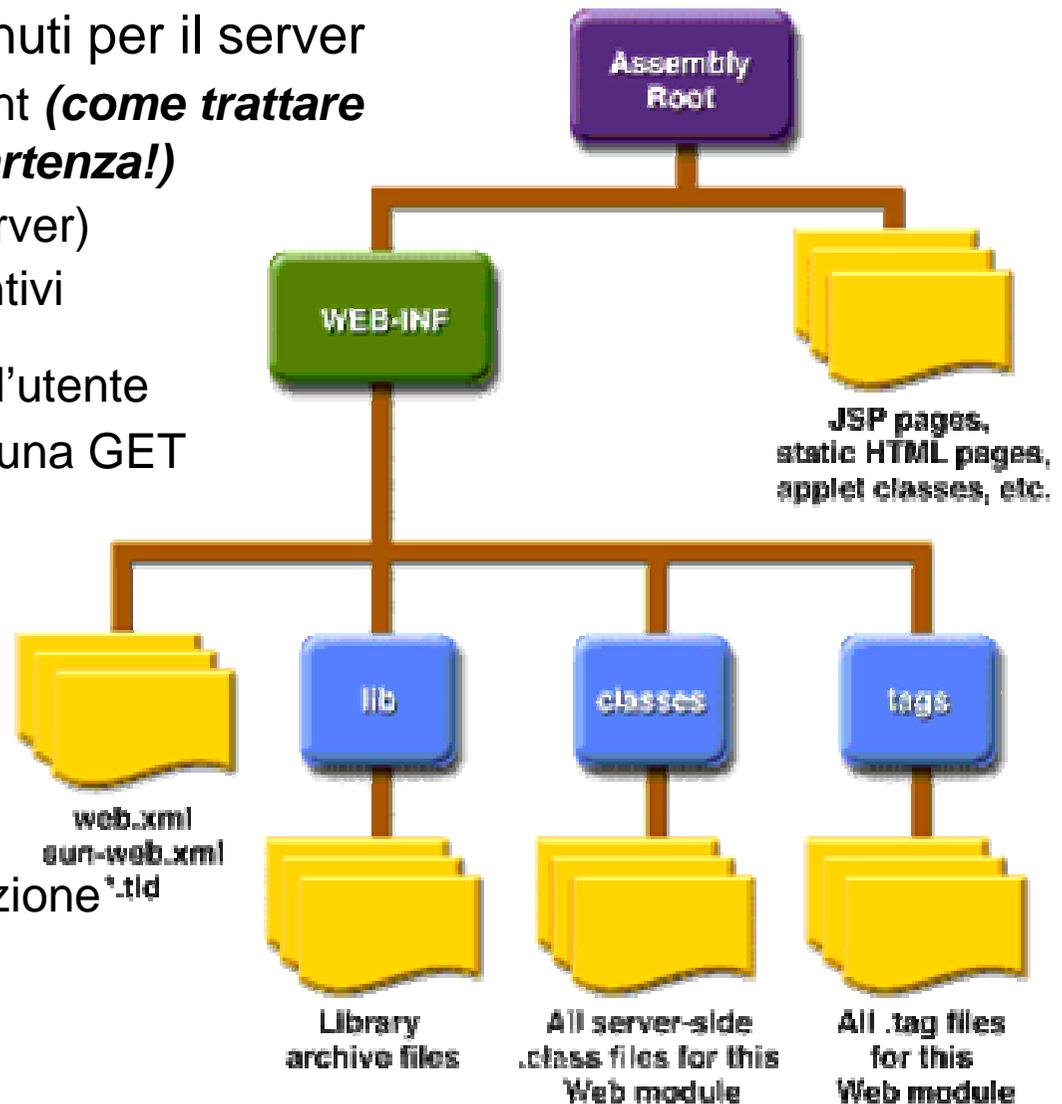
Contenuto dell'esempio

- Sito con risorse statiche (scaricate tali e quali dal server)
 - pagine HTML (renderizzate dal browser)
 - script Javascript (interpretate dal browser)
 - niente codice Java (eseguito dal web server)
- Un primo esempio di deployment descriptor
 - nome dell'applicazione
 - pagine di benvenuto
 - pagine di errore
- L'occasione per...
 - prendere manualità con il binomio IDE + Web Server (due entità diverse anche se spesso gli IDE integrano versioni ridotte dei server o li pilotano)
 - fare considerazioni pratiche su HTML e il protocollo HTTP
 - introdurre il binomio Javascript e DOM



Struttura generale di una webapp Java

- All'interno di WEB-INF: contenuti per il server
 - informazioni per il deployment (**come trattare l'applicazione! il punto di partenza!**)
 - classi Java (eseguite lato server)
 - librerie di componenti aggiuntivi
- Esterno di WEB-INF: risorse per l'utente
 - direttamente accessibili con una GET
 - pagine HTML
 - scripts (Javascript, ...)
 - fogli di stile (CSS)
 - immagini
 - altre risorse (es: flash) per l'uso lato client
 - ottenute attraverso l'elaborazione *.tid da parte del web server
 - pagine JSP
 - pagine JSF



Il deployment descriptor WEB-INF/web.xml

- In questo semplice esempio...
 - ...file di benvenuto
 - una lista di pagine che il server mostra in risposta a un URI valido, ma che non è direttamente mappato su nessun componente web (pagina HTML, pagina JSP, servlet, risorsa, ...)
 - il server usa a default *index.html* o *index.jsp*
 - se non esistono, restituisce (a default) il listato del direttorio
 - ...gestione degli errori
 - controllo delle pagine restituite in caso di errori
 - è possibile specificare associazioni tra
 - pagine d'errore e codici HTTP (403, 404, 500, ...)
 - pagine d'errore e tipi di eccezioni Java



Effetti del deployment descriptor utilizzato

The screenshot illustrates the deployment of a web application. On the left, the Eclipse IDE shows the `web.xml` deployment descriptor with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.d
<web-app>
  <display-name>TemplateHTML</display-name>
  <description>A template web app just to hold HTML pages</description>
  <!-- === Welcome files ===>
  <welcome-file-list>
    <welcome-file>welcome.html</welcome-file>
  </welcome-file-list>
  <!-- === Error handling ===>
  <error-page>
    <error-code>404</error-code>
    <location>/errorpages/notfound.html</location>
  </error-page>
</web-app>
```

On the right, the Mozilla Firefox browser shows the application running at `http://localhost:8080/TemplateHTML/`. The page content is:

Guess, we are about to say...

Hello world!

Click [here](#) to go to another page.
Click [here](#) to go to a non-existing page.

Below this, another browser window shows an error page at `http://localhost:8080/TemplateHTML/nowhere.html` with the message: "The resource you requested does not exist."

Tecnologie client side e siti (più o meno) statici

- Gli utenti accedono alle applicazioni web attraverso il browser, il quale...
 - “renderizza” la pagina (struttura, markup e stili... la parte statica: HTML e CSS)
 - può reagire ad eventi (pressione di tasti, movimento del mouse... DHTML)
 - può eseguire script (es: javascript)
 - attiva oggetti “embedded” (es: flash, wmv, applet), in genere gestiti da plugin
- L’interazione si basa sul protocollo HTTP
 - paradigma **sincrono request / response** (una richiesta per ogni pagina di contenuti)
 - logica di presentazione dei contenuti (lato client) separata da quella di elaborazione delle richieste (lato server) usate per generare i contenuti stessi
 - effetti dinamici lato client ottenuti modificando il DOM della pagina in reazione ad eventi (apertura di menu, evidenziazione, popup, ...)
 - tecniche avanzate (AJAX) permettono di rompere la sincronicità eseguendo script e richieste in background e di aggiornare con nuovi contenuti solo parte della pagina corrente



HTML e HTTP

- In laboratorio l'accesso Internet è limitato alla sola rete



Il logo di Google non fa parte della webapp, ma fa riferimento a una risorsa remota: la richiesta non è diretta in locale, ma in remoto... e fallisce causa firewall del laboratorio ☹

L'immagine che appare/scompare al movimento del mouse è ottenuta raccogliendo un evento DHTML e agendo in sua corrispondenza sugli stili di una parte del DOM della pagina

Le ricerche in Google sono pilotate da parametri di GET. Possiamo creare link predefiniti che ne usano differenti...

...oppure raccogliarli attraverso diversi campi di input all'interno di una form la cui action genera la richiesta che ci interessa

Nonostante siano accorpate in un frameset, si tratta comunque di 3 pagine diverse!

L'uso dei frame è inoltre deprecato! Qui sono solo stati usati solo per vedere l'attributo target degli anchor HTML senza aprire più finestre.

I plugin di Firefox...

- Il browser lascia esaminare il codice sorgente della pagina in maniera molto semplice

- Ad esempio, in Firefox...

View → Page source (CTRL+U)

- L'architettura a plugin dei moderni browser permette tuttavia di aggiungere loro numerose funzionalità utili per lo sviluppo.

- Sul sito <http://addons.mozilla.org>, in particolare...

Firebug (<https://addons.mozilla.org/it/firefox/downloads/file/44490/firebug-1.3.0-fx.xpi>: analisi e modifica al volo di HTML, CSS e Javascript, con supporto al debug), **Web Developer**, **XML Developer Tools**, **XPath Checker**, (https://addons.mozilla.org/it/firefox/downloads/file/5050/xpath_checker-0.4.1-fx.xpi: valutazione di query XPath sul documento aperto, ...) ...**e molti altri**

- Essenziali non solo per imparare, ma proprio per sviluppare!



Altre pagine HTML... esempi di vari tipi di tag

- In realtà, per vedere pagine HTML statiche non c'è bisogno di passare dal web server: possiamo direttamente passare al browser la loro posizione nel file system!

```
$ firefox $WORKSPACE_HOME/TemplateHTML/war/morepages/xxxxxxxxx.html
```

- Esempi di...
 - schema di una generica pagina html (*template.html*)
 - utilizzo degli anchor (*anchor.html*)
 - utilizzo della proprietà base (*base.html*)
 - form con metodi GET e POST (*form.html*)
 - redirect tra pagine (*redirect.html*)
 - markup testuale (*testo.html*), inline (*inline.html*), di titoli (*title.html*) e di paragrafi (*paragrafi.html*)
 - liste (*liste.html*) e tabelle (*tabelle.html*)
- E' molto più intelligente cercare riferimenti online per l'uso di tag, stili, eventi, ecc..
 - <http://www.w3schools.com>
 - <https://developer.mozilla.org>
 - <http://www.google.it>



HTML e l'uso degli stili

- Associare proprietà di formattazione a elementi del documento
 - definiti per mezzo dell'attributo **style** dei vari tag HTML...

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

- ...all'interno dell'elemento **head** del documento

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

- ...o in fogli di stile esterni

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

- Cascading: più livelli di definizione di stile possono applicarsi a ogni documento
 - user-agent → fogli di stile esterni → intestazione → stili contestuali ai tag
 - **le proprietà di stile si sommano, ma in caso di conflitto i livelli più interni sovrascrivono le proprietà di quelli più esterni**



Selettori

- Style definition format:

- head and external sheet:

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

- tag style attribute:

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

- Selections:

- tag name

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

- selector list

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

- DOM pattern

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

- class attribute

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

- id attribute

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

- attribute presence

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

- attribute values

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

- pseudo-classes

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

- pseudo-elements

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

- wildcard usage

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



Quale proprietà posso applicare a quale elemento

- E' possibile fare un sacco di cose con gli stili:
 - text *style*, dimension, **color**, font, alignment
 - background color, immagini
 - posizionamento spaziale, margini, bordi, padding
 - layout flow
- Gli stili possono operare (tra l'altro) in base alle seguenti metriche:
 - CSS keywords e proprietà specifiche (thin, thick, bolder, transparent, ..)
 - Misure del mondo reale (in, cm, mm, pt, ...)
 - Misure relative allo schermo (px, em, ex, %, ...)
 - Codici colore (#rrggbb, rgb(r,g,b), ...)
- Come orientarsi? Ancora una volta, il web!
(e.g., http://developer.mozilla.org/en/docs/CSS_Reference#Properties)
- Gli stili contano! Non è solo estetico, ma separazione di presentazione e contenuti!
@see <http://www.repubblica.it> senza CSS tramite firebug per avere un esempio di questo!



Esperimenti con gli stili

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](#)

Firebug permette di fare molte cose velocemente, ma il sorgente degli stili è accessibile a tutti! Basta seguire i link riportati nel sorgente delle pagine HTML!

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](#)



Alcuni approfondimenti

`/* 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: trasparente;
}
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;
}
```



Ancora approfondimenti

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
...

...ulteriori informazioni... ...ed ispirazione:

<http://www.csszengarden.com/tr/italiano/>

<http://css.html.it/>

<http://developer.mozilla.org/>



Javascript e il DOM

- Il DOM è un modo per rappresentare documenti XML (quindi anche l'HTML)
 - Moltissimi linguaggi di programmazione ne forniscono una implementazione (Java, Perl, PHP, Ruby, Python, ...)
 - Nel caso di Javascript, purtroppo L'IMPLEMENTAZIONE E' FORNITA DAL BROWSER SU CUI LO SCRIPT ESEGUE
 - problemi di cross-browser compatibility
 - anche in diverse versioni dello stesso browser
- Tuttavia...
 - esistono librerie di funzioni che semplificano la vita astruendo la diversità (es: jquery.....)
 - la situazione migliora continuamente (anche se gli utenti continuano a usare i vecchi browser!!)

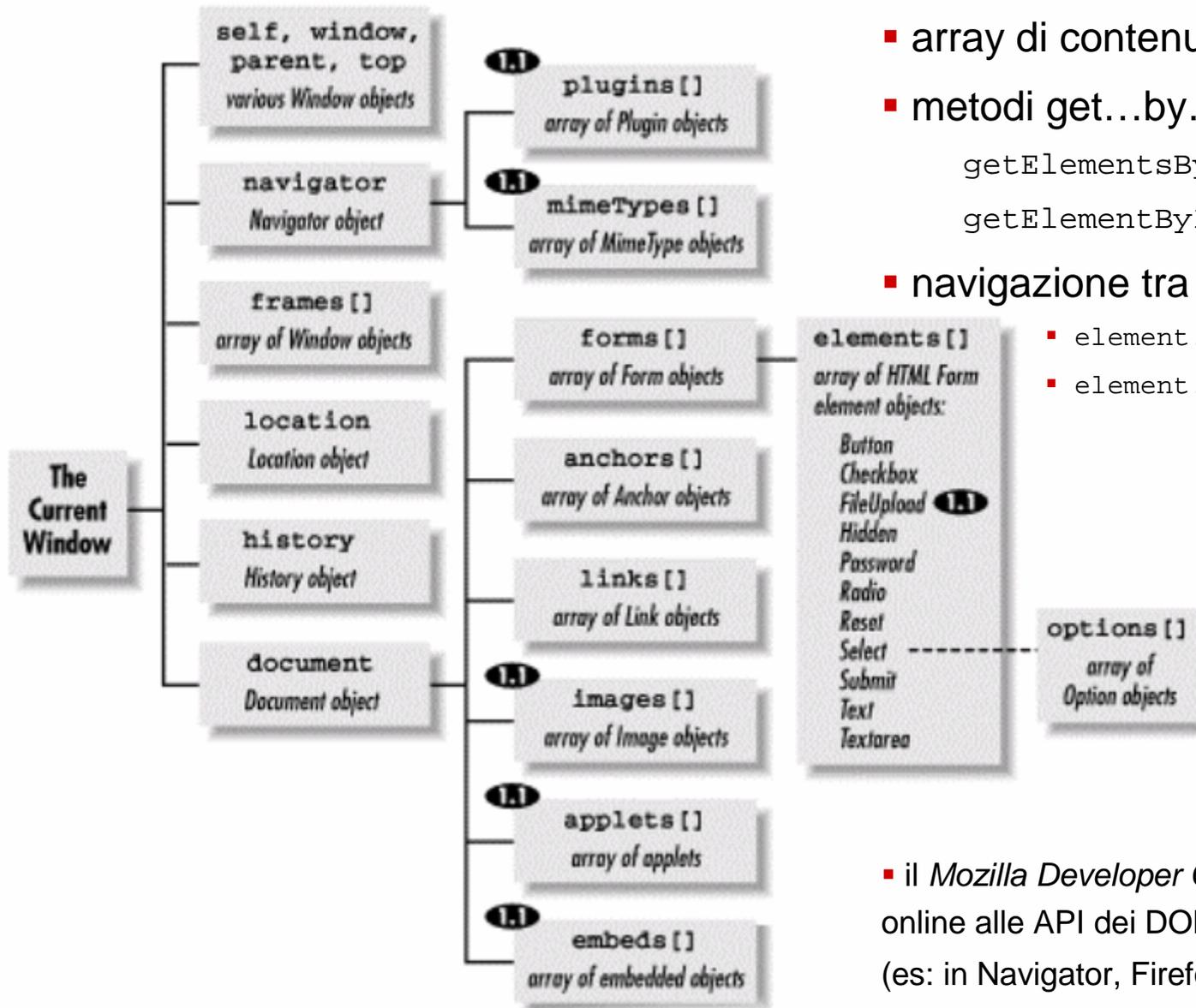


Giusto per avere una dimensione del problema...

	Win 98	Win 2000	Win XP	Mac 10.0	Mac 10.2	Mac 10.3	Mac 10.3.x	Mac 10.4
IE 7.0	n/a	n/a	A-grade	n/a	n/a	n/a	n/a	n/a
IE 6.0	A-grade	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a
IE 5.5	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a	n/a
IE 5.0	C-grade	C-grade	n/a	C-grade	C-grade	C-grade	C-grade	C-grade
Netscape 8.0	X-grade	X-grade	A-grade	n/a	n/a	n/a	n/a	n/a
Firefox 1.5	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Firefox 1.0.7	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Mozilla 1.7.12	X-grade	X-grade	A-grade	X-grade	X-grade	X-grade	X-grade	X-grade
Opera 8.5	X-grade	X-grade	A-grade	C-grade	C-grade	C-grade	X-grade	X-grade
Safari 1.0	n/a	n/a	n/a	X-grade	n/a	n/a	n/a	n/a
Safari 1.1	n/a	n/a	n/a	X-grade	X-grade	n/a	n/a	n/a
Safari 1.2	n/a	n/a	n/a	X-grade	X-grade	X-grade	n/a	n/a
Safari 1.3	n/a	n/a	n/a	n/a	n/a	X-grade	A-grade	n/a
Safari 2.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	A-grade



DOM: struttura generale



▪ Navigazione mediante

- array di contenuti indicizzati

- metodi get...by...

```
getElementsByTagName( tag_name )
```

```
getElementById( element_id )
```

- navigazione tra I nodi

- `element.parentNode`

- `element.appendChild(node)`

- il *Mozilla Developer Center* riporta una guida online alle API dei DOM implementati via Gecko (es: in Navigator, Firefox)\

Nel progetto di esempio...

- Semplici giochini tra Javascript, gli eventi DHTML, il DOM della pagina, gli stili
 - modifica di stili e del DOM, all'atto del caricamento di un documento
 - `window.onload`, definizione di una `function()` in linea, ricerca e rimozione di un nodo
 - aggiunta di stili in corrispondenza di un evento
 - roll over di immagini
 - DHTML + DOM
 - DHTML + DOM + JS
 - validazione di una form
 - `alert(msg)`
 - intervento sui div: rilocazione
 - cross-browser issues!
 - generazione di contenuto: mollica di pane



- **Firebug Inspect & Debug !!**



Il sito del corso

- E' un sito esattamente di questo tipo (risorse statiche, dinamicità lato client via javascript, DOM, CSS)
 - non gira nemmeno su Tomcat (non serve un Servlet Container! Non c'è codice Java... o meglio: non c'era un Servlet Container... e anche PHP era disattivato! Le uniche armi rimaste erano DHTML, CSS e Javascript)
 - seguendo tutti i link si può scaricare per intero
 - ispezione via firebug: come funziona....
 - ...comparsa e scomparsa di parti
 - ...il cambio al volo dello stile
 - **bugfix!** il link diretto a 'Materiale esercitazioni'...
 - ...fa sparire il resto della pagina
 - ...e riapparire la sezione desiderata
 - ...ma senza sottosezioni !!
 - si può correggere modificando sei caratteri ☺
 - oltre ovviamente a un milione di altri modi ☹

