



Università degli Studi di Bologna

Facoltà di Ingegneria

Tecnologie Web L-A
A.A. 2009 – 2010

Esercitazione 1

URI, HTTP, HTML, CSS

Tutor:

Ing. Pasini Samuele

samuele.pasini@unibo.it

Agenda

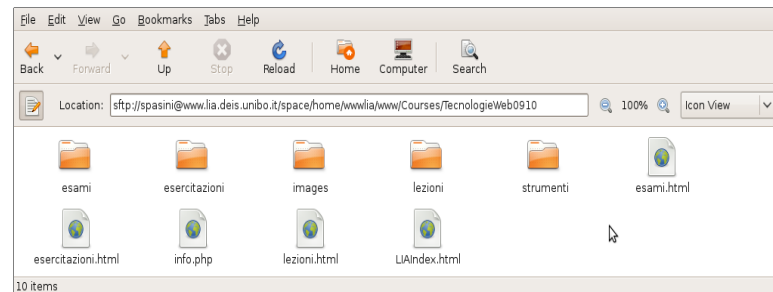
- URL
 - Prove ed esempi
- Protocollo HTTP
 - Scaricamento e utilizzo di un “monitor”
 - Analisi dei dati scambiati tramite HTTP REQUEST e HTTP RESPONSE
- HTML e CSS
 - Corrispondenza tra elementi visualizzati e codice sorgente
 - Strumenti di lavoro
 - “Ispezione” del contenuto di una pagina
 - “Ispezione” degli stili applicati agli elementi di una pagina
 - Esempi da ricreare

URL... non solo pagine Web

- Accesso a una risorsa via HTTP
 - <http://www.lia.deis.unibo.it/Courses/TecnologieWeb0910/esercitazioni.html>
- Scaricamento della stessa risorsa via FTP
 - <sftp://spasini@www.lia.deis.unibo.it/space/home/wwwlia/www/Courses/TecnologieWeb0910/esercitazioni.html>

...o anche...

- <ftp://ftp.iinet.net.au/debian/debian-cd/>
- Streaming di file multimediali
(possibili client: VLC, Windows Media Player, ...)
 - <mms://d74ki.rai.it/RedirectServlet?videoid=3223>
- Eccetera, eccetera...
 - http://en.wikipedia.org/wiki/URI_scheme



URL – Esempio #1

- Un tunnel TCP è un programma che “ascolta” su una specifica porta TCP dell'host su cui viene eseguito e...
 - inoltra tutti i dati in ingresso (es: HTTP REQUEST) a un ben definito endpoint remoto (HOST+PORT)
 - restituisce tutti i dati ottenuti in risposta dall'endpoint remoto (es: HTTP RESPONSE) al richiedente iniziale



- Possiamo utilizzarlo per “monitorare” il flusso di dati (**caratteri!**) che costituisce lo stream HTTP
 - basta **lanciare il tunnel sulla propria macchina...**
 - ...e **modificare adeguatamente l'URL richiamato dal browser al fine di incanalare richieste e risposte attraverso il tunnel!**

Un esempio di tunnel TCP

- Dopo aver “esploso” l'archivio ZIP dell'esercitazione, nel direttorio *tunnel*
 - una libreria Java, **soap.jar**, contiene l'implementazione del tunnel
 - uno script di avvio (**tunnel.sh** o **tunnel.bat**) manda in esecuzione il tunnel
- Il tunnel richiede come parametri
 - la porta su cui porsi in ascolto sulla macchina locale
es: **8081**
 - il nome (o l'indirizzo IP) della macchina remota a cui inoltrare le richieste
es: **www.lia.deis.unibo.it**
 - la porta TCP su cui è in ascolto il server remoto che ci interessa
es: **80** (default per i server Web)
- Infine, nel browser
 - **come deve essere modificato l'URL della home page del corso per osservare il traffico HTTP nel tunnel?**

Richieste e risposte HTTP

- L'interfaccia grafica del tunnel mostra il contenuto delle HTTP REQUEST e HTTP RESPONSE scambiate tra browser e server
 - quante e quali parti studiate nella teoria riuscite a riconoscere?
 - perché non una sola coppia di REQ+RESP, ma tante in successione?
 - riuscite a individuare le coppie corrispondenti?

From localhost:8081
From www.lia.deis.unibo.it:80

```

GET /Courses/TecnologieWeb0910/ HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.8) Gecko/20100215 Ubuntu/9.04 (jaunty) Firefox/3.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

```

```

HTTP/1.1 200 OK
Date: Thu, 25 Feb 2010 22:43:11 GMT
Server: Apache/2.2.3 (Unix) mod_jk/1.2.19
Last-Modified: Sat, 20 Feb 2010 11:42:06 GMT
ETag: "2dc59-3329-b07d3f80"
Accept-Ranges: bytes
Content-Length: 13097
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<LINK media=screen href="images/stile_main.css" type=text/css rel=stylesheet>
<LINK media=screen href="images/stile_colori.css" type=text/css rel=stylesheet>
<LINK media=print href="images/stile_stampa.css" type=text/css rel=stylesheet>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Tecnologie Web LA - AA 2009-2010</title>
</head>

<body>

<table border="0" width="100%" cellspacing="0" cellpadding="0" >
<tr>
<td width="5%" style="padding-left: 6px; padding-right: 6px">
<td width="5%" class="bgTestata">
<td width="90%" class="bgTestataUp">&nbsp;   </td>
</tr>
<tr><td></td><td colspan=2>
</tr>
</table>

```

```

GET /Courses/TecnologieWeb0910/images/stile_main.css HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.8) Gecko/20100215 Ubuntu/9.04 (jaunty) Firefox/3.0.1
Accept: text/css,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://localhost:8081/Courses/TecnologieWeb0910/

GET /Courses/TecnologieWeb0910/home_file/spacer.gif HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.8) Gecko/20100215 Ubuntu/9.04 (jaunty) Firefox/3.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://localhost:8081/Courses/TecnologieWeb0910/

GET /Courses/TecnologieWeb0910/images/stile_colori.css HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.8) Gecko/20100215 Ubuntu/9.04 (jaunty) Firefox/3.0.1
Accept: text/css,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://localhost:8081/Courses/TecnologieWeb0910/

```

Clear

Listening for connections on port 8081 ...

URL – Esempio #2

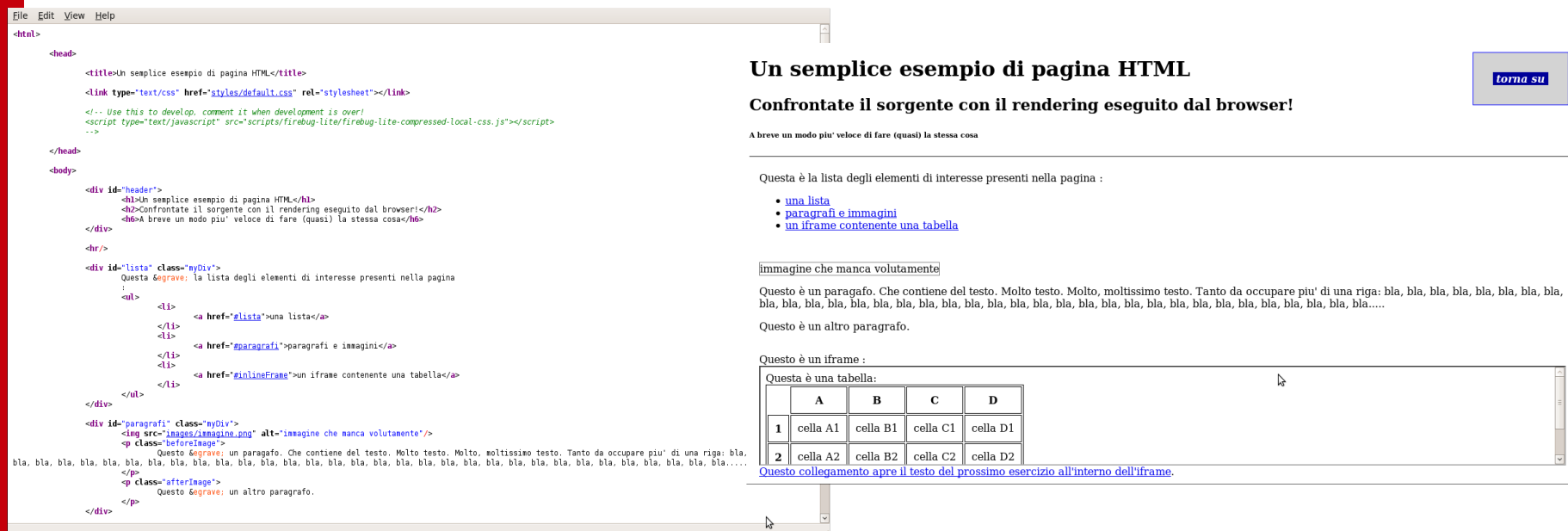
- Poiché non disponiamo ancora di un Web Server su cui esercitarci...
 - è necessario aprire le pagine HTML dell'esercitazione di oggi leggendole da file system
- E' possibile farlo “manualmente”...
 - tramite i menu a tendina del browser (File → Open → ...)
- Oppure nel modo (appena un po') piu' “*geek*”
 - digitando a mano l'URL adeguato nella barra degli indirizzi del browser

- Dopo aver “esploso” l'archivio ZIP dell'esercitazione, nel direttorio *html*
 - alcune semplici pagine HTML di esempio
- Per visualizzarle nel browser
 - **quale URL deve essere immesso nella barra degli indirizzi?**
 - attenzione a fare “escaping” dei caratteri speciali (*blank* = %20)

Editor HTML vs. User-agent

- Confrontiamo il “sorgente” HTML della pagina e la sua versione “renderizzata”

- via editor testuale (es: *notepad.exe*, *gedit*, *kedit*, ...), IDE (*Eclipse*, *JDeveloper*, *NetBeans*, ...), o direttamente dal browser (→ visualizza sorgente..)
- vs.
- user-agent (es: *Microsoft Internet Explorer*, *Mozilla Firefox*, ... ma non solo: *Opera*, *Safari*, *Chrome*, *Epiphany*,)



Strumenti di lavoro

- **Firebug** è un'estensione di Mozilla Firefox (e da poco, anche di Google Chrome) che permette il **debug**, la **modifica al volo** e il **monitoraggio** di tutti gli aspetti di una pagina web, come i fogli di stile, il codice HTML, la struttura DOM e il codice JavaScript.
 - la sua installazione richiede l'accesso al catalogo on-line degli add-on, lo scaricamento e la verifica dell'autenticità del corrispondente plugin
 - oppure, in caso la rete non sia accessibile (come dal Lab4), l'“*hacking*” della propria installazione locale di Firefox (istruzioni e software scaricabili dal sito di Tecnologie Web LA 08/09)

L'installazione di Firebug è vivamente consigliata, ma richiede un tempo che non ci è possibile sottrarre al laboratorio. Siete quindi invitati a eseguire l'installazione, ma per oggi...

- **Firebug-lite** è una versione leggera di Firebug (niente debug Javascript, né funzionalità avanzate) sviluppata interamente in Javascript, che può essere “caricata” su qualsiasi pagina HTML
 - aggiungendo il riferimento al corrispondente **script**, se è possibile modificare il sorgente della pagina
 - inserendo una **bookmarklet** (URL che invoca codice Javascript) nella barra degli indirizzi del browser, dopo aver caricato la pagina da analizzare, se non è possibile modificarne il sorgente

Tuttavia...

- Anche in questo caso è in teoria necessario l'accesso a Internet, per raggiungere le URL dello script o della bookmarklet
- In alternativa, è possibile scaricare tutte le risorse relative al codice di firebug-lite (script js, stili css, immagini) e modificare i riferimenti assoluti presenti affinché diventino riferimenti relativi, quindi far puntare script o bookmarklet alla versione “locale” modificata

- Le pagine HTML dell'esercitazione sono già predisposte per utilizzare una versione “locale” di firebug-lite, opportunamente modificata
- **Abilitare *firebug-lite*** rimuovendo il commento che né impedisce il caricamento dello script (o della bookmarklet)
 - uso della funzione **inspect**
 - aggiunta di **stili** “on-the-fly”
- Elementi di interesse
 - **foglio di stile** esterno: *styles/default.css*
 - anchor che utilizzano **query fragment** (che il browser concatena all'URL corrente) per puntare a specifici elementi della loro stessa pagina
 - **iframe** per includere codice HTML “esterno” (oggi tipicamente usati per permettere applicazioni cross-site → es: Google Maps!)
 - **meta attributo** per forzare la redirectione (ma spesso anche il refresh, nei siti reali) verso un'altra pagina, dopo un certo intervallo di tempo

Can you do that?

- Ottenere il seguente risultato:

- aggiungendo gli opportuni stili in un secondo file CSS
- modificando il sorgente HTML solo per includere tale file, inserire gli elementi della form e ottenere l'unione di alcune celle della tabella

Un semplice esempio di pagina HTML

Confrontate il sorgente con il rendering eseguito dal browser!

una lista

paragrafi e immagini

un iframe contenente una tabella

Questa è la lista degli elementi di interesse presenti nella pagina :

This area contains two paragraphs.

Questo è un paragrafo. Che contiene del testo. Molto testo. Molto, moltissimo testo. Tanto da occupare piu' di una riga: bla, bla.....

Questo è un altro paragrafo.

immagine che manca volutamente

Questo è un iframes :

Questa è una tabella, le cui celle consentono di inviare dati tramite una form:

A	B	C	D
1 cella A1	cella B1	cella C1	cella D1
2 cella A2	contenuto presente all'interno della cella B2		contenuto presente all'interno della cella D2
3 cella A3	cella B3	cella C3	

Una form HTML richiede un pulsante di invio!