



**Università degli Studi di Bologna**  
**Facoltà di Ingegneria**

*Tecnologie Web T*  
*A.A. 2019 – 2020*

**Esercitazione 1**  
**URI, HTTP, HTML, CSS**

1

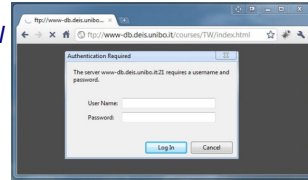
**Agenda**

- URL e Protocollo HTTP
  - prove ed esempi
- HTML e CSS
  - corrispondenza tra elementi visualizzati e codice sorgente
  - “ispezione” del contenuto di una pagina
  - “ispezione” degli stili applicati agli elementi di una pagina
  - esempi da ricreare

2

## URL... non solo pagine Web

- Accesso a una risorsa via HTTP
  - <http://www-db.disi.unibo.it/courses/TW/index.html>
- Scaricamento della stessa risorsa via FTP
  - <ftp://www-db.disi.unibo.it/courses/TW/index.html>
  - ...o anche...
  - <ftp://ftp.iinet.net.au/debian/debian-cd/>
- Streaming di file multimediali  
(possibili client: VLC, Windows Media Player, ...)
  - <mms://151.1.245.36/rtl102.5lq>
- Eccetera, eccetera...
  - [http://en.wikipedia.org/wiki/URI\\_scheme](http://en.wikipedia.org/wiki/URI_scheme)



## URL e pagine HTML

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

## Ispezione di codice HTML esistente

- Dopo aver estratto i file presenti nell'archivio “01\_TecWeb.zip” dell'esercitazione, nelle directory *html1* e *html2* trovate
  - 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)

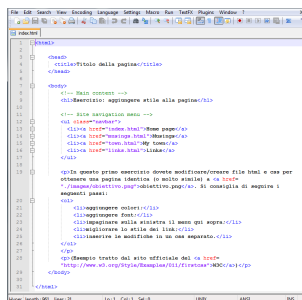
Esercitazione 1

5

5

## Editor HTML vs. User-agent

- Confrontiamo il “sorgente” HTML della pagina e la sua versione “renderizzata”
  - via editor testuale (es: *notepad.exe*, *notepad++*, *gedit*, *kedit*, ...)
  - IDE (*Eclipse*, *JDeveloper*, *NetBeans*, ...)
  - direttamente dal browser (→ visualizza sorgente...)
- vs.
- user-agent (es: *Microsoft Internet Explorer*, *Mozilla Firefox*, *Opera*, *Safari*, *Google Chrome*, *Epiphany*, ...)



```
1 <!-- HTML -->
2 <html>
3 <!-- titolo della pagina -->
4 <title>
5
6 </title>
7 <!-- Main content -->
8 <!-- descrizione: aggiungere stile alla pagina -->
9
10 <!-- fine dell'area -->
11 </div>
12 <!-- footer -->
13 <div class="foot">
14 <div class="messaggi">
15 <div class="messaggi">
16 <div class="messaggi">
17 </div>
18 </div>
19
20 <!-- questo primo esercizio dovrebbe modificare/montare il file html e non per
21 ottenere una pagina identica in tutto e per tutto -->
22 <!-- esempio tratto dal sito ufficiale del W3C -->
23 <div class="messaggi">
24 <div class="messaggi">
25 <div class="messaggi">
26 <div class="messaggi">
27 </div>
28 </div>
29 </div>
30 </div>
```

### Esercizio: aggiungere stile alla pagina

- [Home page](#)
- [Musings](#)
- [My town](#)
- [Links](#)

In questo primo esercizio dovete modificare/creare file html e css per ottenere una pagina identica (o molto simile) a [obiettivo.png](#). Si consiglia di seguire i seguenti passi:

1. aggiungere colori;
2. aggiungere font;
3. impaginare sulla sinistra il menu qui sopra;
4. migliorare lo stile dei link;
5. inserire le modifiche in un css separato.

(Esempio tratto dal sito ufficiale del W3C)

Esercitazione 1

6

6

## *index.html*

- Utilizzare **Google Chrome Developer Tools** o **Firebug** di **Firefox** nella sua versione **Portable** (disponibile dal sito web del corso) per analizzare/modificare i file
  - uso della funzione inspect
  - aggiunta di stili “on-the-fly”
- In alternativa, utilizzare **Firebug-Lite** (compatibile per ogni browser)
  - se non siete collegati a Internet l’add-on Firebug-Lite non è accessibile; è comunque 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 ne impedisce il caricamento dello script (o della bookmarklet)
- **Elementi di interesse**
  - **foglio di stile** esterno: `styles/default.css`
  - **ancore interne** (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)

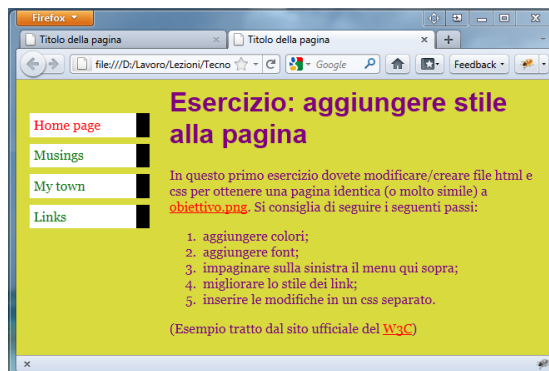
Esercitazione 1

7

7

## Can you do that? (html1)

- **Ottenere il seguente risultato:**
  - aggiungere colori
  - aggiungere font
  - impaginare sulla sinistra il menu qui sopra
  - migliorare lo stile dei link
  - inserire le modifiche in un file CSS separato



Esercitazione 1

8

8

## Can you do that? (html2)

- Ottenere il seguente risultato:

- aggiungere gli opportuni stili in un secondo file CSS
- modificare 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!**

[illegible]

### Esercitazione 1

9

9

## APPENDICE

**(altri esempi html/css da cui apprendere...)**

10

## Esempio “Lisa Simpson”: formattazione tipografica

- “01\_TecWeb.zip”, directory *esempi/Simpson*



Esercitazione 1

11

11

## Esempio “PosizionamentoSenzaFrame”: layout liquidi

- “01\_TecWeb.zip”, directory *“esempi/PosizionamentoSenzaFrame”*



Esercitazione 1

12

12

## Esempio HTML 5 CSS 3: “Le griglie”

- “01\_TecWeb.zip”, directory `esempi/HTML5CSS3/grid.html`



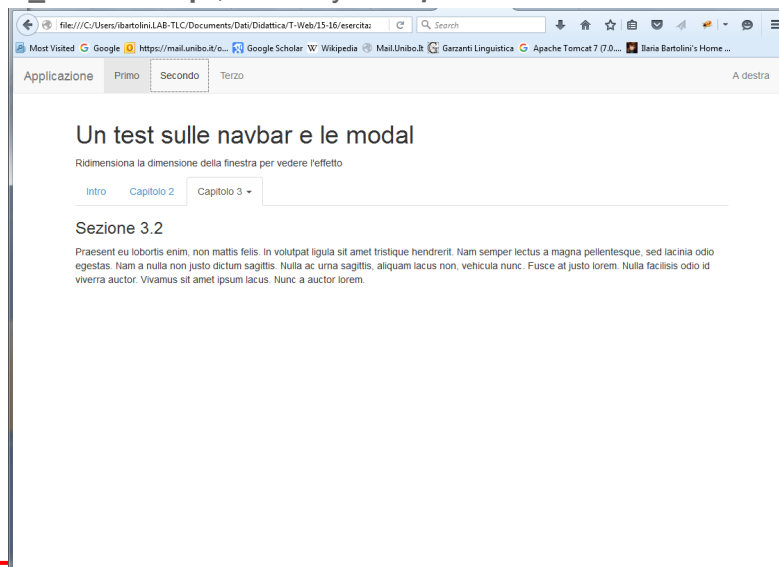
Esercitazione 1

13

13

## Esempio HTML 5 CSS 3: “Navbar e modal”

- “01\_TecWeb.zip”, directory `esempi/HTML5CSS3/navbar.html`



Esercitazione 1

14

14

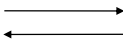
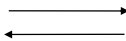
## APPENDICE

### (implementazione di un tunnel...)

15

### Analisi dati scambiati tramite HTTP REQ/RESP via tunnel

- 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**

Esercitazione 1

16

16

## Un esempio di tunnel TCP

- Dopo aver estratto i file presenti nell'archivio **"01\_TecWeb.zip"** dell'esercitazione, nella directory **tunnel** trovate
  - una libreria Java **soap.jar**, contenente l'implementazione del tunnel
  - uno script di avvio **tunnel.sh** o **tunnel.bat**, che 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-db.disi.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 dei corsi del LIA per osservare il traffico HTTP nel tunnel?**

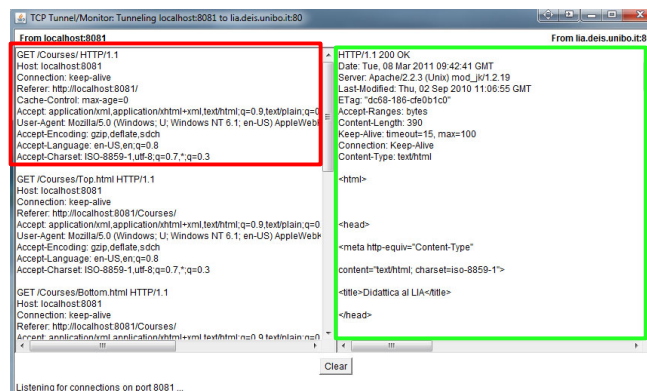
Esercitazione 1

17

17

## 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?**



Esercitazione 1

18

18