



# Programmazione WEB Avanzata per la Fisica Sperimentale

# WEB



Strumenti per il trattamento  
dei dati client-side al tempo di HTML5

Le trasparenze del corso  
sono disponibili qui:

[http://hal9000.mib.infn.it/~menasce/CORSO\\_Specialistico](http://hal9000.mib.infn.it/~menasce/CORSO_Specialistico)

# A chi è indirizzato il corso

- A ricercatori, tecnologi e studenti in generale
- In particolare a chi ha necessità di creare interfacce Web a programmi (DAQ, Monitor, Analisi, Amministrazione...)
- Non si assume alcuna conoscenza pregressa di HTML o di tecnologie Web in particolare
- Ma si presuppone una certa conoscenza di programmi compilati (C++) e di programmi interpretati (scripting, ad esempio Perl)
- La filosofia generale è “***non reinventiamo l’acqua calda***”
  - Molti già posseggono programmi a loro utili che vorrebbero fossero collegati ad un’interfaccia WEB.
  - Spesso non è pratico o possibile riscriverli ma solo modificarli
  - Nel corso mi concentro su programmi in ***perl***, semplicemente perché didatticamente più semplici da illustrare

- Impareremo a scrivere opportune interfacce WEB specificatamente per questa tipologia di programma, in una prima fase.
- Vedremo poi, in una seconda parte, come sia possibile un approccio più semplice nel caso fosse possibile scrivere tutto il progetto da zero, utilizzando sia per il client che per il server, solamente Javascript.
- Il corso è strutturato attorno ad un certo numero di esempi pratici, di complessità crescente, partendo da una semplice pagina web statica sino ad una complessa interfaccia ad un sistema di acquisizione dati da sensore basato su un'App installata su uno smart phone.
- Tutti gli esempi saranno disponibili nella seguente area:

## Breve storia della tecnologia WWW

- Nasce al CERN da un progetto personale di Tim Berners Lee nel 1989
- L'idea alla base è quella dello hyper-text
  - ✓ Una singola parola/frase possono essere utilizzate come riferimento diretto ad un altro documento (in qualsiasi altro computer nel mondo)
  - ✓ Separazione concettuale tra **contenuto** di un documento e istruzioni per la sua **rappresentazione** (HyperText Markup Language, HTML)
- Idee che a loro volta si basano su concetti di alto livello come il *Document Object Model* (DOM) e lo *Extensible Markup Language* (XML) che sono di uso comune ormai nel mondo delle applicazioni web



# Introduzione

Ad oggi, però, una pagina web non contiene solo hyperlinks ad altri documenti (testuali, audio o video che siano), ma può essere una complessa interfaccia dinamica il cui contenuto può variare anche solo in una sua parte in seguito ad un'azione intrapresa da un utente.

Vediamo una serie di esempi, partendo da pagine web semplici, direi triviali, sino a pagine particolarmente accattivanti e complesse, dotate di un altissimo livello di interattività:



# Perchè questo corso?

La fisica sperimentale delle Alte Energie è caratterizzata (ma non è una sua esclusiva), dall'utilizzo di apparecchiature molto complesse la cui gestione è affidata ad un gran numero di esperti che le deve gestire e monitorare in tempo reale e contemporaneamente tra loro, spesso in luoghi tra loro molto distanti fisicamente. Per farlo, il modo più efficiente è quello di utilizzare un browser come una semplice interfaccia grafica remota all'apparecchio in questione.

Maggiore è la complessità di ciò che l'interfaccia deve permettere all'utente di fare, maggiore deve essere la cura nel fornire uno strumento semplice nell'aspetto ma corrispondentemente complesso al suo interno (la complessità, se esiste in partenza, può solo essere spostata altrove, non cancellata...)

# Perchè questo corso?

Nel caso dei grandi esperimenti, in genere le collaborazioni trovano al loro interno le necessarie risorse di man-power per affrontare e risolvere questo tipo di problematiche (anche se non sempre nel modo più efficace ed efficiente...)

Diverso il caso di collaborazioni piccole o addirittura di singoli individui che abbiano la necessità di rendere disponibile la funzionalità del proprio codice ad altri usando **http** come protocollo di comunicazione.

Scopo di questo corso è di fornire una panoramica di ciò che è possibile realizzare in questo campo, cosa è necessario sapere ed imparare e il tutto viene fornito tramite esempi ragionevolmente compatti, che i partecipanti al corso avranno modo di provare.

# Perchè questo corso?

La problematica di fornire accesso a programmi complessi tramite interfacce che ne semplifichino la gestione da parte dell'utente non è peculiare solo della fisica e della relativa strumentazione utilizzata.

Esempi significativi sono la navigazione all'interno di un database, la presentazione grafica di risultati scientifici, la gestione di sistemi di *accounting* e *monitoring* in vari campi (dai rivelatori sino all'uso di risorse di calcolo offerte in un pool di tipo Grid o Cloud).

Conseguenza di ciò è che la platea di persone potenzialmente interessate allo sviluppo di questi strumenti si sta allargando più velocemente della capacità e possibilità individuale di apprendere le tecniche necessarie a farlo. Ecco quindi l'idea di un corso rivolto a questa platea e sviluppato mediante semplici esempi pratici.

- **Apache**: il motore di un server web
- Il protocollo **http**
- Il protocollo **html**, con una serie di esempi, da semplici (pagine statiche) sino a moderatamente complessi (pagine con FORM di input per dati), con cenni sul protocollo **CSS** per il rendering
- Lo scripting language **JavaScript**: cos'è, perché si usa ed esempi relativi
- La tecnologia **Ajax**: il moderno modo di modificare il contenuto di una pagina web senza dover ogni volta ricaricarne il contenuto da remoto
- Cenni al protocollo **REST**
- Introduzione ai moderni **frameworks** di **JavaScript**, con particolare enfasi a **ExtJs** (con relativi esempi ed esercizi)
- Cenni a **DOM, XML, JSON, SVG, HTML5, WebGL** ed altro ancora

# Architettura

Molto schematicamente il Web si articola su tre componenti:

- Dei server
- Una rete telematica
- Dei client (dei browser)



- **1930:** **Vannevar Bush** espone l'idea di un sistema di collegamenti virtuali tra documenti (microfilm) nel progetto **Memex**, ma la tecnologia dell'epoca non era in grado di implementarla
- **1965:** **Ted Nelson** propone il concetto di **HyperText** nell'ambito del progetto **Xanadu**, una possibile base concettuale per l'idea di Vannevar Bush, ma ancora lontana da una possibile implementazione tecnologicamente sostenibile e praticabile.
- **1989:** **Tim Berners-Lee** definisce il protocollo **http** ed **html** e propone la loro implementazione nel WorldWideWeb.

Da quella data esplode una serie impressionante di iniziative volte a definire protocolli e standard, la cui gestione è oggi demandata al W3 Consortium

Un web server è un computer, connesso alla rete tramite protocollo TCP/IP, dotato di un sistema operativo (tipicamente Linux, ma non solo), nel quale è attivo un processo in *background* (un *daemon*) che accetta richieste di connessione dalla porta 80 (di *default*). Il *daemon* più diffuso oggi è Apache, un prodotto Open Source.

Vediamo come fare a manipolare il comportamento del web server sulla macchina virtuale che vi metteremo a disposizione per gli esercizi.

Anzitutto occorre connettersi in qualità di **root** per avere i privilegi per configurare il server Apache.

Poi vedremo dove stanno i file che determinano la configurazione di Apache, faremo partire il server e proveremo a connetterci da remoto tramite un browser.

Ognuno di voi avrà a disposizione una macchina virtuale personale, col seguente indirizzo IP:

192.168.168.**XX**

dove **XX** sarà un numero progressivo da 01 in avanti che vi verrà assegnato individualmente.

Per connettersi alla propria macchina virtuale, basta dare, dalla propria shell, il comando:

```
> ssh -XY root@192.168.XX
```

```
root@webber23:~ - ssh
Oct 08 12:18:07 webber23.mi.infn.it systemd[1]: Stopping The Apache HTTP Server...
Oct 08 12:18:08 webber23.mi.infn.it systemd[1]: Stopped The Apache HTTP Server.
Oct 08 12:27:43 webber23.mi.infn.it systemd[1]: Stopped The Apache HTTP Server.
You have new mail in /var/spool/mail/root
[root@webber23 ~]# systemctl start httpd
[root@webber23 ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
   Active: active (running) since Thu 2015-10-08 13:37:11 CEST; 1min 16s ago
     Main PID: 32197 (httpd)
        Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
      CGroup: /system.slice/httpd.service
              ├─32197 /usr/sbin/httpd -DFOREGROUND
              ├─32198 /usr/sbin/httpd -DFOREGROUND
              ├─32199 /usr/sbin/httpd -DFOREGROUND
              ├─32200 /usr/sbin/httpd -DFOREGROUND
              ├─32201 /usr/sbin/httpd -DFOREGROUND
              ├─32202 /usr/sbin/httpd -DFOREGROUND
              └─32203 /usr/sbin/httpd -DFOREGROUND

Oct 08 13:37:11 webber23.mi.infn.it systemd[1]: Starting The Apache HTTP Server...
Oct 08 13:37:11 webber23.mi.infn.it systemd[1]: Started The Apache HTTP Server.
[root@webber23 ~]#
```

```
[root@webber23 ~]# ps aux | grep httpd
root      32197  0.0  1.6 242480  8492 ?          Ss   13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32198  0.0  0.7 244492  3524 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32199  0.0  0.7 244564  3804 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32200  0.0  0.7 244564  3804 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32201  0.0  0.7 244564  3804 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32202  0.0  0.7 244564  3804 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
apache    32203  0.0  0.7 244564  3804 ?          S    13:37  0:00 /usr/sbin/httpd -DFOREGROUND
root      32215  0.0  0.1 112640   936 pts/0        R+   13:40  0:00 grep --color=auto httpd
[root@webber23 ~]#
```

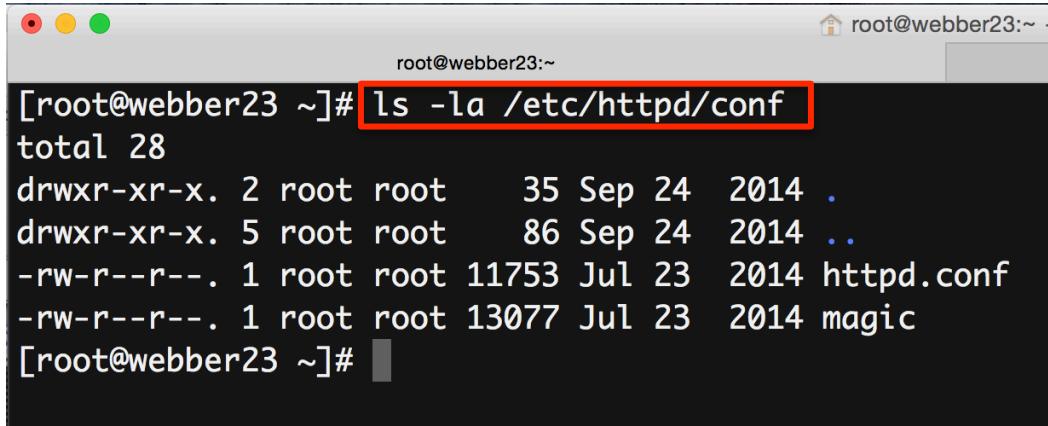
```
root@webber23:~ - ssh
root@webber23:~#
[root@webber23 ~]# systemctl stop firewalld
[redacted]

[root@webber23 ~] setenforce 0
```

# Il file di configurazione

Il processo *daemon* che accetta connessioni (richieste) dalla porta 80 si chiama **httpd** e, come abbiamo visto, lo si attiva istanziandolo come un servizio

Esso agisce in base a delle specifiche che l'amministratore di sistema (**root**) definisce in un opportuno file di configurazione, **httpd.conf**, nel quale indica cosa può fare chi e come nel contattare **httpd** via rete.



```
[root@webber23 ~]# ls -la /etc/httpd/conf
total 28
drwxr-xr-x. 2 root root    35 Sep 24  2014 .
drwxr-xr-x. 5 root root   86 Sep 24  2014 ..
-rw-r--r--. 1 root root 11753 Jul 23 2014 httpd.conf
-rw-r--r--. 1 root root 13077 Jul 23 2014 magic
[root@webber23 ~]#
```

INFN Istituto Nazionale di Fisica Nucleare

# Diamogli un'occhiata

CMS

File Edit Search Preferences Shell Macro Windows Help

```

# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.

# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.

# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', whereas '/log/access_log' will be
# interpreted as '/log/access_log'.

# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

```

# Dove stanno i files del server?

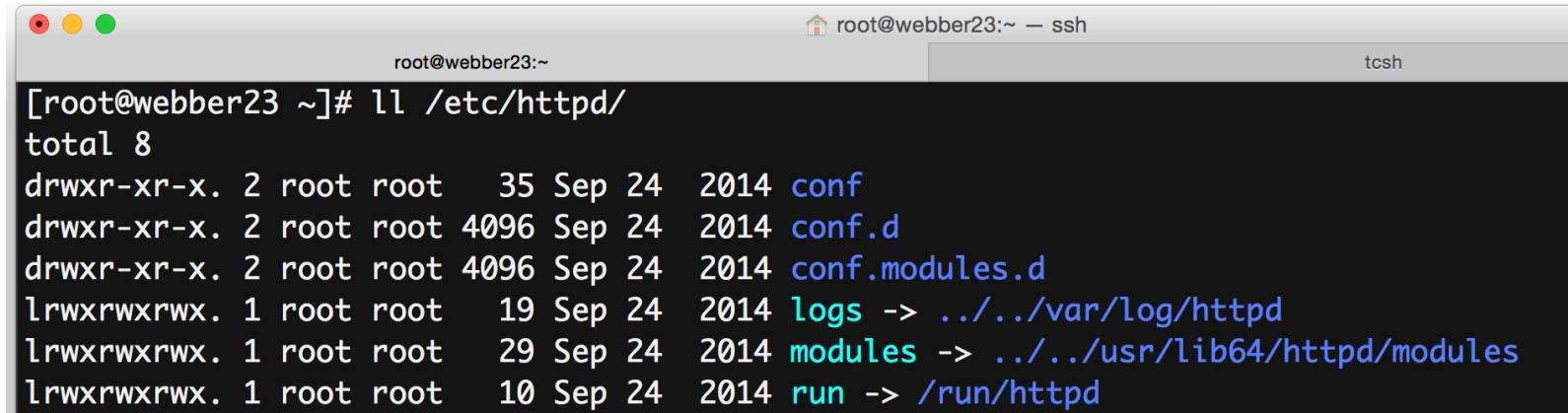
Tra le prime cose che l'amministratore di sistema può definire in questo file di configurazione vi è il **path** assoluto in cui sono disponibili i files maneggiati da **httpd** stesso, quali il file di configurazione, i messaggi di errore e i log di connessione

```

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs/2.2/mod/mpm_common.html#lockfile>
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/httpd"

```

/etc/httpd/conf/httpd.conf



root@webber23:~ — ssh

```
[root@webber23 ~]# ll /etc/httpd/
total 8
drwxr-xr-x. 2 root root 35 Sep 24 2014 conf
drwxr-xr-x. 2 root root 4096 Sep 24 2014 conf.d
drwxr-xr-x. 2 root root 4096 Sep 24 2014 conf.modules.d
lrwxrwxrwx. 1 root root 19 Sep 24 2014 logs -> ../../var/log/httpd
lrwxrwxrwx. 1 root root 29 Sep 24 2014 modules -> ../../usr/lib64/httpd/modules
lrwxrwxrwx. 1 root root 10 Sep 24 2014 run -> /run/httpd
```

Un server web, in funzione della sua popolarità, può ricevere un numero di connessioni per unità di tempo che va da pochissime fino ad un numero ingestibile (Denial Of Service, DoS): in questo file di configurazione, **httpd.conf**, l'amministratore di sistema può definire quanti processi paralleli ci possono essere in ascolto sulla porta 80, e quante richieste contemporanee può gestire ognuno di essi (anche se non sono queste le soluzioni per ovviare a problemi DoS)

```
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# ServerLimit: maximum value for MaxClients for the lifetime of the server
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule prefork.c>
StartServers      8
MinSpareServers  5
MaxSpareServers  20
ServerLimit      256
MaxClients       256
MaxRequestsPerChild 4000
</IfModule>
```

**/etc/httpd/conf/httpd.conf**

La porta 80 è la porta storicamente dedicata ai web server: un amministratore di sistema può decidere di mettere il *daemon* in ascolto su una porta differente:

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
```

</etc/httpd/conf/httpd.conf>

Un *web server* può (potenzialmente) scrivere dati nel *file system* locale in seguito a specifiche richieste di un utente: l'amministratore può decidere a nome di chi debba agire **httpd** nel fare ciò (con ovvie ripercussioni sui relativi privilegi di R/W).

```
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".
# . On HPUX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User apache
Group apache
```

</etc/httpd/conf/httpd.conf>

# Accesso al file system locale

Quando un web server riceve una richiesta per produrre un documento in risposta, deve sapere dove andare a cercarlo nel *file system* locale:

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"
```

/etc/httpd/conf/httpd.conf

La specifica di cui sopra serve a far sì che, se il *daemon* **httpd** risponde sulla porta 80 del nodo **192.168.168.xx**, un indirizzo web (**URL**, Uniform Resource Locator) del tipo **http://192.168.168.xx/page.html** corrisponderà alla richiesta di ottenere il file

**/var/www/html/page.html** sul file system di **192.168.168.xx**

Uno **URL** del tipo **http://192.168.168.xx**, nel quale non è specificato alcun tipo di documento, viene risolto da **httpd** cercando un file chiamato **index.html** nella directory **/var/www/html**

# Una prima pagina Web

In un sistema operativo installato ex-novo, la zona predisposta per i documenti non contiene nulla:

```
[root@webber23 ~]# ls -la /var/www/html
total 0
drwxr-xr-x. 2 root root 6 Jul 23 2014 .
drwxr-xr-x. 4 root root 31 Jul 23 2014 ..
[root@webber23 ~]#
```

Ecco allora che ci converrà definire un'opportuna pagina di benvenuto nel nostro web server, pagina che sarà quella di default per chi si collega al sito con uno URL breve.



The screenshot shows a window titled "index.html - /var/www/html/" with a menu bar including File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The main text area contains the following HTML code:

```
<html>
<title>
Welcome to the 192.168.168.23 web server
</title>
Welcome to the 192.168.168.23 web server DEFAULT home page!
</html>
```

# Una prima connessione http

```
Terminal — tcsh
root@webber23:/var/www/html          tcsh
[macdario:~] menasce% telnet 192.168.168.23 80
Trying 192.168.168.23...
Connected to 192.168.168.23.
Escape character is '^]'.
GET
$<html>

<title>
  Welcome to the 192.168.168.23 web server
</title>

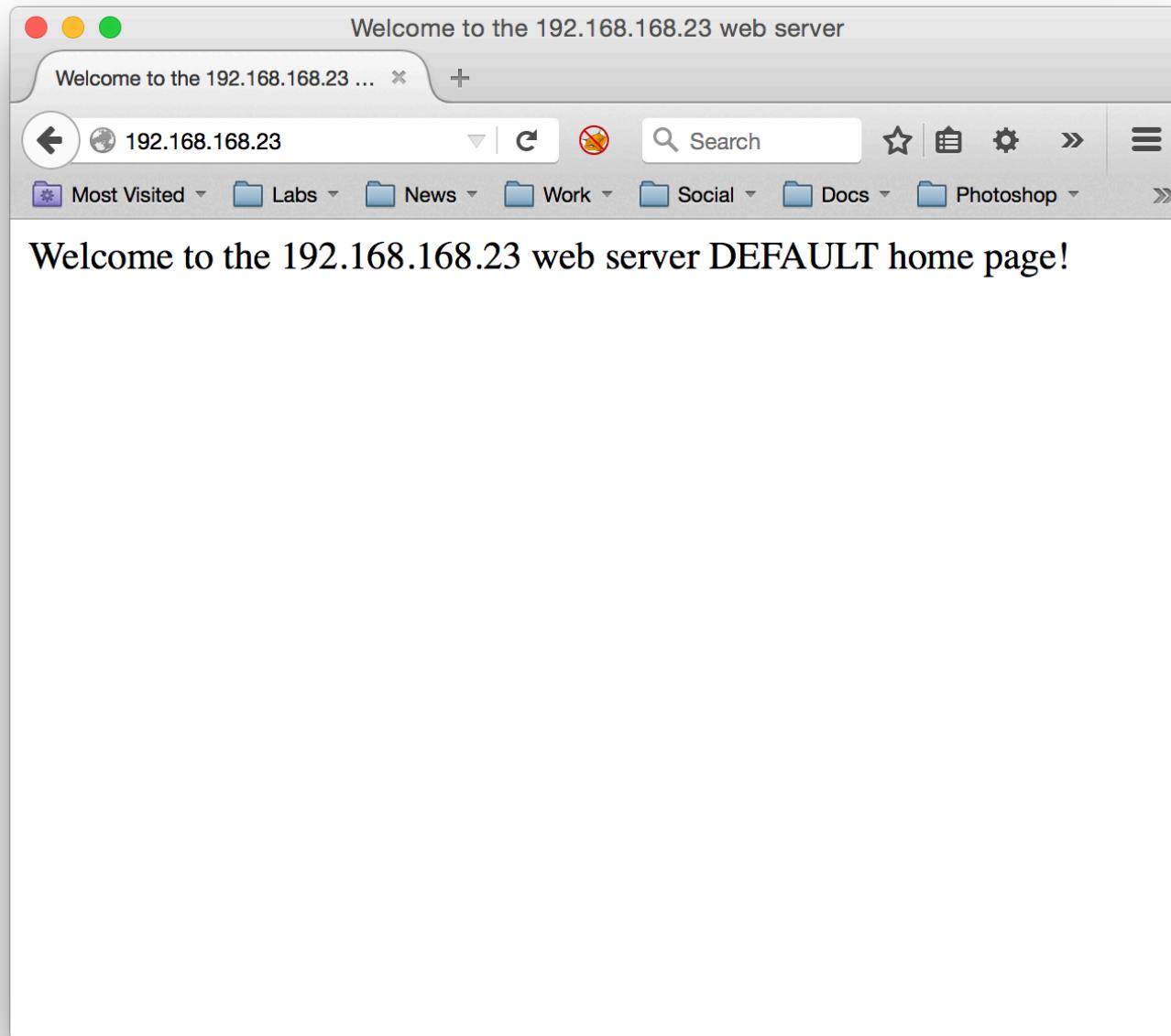
Welcome to the 192.168.168.23 web server DEFAULT home page!

</html>

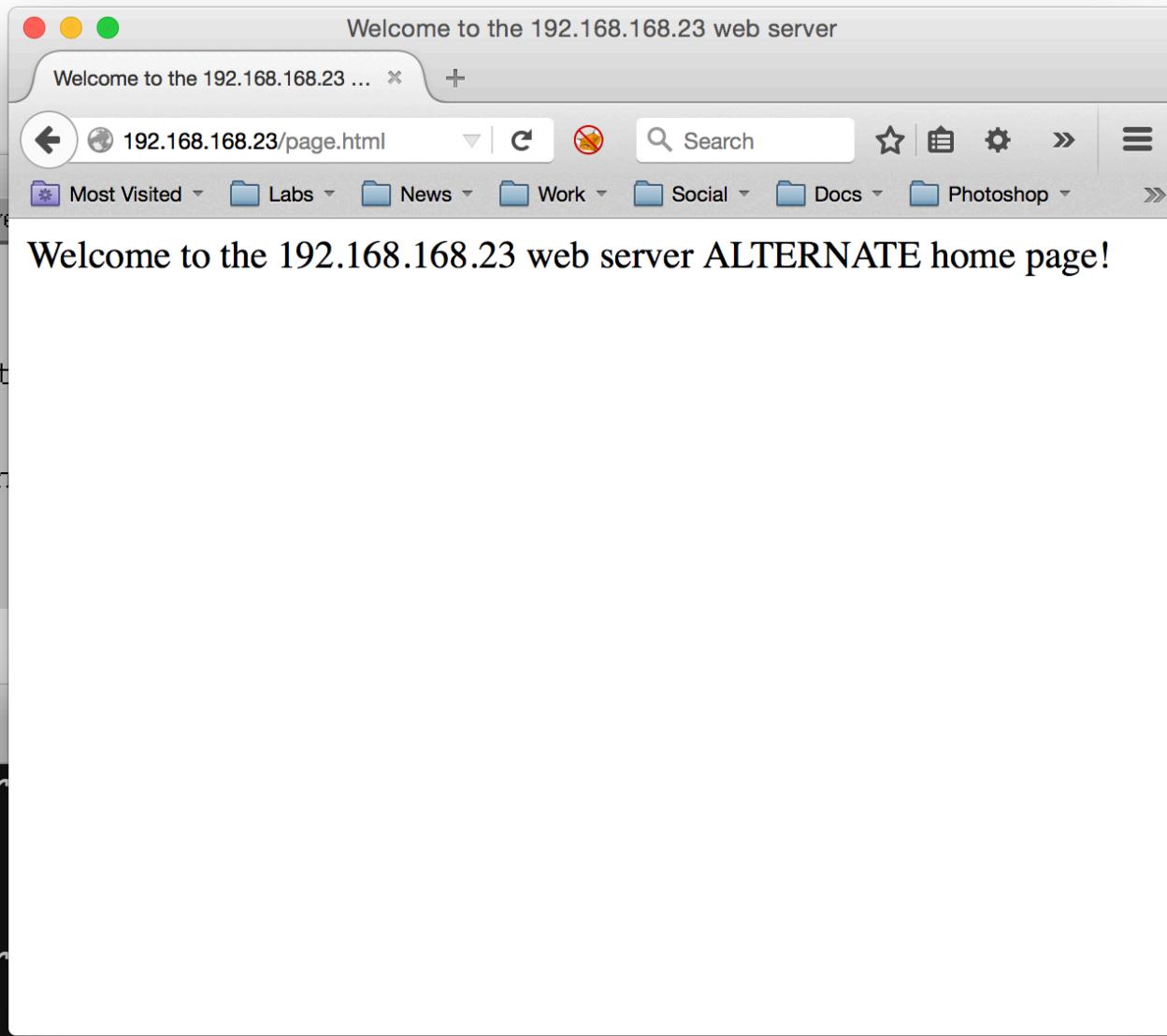
Connection closed by foreign host.
[macdario:~] menasce%
```

Non abbiamo ancora finito di esaminare in dettaglio il contenuto di **httpd.conf**, ma già così possiamo mettere in ascolto il server. Contattiamo il *daemon* di **httpd** sul server tramite **telnet** sulla porta 80:

# Passiamo al browser



# Pagine alternative...



Welcome to the 192.168.168.23 web server

Welcome to the 192.168.168.23 ...

192.168.168.23/page.html

File Edit Search Pre

```
<html>

<title>
Welcome to t
</title>

Welcome to th

</html>
```

[root@webber total 8 -rw-r--r--. -rw-r--r--. [root@webber]

- Il comando Linux **service**:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/System\\_Administrators\\_Guide/sect-Managing\\_Services\\_with\\_systemd-Services.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sect-Managing_Services_with_systemd-Services.html)

- Il Web server **httpd**:

<http://httpd.apache.org/>

- Argomenti di storia di **Internet** e del **WWW**:

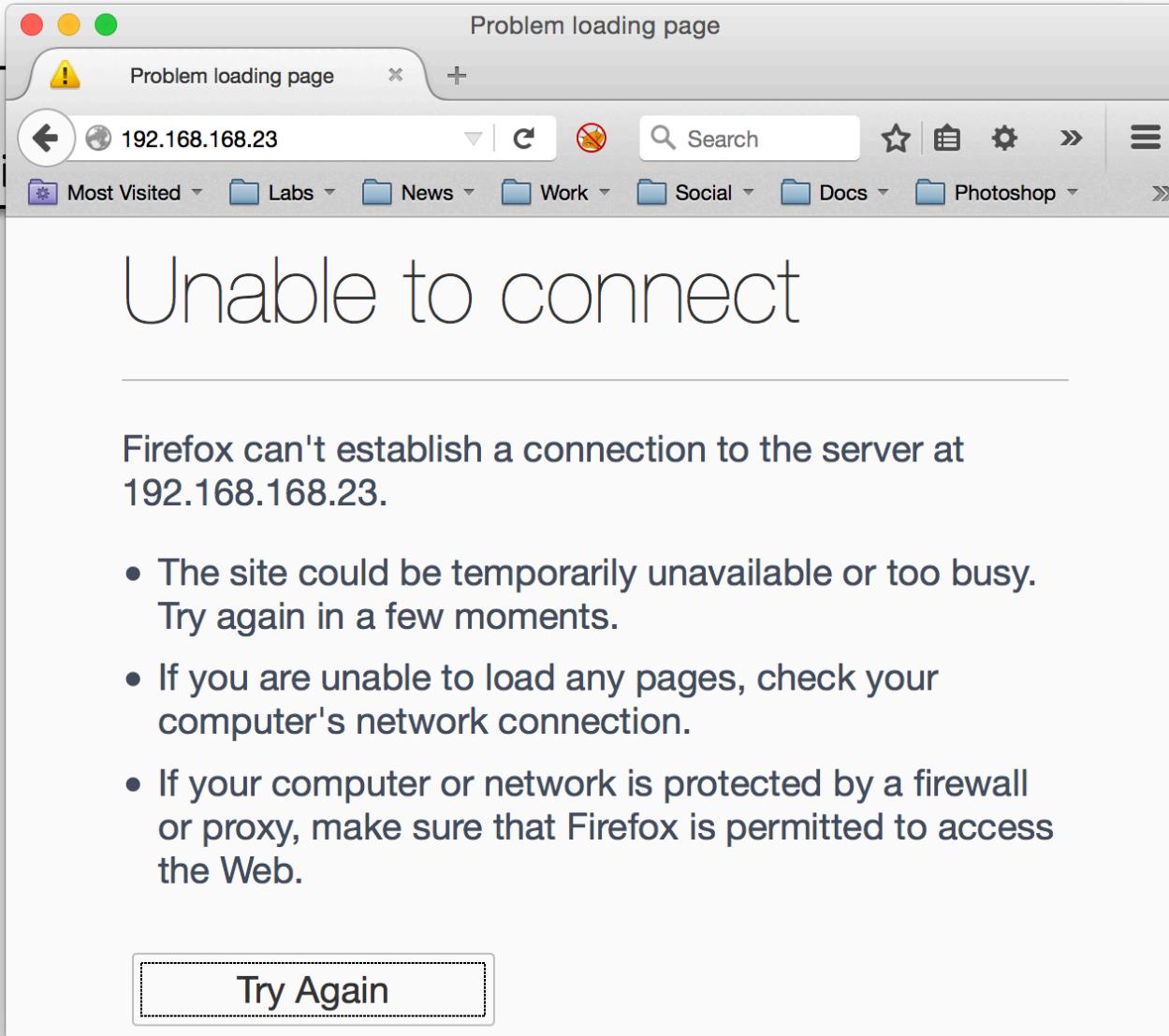
<http://www.internetsociety.org/internet/what-internet/history-internet>

Passiamo ora ad esaminare come si comporta un client (in genere un browser) nei confronti di richieste al server e sue relative risposte.

Useremo allo scopo delle semplicissime pagine HTML statiche, ossia pagine il cui contenuto rimane costante tutte le volte che vengono richieste al server.

Vediamo  
caso in cui

mpio nel  
cessibile:



The screenshot shows a Firefox browser window with the title "Problem loading page". The address bar displays "192.168.168.23". The main content area shows the text "Unable to connect" followed by an error message: "Firefox can't establish a connection to the server at 192.168.168.23." Below this, a bulleted list provides troubleshooting steps, and at the bottom is a "Try Again" button.

Problem loading page

192.168.168.23

Search

Most Visited | Labs | News | Work | Social | Docs | Photoshop

## Unable to connect

Firefox can't establish a connection to the server at 192.168.168.23.

- The site could be temporarily unavailable or too busy.  
Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

Supponiamo invece che il server sia attivo ed accessibile, ma che si richieda una pagina inesistente (un file che, nella zona deputata del server, non esiste)

## Configuration

Custom error documents are configured using the `ErrorDocument` directive, which may be used in global, virtualhost, or directory context. It may be used in .htaccess files if `AllowOverride` is set to `FileInfo`.

```
ErrorDocument 500 "Sorry, our script crashed. Oh dear"  
ErrorDocument 500 /cgi-bin/crash-recover  
ErrorDocument 500 http://error.example.com/server_error.html  
ErrorDocument 404 /errors/not_found.html  
ErrorDocument 401 /subscription/how_to_subscribe.html
```

The syntax of the `ErrorDocument` directive is:

```
ErrorDocument <3-digit-code> <action>
```

where the action will be treated as:

1. A local URL to redirect to (if the action begins with a "/").
2. An external URL to redirect to (if the action is a valid URL).
3. Text to be displayed (if none of the above). The text must be wrapped in quotes ("") if it consists of more than one word.

`/etc/httpd/conf/httpd.conf`

# Diagnostica

Mozilla Firefox

http://192.168.168.23/fake.html

Search

File Edit Search

192.168.168.23/fake.html

Most Visited Labs News Work Social Docs Photoshop

Sorry the page you're looking for does not exist!

```
<IfModule min
#
# The mod
# content
# direct
#
MIMEMagic
</IfModule>

#
# Customizabl
# 1) plain te
#
# Some examp
#ErrorDocume
#ErrorDocume
ErrorDocume
#ErrorDocume
#
#
# EnableMMAP
# memory-map
# files. Th.
```

from the  
cFile  
d.

Supponiamo  
file default

iva del  
to con

# Diagnostica

Test Page for the Apache HTTP Server on Scientific Linux

Test Page for the Apache HTTP ... 192.168.168.23

Most Visited Labs News Work Social Docs Photoshop

## Scientific Linux Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

**If you are a member of the general public:**

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

**If you are the website administrator:**

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

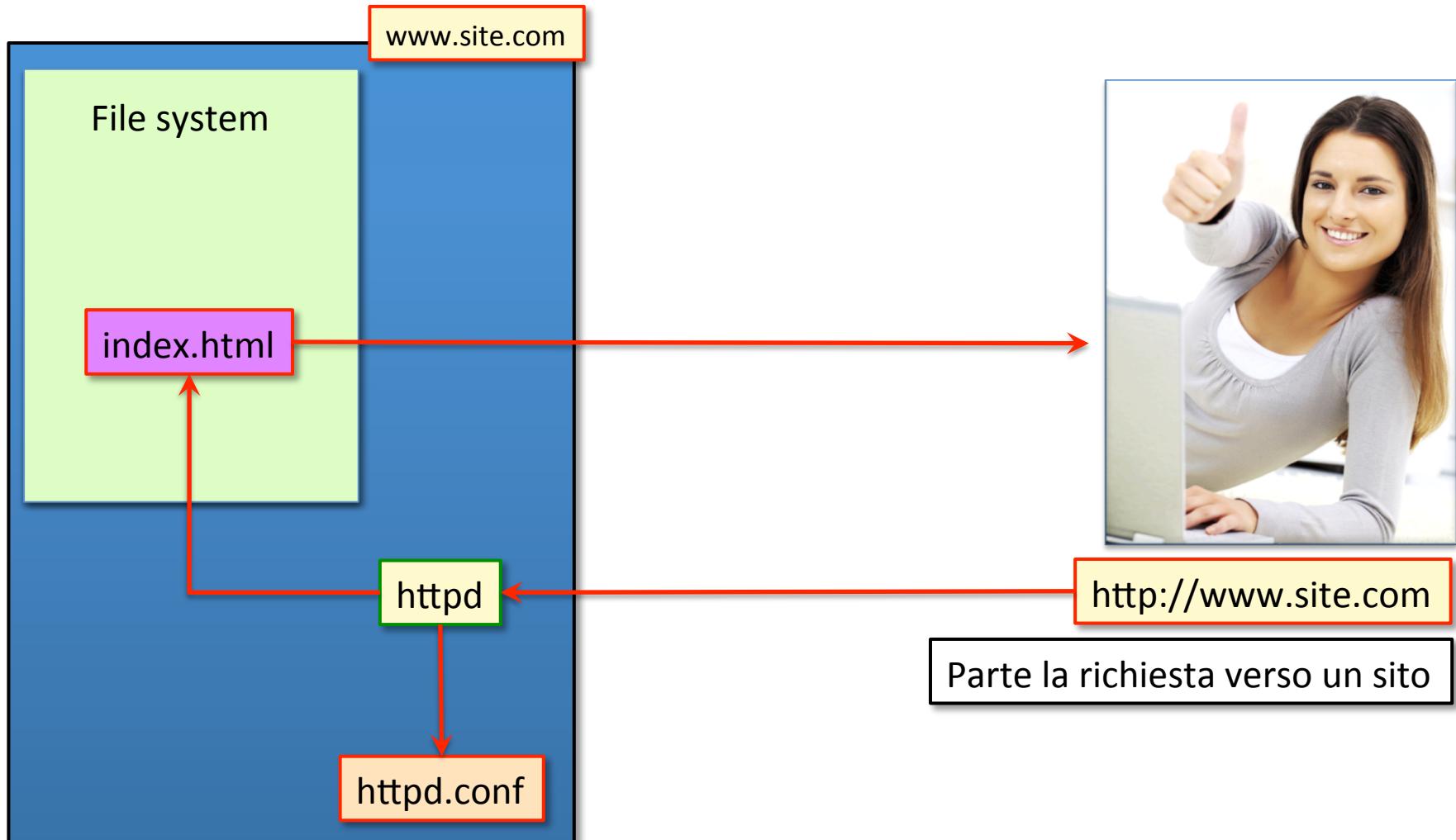


Anche questo messaggio corrisponde ad un file localizzato nel filesystem accessibile al web server: in particolare il modo per trovarlo ed eventualmente personalizzarlo è il seguente:

```
index.html - /usr/share/httpd/noindex/
File Edit Search Preferences Shell Macro Windows Help
httpd.conf welcome.conf index.html x

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <title>Test Page for the Apache HTTP Server on Scientific Linux</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <style type="text/css">
            /*<![CDATA[*/
            body {
                background-color: #fff;
                color: #000;
                font-size: 0.9em;
                font-family: sans-serif, helvetica;
                margin: 0;
                padding: 0;
            }
            :link {
                color: #c00;
            }
            :visited {
                color: #c00;
            }
        </pre>
```

# Come funziona http



# I necessari privilegi di sistema

La directory **/var/www/html** non è però la più adatta a contenere materiale di proprietà di singoli individui sul *server* in quanto sono richiesti privilegi di amministratore per poter creare e modificare *files* in quella zona, che di default appartiene a **root**.

Per ovviare a ciò sono possibili due soluzioni alternative:

- Creare in quella zona dei soft-link alle aree personali e abilitare il server **httpd** a fornire materiale proveniente da quelle aree a chi ne facesse richiesta con privilegi identici per tutti
  - ✓ I privilegi sarebbero però, per tutti, gli stessi definiti per l'area **/var/www/html** di provenienza, senza possibilità di differenziarli su base individuale
- Creare delle aree personali e abilitare privilegi specifici e potenzialmente differenti tra loro per ognuna di esse
  - ✓ In questo caso è possibile definire per ogni area dei privilegi particolarizzati: il prezzo che si paga per questa possibilità è che l'amministratore deve specificare, nel file di configurazione, le caratteristiche specifiche per ogni utente e per ogni sua area

# Privilegi individuali

httpd.conf - /etc/httpd/conf/

File Edit Search Preferences Shell Macro Windows Help

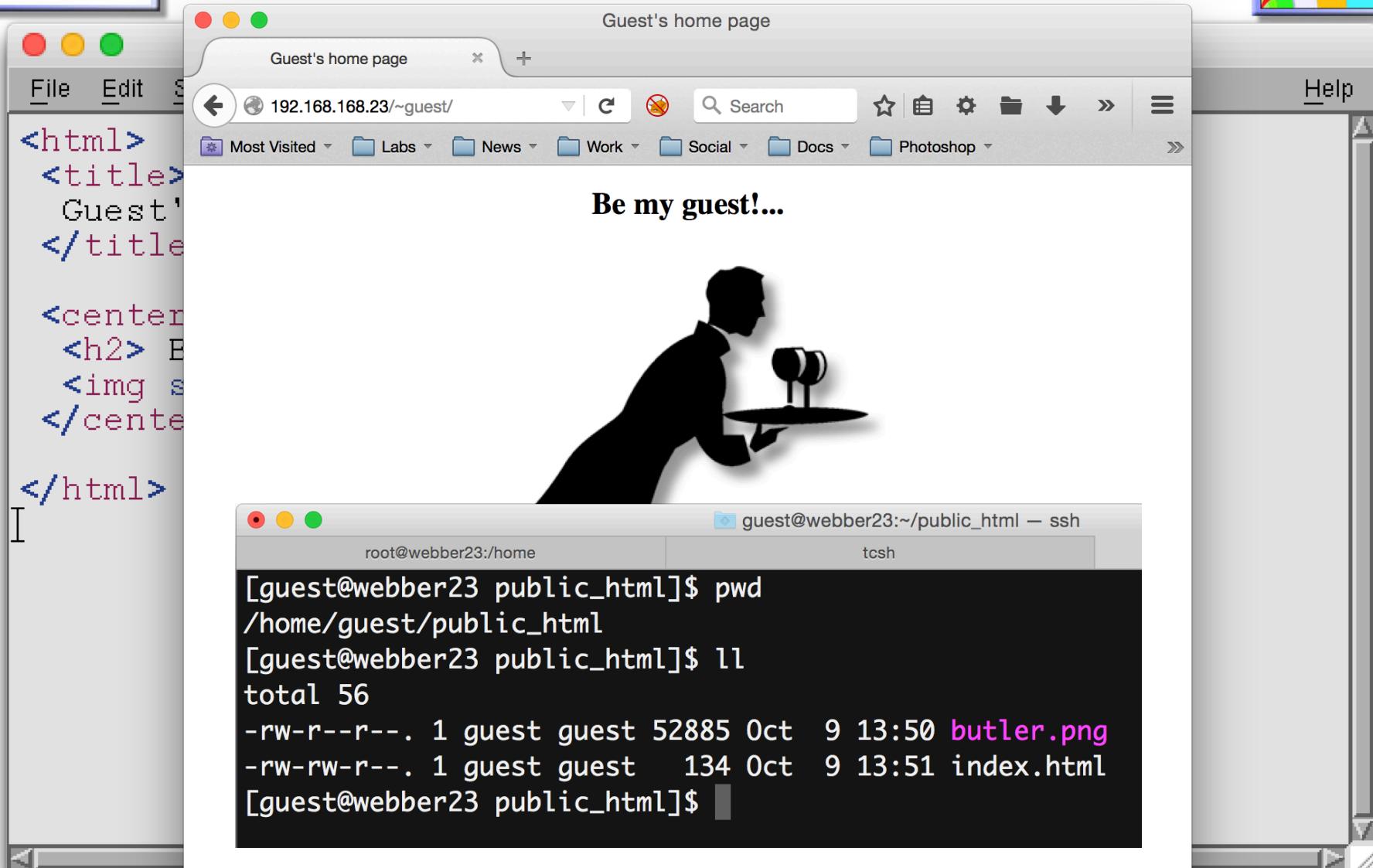
```
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid. This usually means that ~userid
# must have permissions of 711, ~userid/public html must have permissions
# of 755, and document
# Otherwise, the client
#
# See also: http://h
#
<IfModule mod_userdir>
    #
    # UserDir is disabled by default
    # of a username on the system (depending on home directory
    # permissions).
    #
    UserDir enabled guest
    #
    # To enable requests to /~user/ to serve the user's public_html
    # directory, remove the "UserDir disable" line above, and uncomment
    # the following line instead:
    #
    UserDir public_html
</IfModule>
```

[root@webber23 home]# systemctl restart httpd
[root@webber23 home]# chmod 711 /home/guest
[root@webber23 home]# chmod 755 /home/guest/public\_html/
[root@webber23 home]#

root@webber23:/home tcsh 9

/etc/httpd/conf/httpd.conf

# La zona pubblica degli user



The image shows a Mac OS X desktop environment with several windows open:

- A web browser window titled "Guest's home page" showing the URL "192.168.168.23/~guest/". The page content includes the text "Be my guest!..." and a silhouette of a waiter holding a tray with two glasses.
- A terminal window titled "guest@webber23:~/public\_html — ssh" showing the command-line output:

```
[guest@webber23 public_html]$ pwd
/home/guest/public_html
[guest@webber23 public_html]$ ll
total 56
-rw-r--r--. 1 guest guest 52885 Oct  9 13:50 butler.png
-rw-rw-r--. 1 guest guest   134 Oct  9 13:51 index.html
[guest@webber23 public_html]$
```
- An Xcode code editor window showing the source code for "index.html":

```
<html>
<title>
Guest'
</title>

<center>
<h2> Be my guest!...
<img s
</center>

</html>
```
- A partially visible window on the right labeled "Help".

Nel file di configurazione, **httpd.conf**, definiremo quindi i privilegi per singole aree:  
 ad esempio definiamo i privilegi per tutti i file contenuti nelle cartelle  
**public\_html** contenute in **/user/gr1/e831/\*** (ci possono essere varie  
 zone utenti)

```
#  
# Control access to UserDir directories. The following is an example  
# for a site where these directories are restricted to read-only.  
#  
<Directory /user/gr1/e831/*/public_html>  
    AllowOverride FileInfo AuthConfig Limit  
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec  
    <Limit GET POST OPTIONS>  
        Order allow,deny  
        Allow from all  
    </Limit>  
    <LimitExcept GET POST OPTIONS>  
        Order deny,allow  
        Deny from all  
    </LimitExcept>  
</Directory>
```

**/etc/httpd/conf/httpd.conf**

```
[root@mobile html]# service httpd restart  
Stopping httpd:  
Starting httpd:  
[root@mobile html]#
```

[ OK ]  
[ OK ]

# Accesso alla directory



Index of /~menasce

httpd.conf - /etc/httpd/conf/

File Edit Search Preferences Shell Macro Windows Help

```

#
# UserDir public_html

</IfModule>

<Directory /home/quest/public_html>
    Options Indexes
    Options FollowSymLinks
    Options +Includes
</Directory>

```

root@webber23:/home tcsh guest@webber23:~/public\_html/varie — ssh

```

# [guest@webber23 varie]$ pwd
# Directo: /home/guest/public_html/varie
# is requ: [guest@webber23 varie]$ ln -s ../butler.png maggiordomo.png
# <IfModule [guest@webber23 varie]$ ll
# Direct: total 0
# </IfModule> lrwxrwxrwx. 1 guest guest 13 Oct  9 14:00 maggiordomo.png -> ../butler.png
# [guest@webber23 varie]$ 
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
```

JSD OFF

Index of /~guest/varie

192.168.168.23/~guest/varie/

Most Visited Labs News Work Social Docs Photoshop

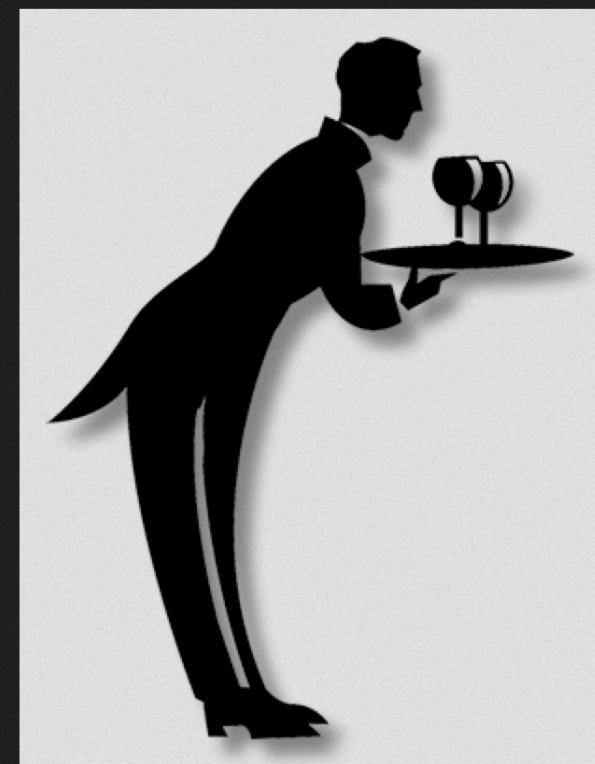
# Index of /~guest/varie

Name	Last modified
<a href="#">Parent Directory</a>	
 <a href="#">maggidomo.png</a>	2015-10-09 13:51

maggiordomo.png (PNG Image, 310 x 405 pixels)

192.168.168.23/~guest/varie/maggio

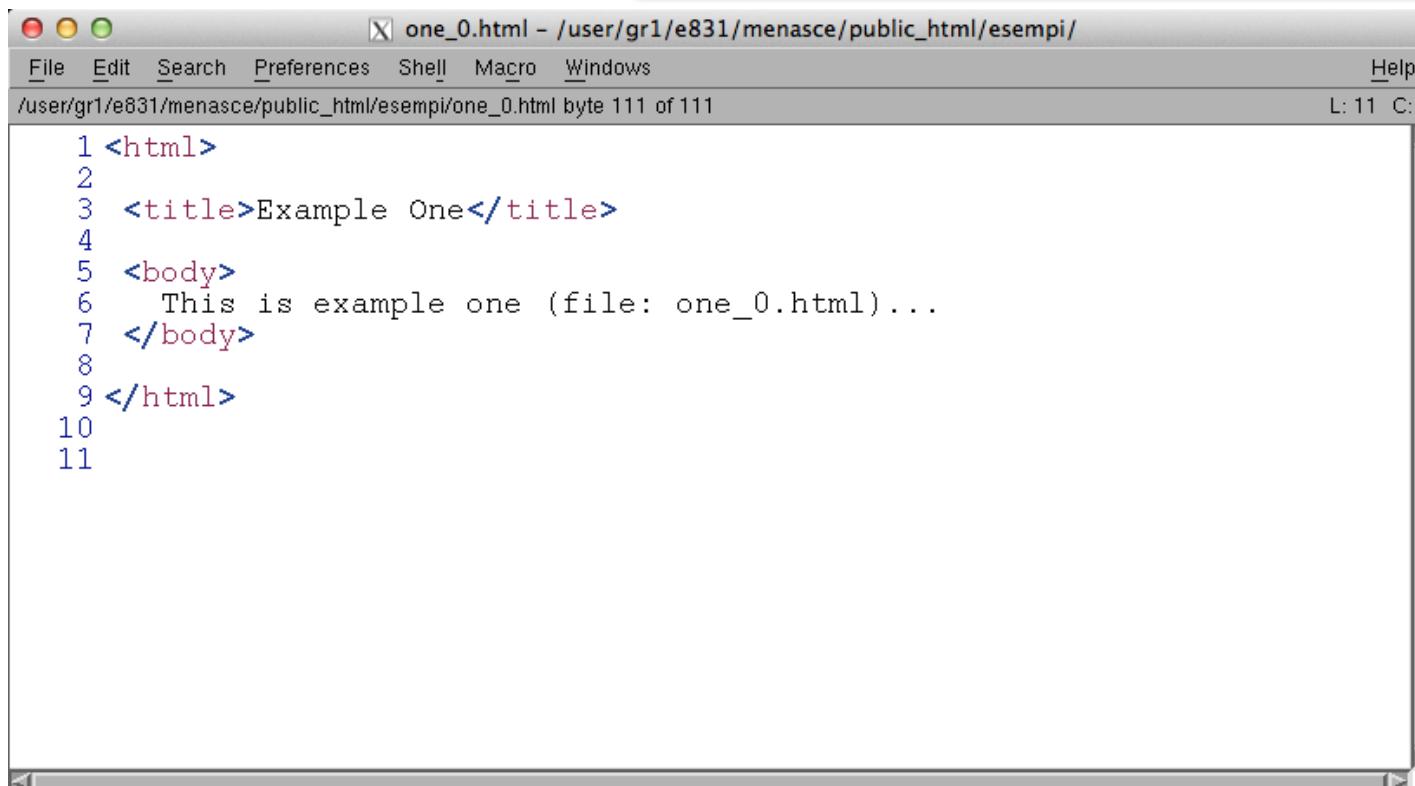
Most Visited Labs News Work Social Docs Photoshop



# Un primo esempio user

```
> cd ~/public_html  
> mkdir esempi  
> nedit-client one_0.html
```

```
> pwd  
/user/gr1/e831/menasce/public_html/esempi  
> ll  
total 12  
drwxr-xr-x. 2 menasce e831 4096 May  9 09:45 .  
drwxr-xr-x. 5 menasce e831 4096 May  9 09:45 ..  
-rw-r--r--. 1 menasce e831   111 May  9 09:18 one_0.html  
>
```



A screenshot of a Mac OS X application window titled "one\_0.html - /user/gr1/e831/menasce/public\_html/esempi/". The window contains a text editor with the following content:

```
1 <html>  
2  
3 <title>Example One</title>  
4  
5 <body>  
6   This is example one (file: one_0.html)...  
7 </body>  
8  
9 </html>  
10  
11
```

Index of /~menasce/esempi +

mobile.mib.infn.it/~menasce/esempi/

# Index of /~menasce/esempi

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>	-		
<a href="#">one_0.html</a>	09-May-2014 09:18	111	

Apache/2.2.15 (Red Hat) Server at mobile.mib.infn.it Port 80

This is example one (file: one\_0.html)...

Esteticamente molto poco soddisfacente....

Cerchiamo di migliorare l'aspetto estetico della pagina agendo sul contenuto dei **tag html**

# Pagina con sfondo

```
> mkdir icons
> wget .....
> cp one_0.html one_1.html
> alias nc nedit-client
> nc one_1.html
```

```
> ll
total 20
drwxr-xr-x. 3 menasce e831 4096 May  9 09:58 .
drwxr-xr-x. 5 menasce e831 4096 May  9 09:45 ..
drwxr-xr-x. 2 menasce e831 4096 May  9 10:00 icons
-rw-r--r--. 1 menasce e831   11 May  9 09:18 one_0.html
-rw-r--r--. 1 menasce e831   11 May  9 09:57 one_1.html
> ll icons/
total 292
drwxr-xr-x. 2 menasce e831 4096 May  9 10:00 .
drwxr-xr-x. 3 menasce e831 4096 May  9 09:58 ..
-rw-r--r--. 1 menasce e831 287426 May  9 10:00 image.png
> display icons/image.png
```



one\_1.html - /user/gr1/e831/menasce/public\_html/esempi/

File Edit Search Preferences Shell Macro Windows Help

one\_1.html |one\_2.html |

/user/gr1/e831/menasce/public\_html/esempi/one\_1.html byte 140 of 140 L: 11 C: 0

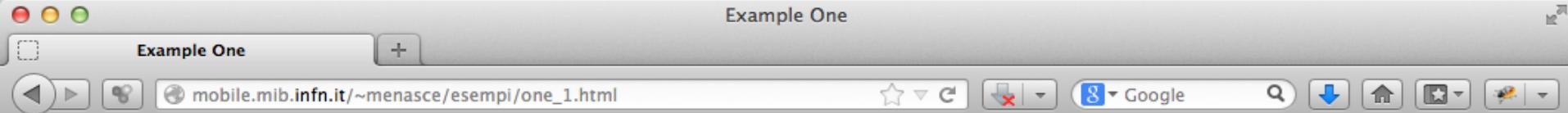
```

1 <html>
2
3 <title>Example One</title>
4
5 <body background="icons/image.png">
6 This is example one (file: one_1.html)...
7 </body>
8
9 </html>
10
11]
```



This is example one (file: one\_0.html)...

Ma così il testo non si legge più! (nero su nero....)

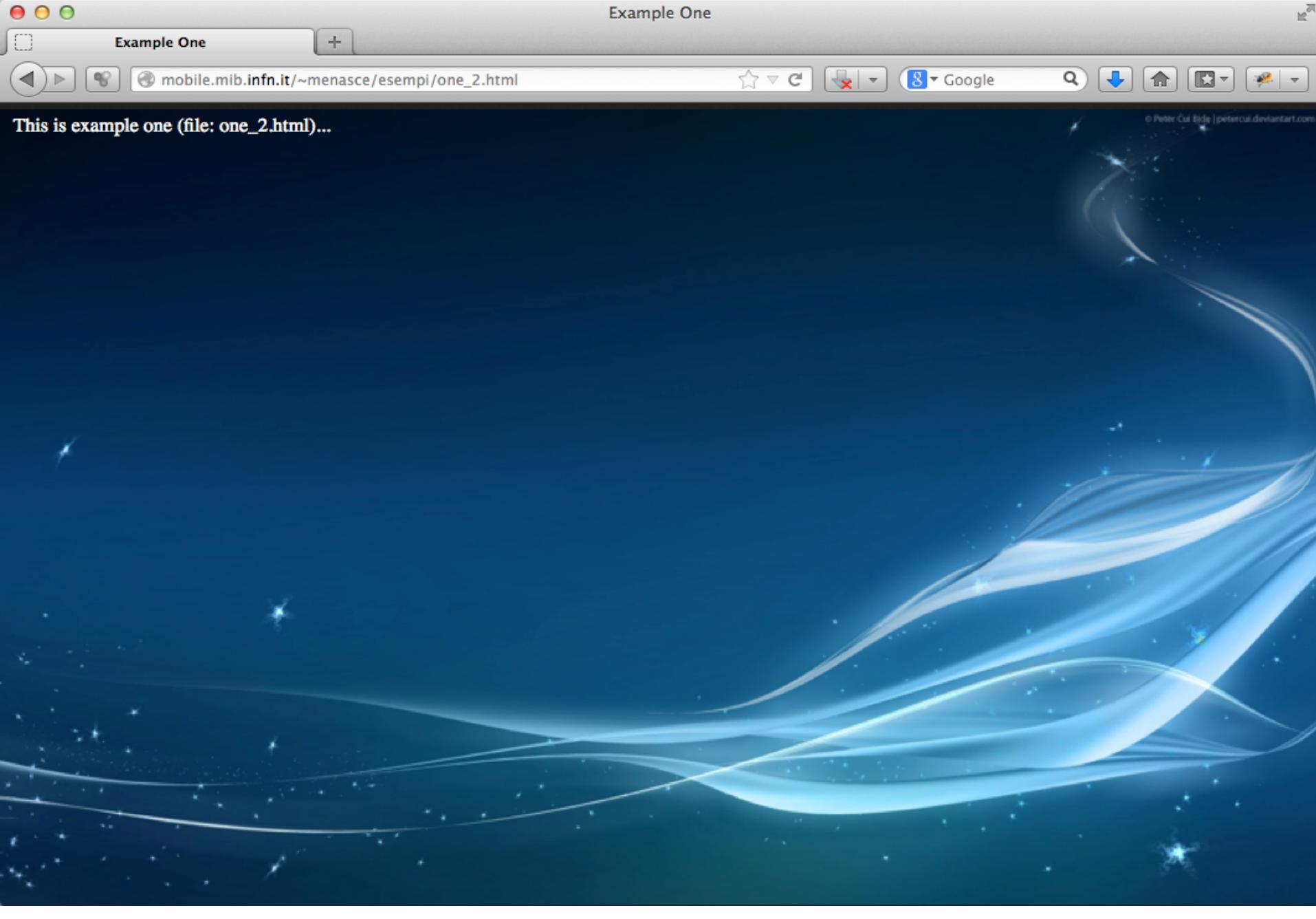


This is example one (file: one\_0.html)...

© Peter Čul Bide | petercul.deviantart.com

```
one_2.html - /user/gr1/e831/menasce/public_html/esempi/
File Edit Search Preferences Shell Macro Windows Help
/usr/gr1/e831/menasce/public_html/esempi/one_2.html byte 178 of 178 L: 13 C: 0

1 <html>
2
3 <title>Example One</title>
4
5 <body background="icons/image.png">
6 <div style="color: white">
7 This is example one (file: one_2.html)...
8 </div>
9 </body>
10
11 </html>
12
13[
```



[Register](#) or [Log in](#)[Other Applications](#) ▾

mozilla ▾



# ADD-ONS

[EXTENSIONS](#) | [THEMES](#) | [COLLECTIONS](#) | [MORE...](#) search for add-ons[Home](#) » [Extensions](#) » [Web Developer](#)

**Web Developer** 1.2.5  
by chrispederick

The Web Developer extension adds

[+ Add to Firefox](#)

Utilizziamo degli utili strumenti di controllo per lo sviluppo di pagine Web: uno di questi, un plug-in di Firefox, si chiama **Web Developer** e lo troviamo sul sito di Mozilla:



179 user reviews

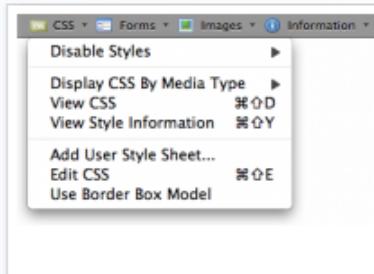
750,510 users

[Add to collection](#)[Share this Add-on](#)*Enjoy this add-on?*

The developer of this add-on asks that you help support its continued development by making a small contribution.

[Contribute](#)

\$9.99 suggested



Dopo aver installato il plug-in, fare right-click sulla barra di navigazione per abilitarlo.

Mozilla Foundation (US) | https://addons.mozilla.org/en-US/firefox/addon/web-c  

Navigation Toolbar  
Bookmarks Toolbar  
**Web Developer Toolbar**  
Add-on Bar  
Customize...

**mozilla** ▾



# ADD-ONS

EXTENSIONS | THEMES | COLLECTIONS | MORE...

Extensions » Web Developer

 **Web Developer** 1.2.5

by chrispederick

The Web Developer extension adds various web developer tools to the browser.

**+ Add to Firefox**

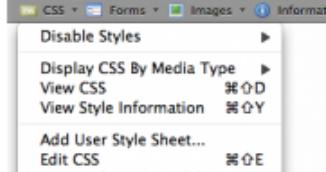
★★★★★  
179 user reviews  
750,510 users

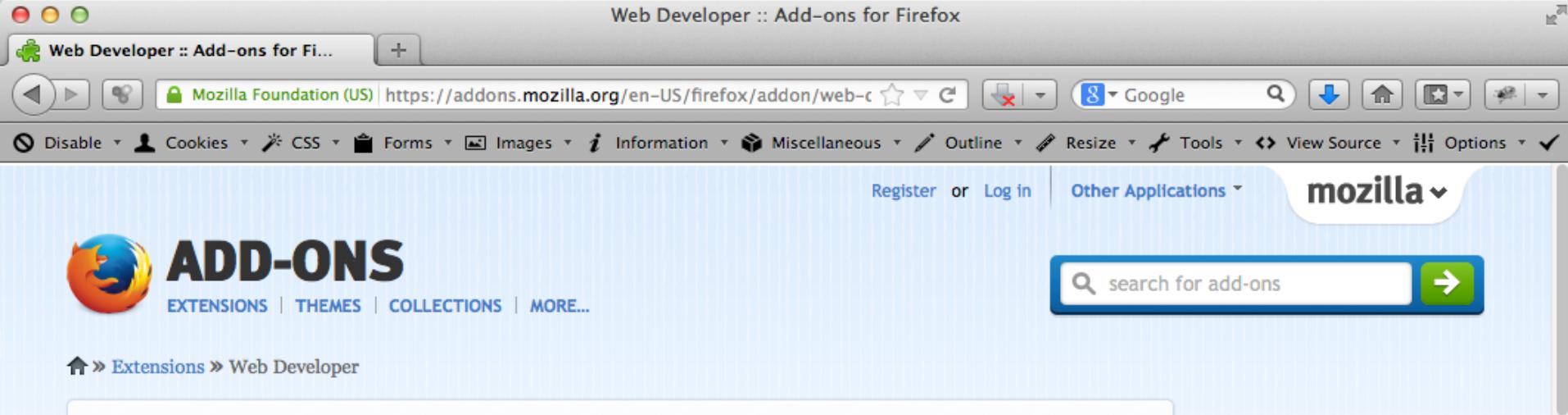
 Add to collection  
 Share this Add-on

 Enjoy this add-on?

The developer of this add-on asks that you help support its continued development by making a small contribution.

**Contribute**  
\$9.99 suggested

«  »





# ADD-ONS

EXTENSIONS | THEMES | CO

↑ » Extensions » Web Developer



*Web Developer* 1.2.5

by chrispederick

The Web Developer extension adds various web developer tools to the browser.

[+ Add to Firefox](#)



179 user reviews

750,510 users 

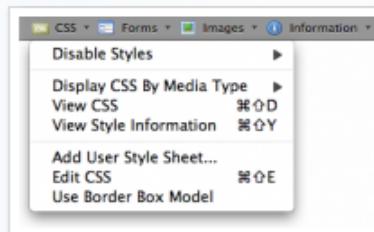


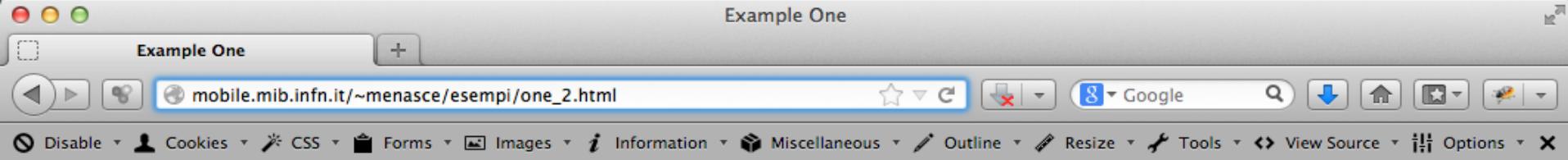
*Enjoy this add-on?*

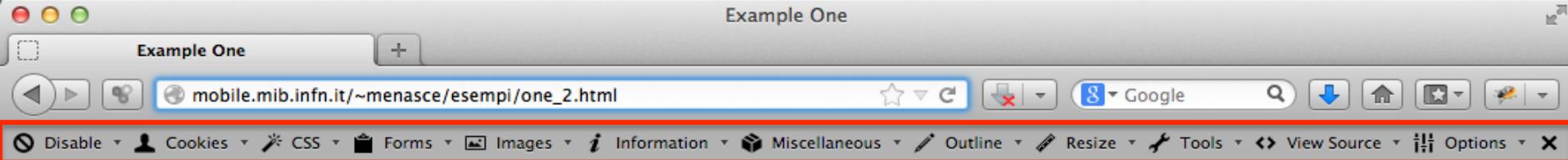
The developer of this add-on asks that you help support its continued development by making a small contribution.

 Contribute

**\$9.99 suggested**







© Peter Culicida | petercul.deviantart.com

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options X

This is example one (file: one\_2.html)...

View Frame Source  
View Generated Source

View Source

⤵⤶

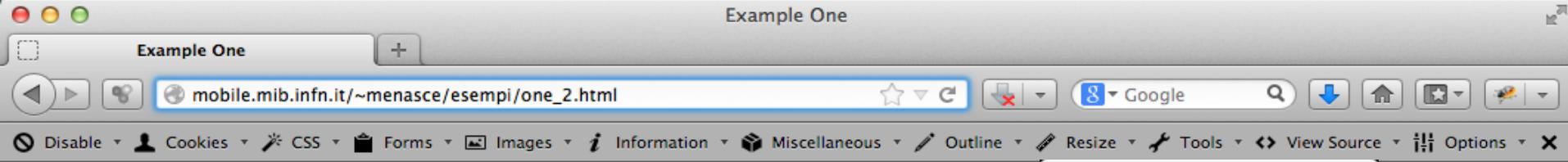
Edit View Source With Applications...

DOM Source of Selection

```
1 <html><head><title>Example One</title>
2
3 </head><body background="icons/image.png">
4 <div style="color: white">
5   This is example one (file: one_2.html)...
6 </div>
7
8
9
10
11 </body></html>
```

Source of: http://mobile.mib.infn.it/~menasce/esempi/one\_2.html

```
1 <html>
2
3   <title>Example One</title>
4
5   <body background="icons/image.png">
6     <div style="color: white">
7       This is example one (file: one_2.html)...
8     </div>
9   </body>
10
11 </html>
12
```



This is example one (file: one\_2.html)...

Valida CSS  
Valida Feed  
Valida HTML  
Valida collegamenti  
Valida Section 508  
Valida WAI

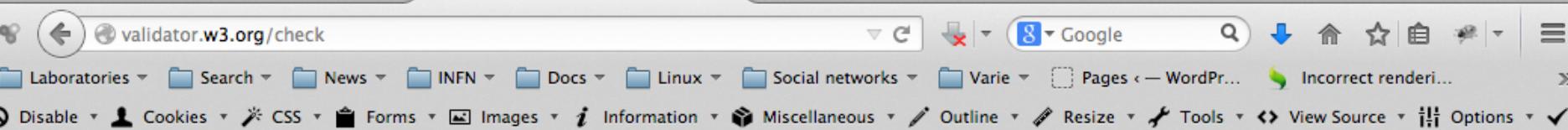
Edit Tools...

Display Page Validation

DOM Inspector  
Error Console

Validate Local CSS

Validate Local HTML



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: Notes and Potential Issues Validation Output Source Listing

### Error found while checking this document as HTML 4.01 Transitional!

**Result:** 1 Error, 4 warning(s)

**File :**  No file selected.

*Use the file selection box above if you wish to re-validate the uploaded file web-developer-mobile.mib.infn.it-1400758720328.html*

**Modified:** (undefined)

**Server:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:29.0) Gecko/20100101 Firefox/29.0

**Size:** (undefined)

**Content-Type:** text/html

**Encoding :** utf-8

**Dootype :** HTML 4.01 Transitional

**Root Element:** html



Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.



The W3C validators are hosted on server technology donated by HP, and supported by community donations.

[Donate](#) and help us build better tools for a better web.

4841

Flattr

validator.w3.org/check

Laboratories Search News INFN Docs Linux Social networks Varie Pages — WordPress Incorrect rendering Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

## ⚠ No DOCTYPE found! Checking with default HTML 4.01 Transitional Document Type.

No DOCTYPE Declaration could be found or recognized in this document. This generally means that the document is not declaring its Document Type at the top. It can also mean that the DOCTYPE declaration contains a spelling error, or that it is not using the correct syntax.

The document was checked using a default "fallback" Document Type Definition that closely resembles "HTML 4.01 Transitional".

It is recommended to use a DOCTYPE declaration as the very first thing in your HTML document, to identify the precise language being used. For example, for a typical [XHTML 1.0](#) document:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <title>Title</title>
  </head>

  <body>
    <!-- ... body of document ... -->
  </body>
</html>
```

For XML documents, you may also wish to include an "XML Declaration" even before the DOCTYPE Declaration, but this is not well supported in older browsers. More information about this can be found in the [XHTML 1.0](#) Recommendation.



Disable

Cookies

CSS

Forms

Images

Information

Miscellaneous

Outline

Resize

Tools

View Source

Options

This is example one (file: one\_2)

## Disable Images

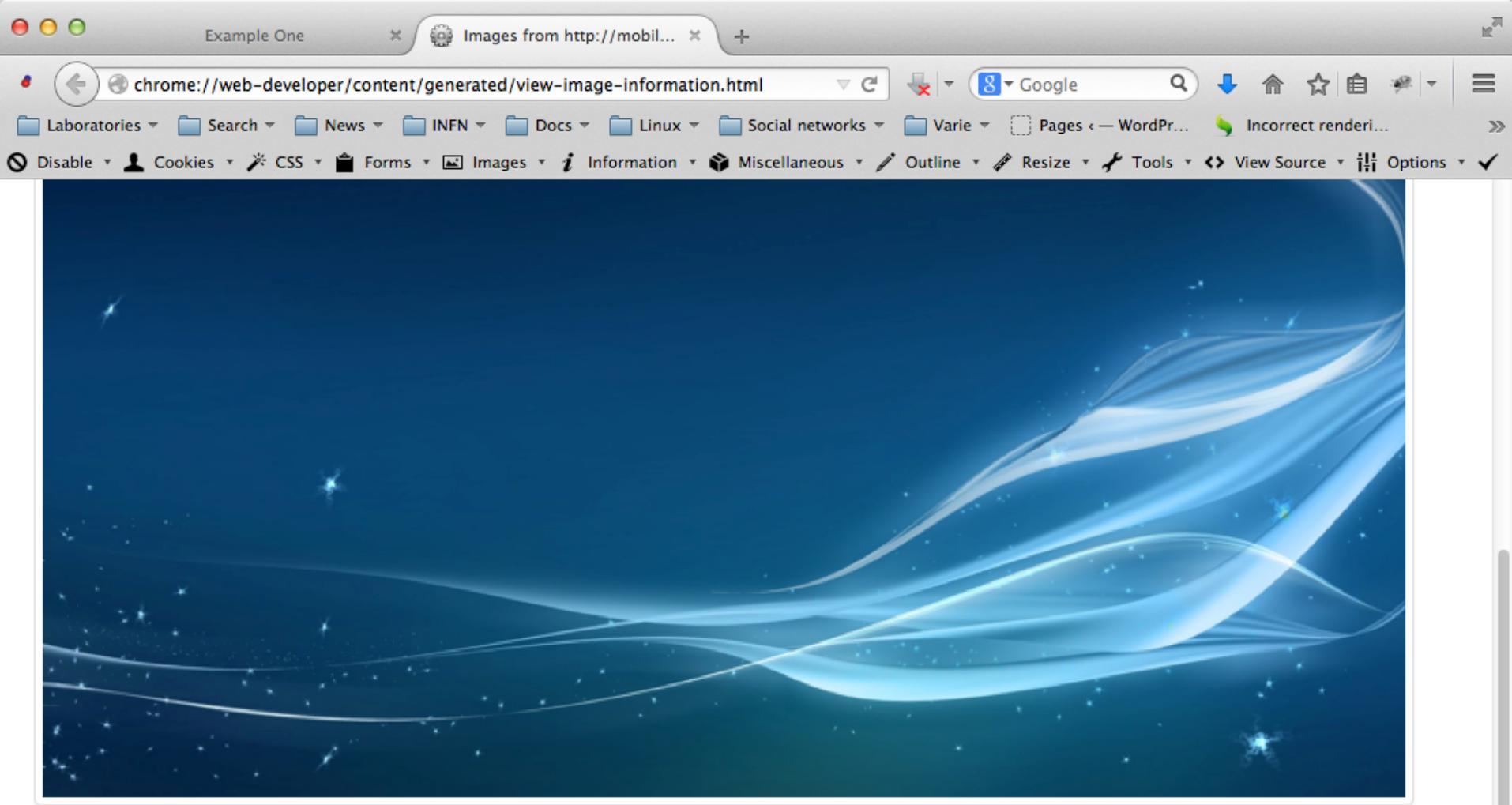
- Display Alt Attributes
- Display Image Dimensions
- Display Image File Sizes
- Display Image Paths

- Find Broken Images
- Hide Background Images
- Hide Images
- Make Images Full Size
- Make Images Invisible

- Outline Images
- Reload Images
- Replace Images With Alt Attributes

- View Image Information**

© Peter Cui Ridge | peteircui.deviantart.com



Property	Value
Src	<a href="http://mobile.mib.infn.it/~menasce/esempi/icons/image.png">http://mobile.mib.infn.it/~menasce/esempi/icons/image.png</a>
Width	798
Height	504

Example One

mobile.mib.infn.it/~menasce/esempi/one\_2.html

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Disable Styles Add User Style Sheet... Display Style Information Display Styles By Media Type Edit CSS Reload Linked Style Sheets Use Border Box Model

Disable Proxy View CSS

✓ Use Auto-detect Proxy Use Configuration URL Proxy Use Manual Proxy Use System Proxy

© Peter Cui Ridge | petercui.deviantart.com



Disable

Cookies

CSS

Forms

Images

Information

Miscellaneous

Outline

Resize

Tools

View Source

Options

X

This is example one (file: one\_2.html)...

© Peter Cui Ridge |petercui.deviantart.com

Vediamo ora come decorare la pagina  
con elementi grafici utilizzando i  
CSS (Cascading Style Sheets)

File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_3.html - /user/gr1/e831/menasce/public\_html/esempi/one\_3.html byte 1253 of 1253 L: 38 C: 0

```
1 <html>
2   <title>Two</title>
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 <body background="icons/image.png">
21   <div style="color: white">
22     This is page one... (file one 3.html)
23
24
25
26
27
28   </div>
29
30 NB: Questo testo consiste in un rimaneggiamento casuale dell'originale testo latino di
31 Cicerone De finibus bonorum et malorum, dedicato a Marco Bruto e viene usato sin
32 dal '500 in tipografia come placeholder di testi decorativi
33
34 </body>
```

Nel protocollo HTML, **div** è un *tag* che permette di raccogliere elementi testuali e grafici in un *container* dotato di proprietà gestite dai CSS in maniera globale.

Supponiamo di volere aggiungere del nuovo testo, posizionato in un punto arbitrario della *viewport* del *browser*, ad esempio 30 *pixel* sotto il bordo superiore:

**NB:** Questo testo consiste in un rimaneggiamento casuale dell'originale testo latino di Cicerone ***De finibus bonorum et malorum***, dedicato a Marco Bruto e viene usato sin dal '500 in tipografia come *placeholder* di testi decorativi



File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_3.html byte 1249 of 1250 L: 37 C: 0

```
1 <html>
2   <title>Two</title>
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 <body background="icons/image.png">
21   <div style="color: white">
22     This is page one... (file one_3.html)
23     <div style="position: relative; top: 30;">
24       <hr/>
25       Lorem ipsum dolor sit amet, consectetur adipisicing
26       elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
27     </div>
28   </div>
29
30
31
32
33
34
35 </body>
```

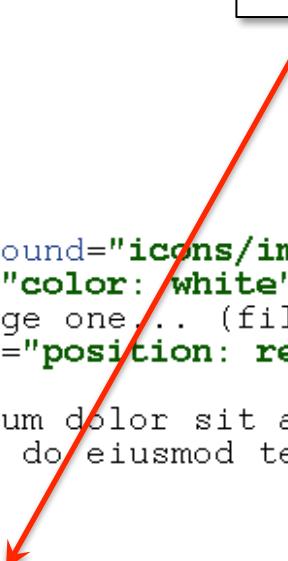
Supponiamo poi di voler aggiungere una breve scritta di benvenuto che sia sempre posizionata a 10 pixel sotto il bordo superiore della *viewport* e a 200 pixel a destra del bordo sinistro (in modo che resti fissa durante operazioni di *scroll* dello schermo), dotata di un certo numero di caratteristiche grafiche quali bordo colorato, fonte particolare ecc...

File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_3.html byte 1249 of 1250 L: 37 C: 0

```
1 <html>
2   <title>Two</title>
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 <body background="icons/image.png">
21   <div style="color: white">
22     This is page one... (file one_3.html)
23     <div style="position: relative; top: 30;">
24       <hr/>
25       Lorem ipsum dolor sit amet, consectetur adipisicing
26       elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
27     </div>
28   </div>
29
30
31
32
33
34
35 </body>
```

Se dotiamo il **div**, che contiene questo messaggio di benvenuto, di un *qualificatore*, **id**, potremo spostare tutte le specifiche di stile in un altro punto del file **html**, in modo da poter usare quelle stesse specifiche anche in altri contesti: scegliamo come **id** la stringa **"menu"**



File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_3.html byte 1249 of 1250 L: 37 C: 0

```
1 <html>
2   <title>Two</title>
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 <body background="icons/image.png">
21   <div style="color: white">
22     This is page one... (file one_3.html)
23   <div style="position: relative; top: 30;">
24     <hr/>
25     Lorem ipsum dolor sit amet, consectetur adipisicing
26     elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
27   </div>
28 </div>
29
30   <div id="menu" style="float: top;">
31     Welcome!!!
32   </div>
33
34
35 </body>
```

Predisponiamo ora un nuovo *tag* di tipo *style*, nel quale porremo le specifiche di stile (CSS) raggruppate per *id-name* (nel nostro caso per ora un solo **id**, **menu**).

Il *rendering engine* di Firefox, ad ogni *refresh* della *viewport*, eseguendo il *parsing* dell'html e trovando il *tag* **div** con il corrispondente **id** labellato **menu**, verificherà l'esistenza di un corrispondente **#id** negli *styles* definiti e ne applicherà i valori predisposti

This is page one... (file one\_3.html)

Welcome!!!

© Peter Cul Ride | petercul.deviantart.com

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Vogliamo a questo punto manipolare il contenuto della pagina in maniera programmatica, facendo in modo che il testo di benvenuto cambi ogni due secondi e mostri invece l'ora corrente (un orologio digitale)

Useremo a tale scopo **JavaScript**

Prima, però, una breve storia dello sviluppo dell'HTML...

- L'HTML nasce al CERN da un'idea di Tim Berners-Lee, che si basa su di un preesistente protocollo, chiamato GML, sviluppato all'IBM per la gestione e la formattazione di testi.
- L'idea di base consiste in un **disaccoppiamento** tra il **contenuto** di un messaggio da trasmettere e la sua **rappresentazione**. Questo disaccoppiamento è implementato nel messaggio stesso tramite degli opportuni simboli, chiamati **markup**, che informano il ricevente (ad esempio un browser) sulla modalità di rappresentazione del contenuto: nel caso di un testo, la rappresentazione concernerà il tipo di fonte, il colore e la dimensione piuttosto che non la posizione. Nel caso di una fotografia o di un filmato la rappresentazione avrà a che vedere con i colori, la dimensione, la posizione o il frame-rate, mentre nel caso di un audio avrà a che vedere con volume, filtri bassi e acuti e così via.
- Il protocollo HTML specifica i tipi e la sintassi di questi markup: sarà poi compito di chi scrive un'applicazione capace di ricevere un file di questo tipo di predisporre gli strumenti per intercettare i markup (**parsing** del documento) e a procedere alla rappresentazione opportuna del messaggio (**rendering**)

one\_2.html - /user/gr1/e831/menasce/public\_html/esempi/

File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_2.html byte 178 of 178 L: 13 C: 0

```

1 <html>
2
3 <title>Example One</title>
4
5 <body background="icons/image.png">
6 <div style="color: white">
7   This is example one (file: one_2.html)...
8 </div>
9 </body>
10
11 </html>
12
13 I

```

Messaggio da rappresentare

Modalità di rappresentazione

Tre sono le componenti che è bene tenere distinte:

- Sintassi
- Contenuto
- Rappresentazione grafica del contenuto e formattazione (compito dei CSS)

Quando un browser riceve un documento HTML da un server che glielo fornisce su richiesta, parte una fase, chiamata **parsing**, nella quale il documento viene analizzato sintatticamente: le parti contenenti errori di sintassi vengono ignorate ed il tutto viene trasformato in una struttura ad albero (un **B-Tree**) in memoria, struttura che è costruita come un **DOM** (Document Object Model).

Gli errori di sintassi vengono ignorati nel senso che le corrispondenti direttive non vengono trasmesse al *rendering engine* del *browser* e di conseguenza la rappresentazione aderirà ai valori facoltativi **previsti** invece che a quelli **richiesti**.

Di fatto, il *rendering engine* costuirà la *bit-map* necessaria a rappresentare il documento nella *viewport* del *browser* e la proietterà una prima volta sullo schermo.

Se il documento è di tipo **statico**, ossia se contiene solamente direttive di tipo HTML senza componenti di ***JavaScript***, la *bit-map* in memoria non subirà modifiche nel tempo: all'interno del browser è costantemente attivo un ***loop***, chiamato ***event-loop*** (caratteristico di tutti i programmi a carattere interattivo), che, con una certa frequenza, verifica se parti del **DOM** in memoria vengono marcate come **dirty**. Questo può avvenire solo tramite l'esecuzione di uno script che ne altera delle parti: se questo avviene, le parti modificate vengono marcate come **dirty** ed il *rendering engine* provvede ad aggiornare periodicamente la *bit-map* in memoria, con la conseguenza che il contenuto della pagina appare modificarsi nel tempo.

Oltre al **parser** HTML e al **rendering engine**, un moderno browser contiene anche altre componenti, una delle quali è un **interpreter** JavaScript, capace di alterare il contenuto della bit-map in memoria in modo programmatico.

Questa componente viene attivata dal *parser* HTML ogniqualvolta un *tag* faccia riferimento ad un pezzo di codice JavaScript: a quel punto il codice viene passato, *verbatim*, al *parser* dell'*interpreter* il quale genera l'opportuno *byte-code* in memoria e lo esegue.

Anche nel caso dell'*interpreter*, la modalità facoltativa di esecuzione prevede che un errore di sintassi venga intercettato ed ignorato, con la conseguenza che in quel caso la pagina non godrà della funzionalità prevista.

In generale JavaScript agisce sul DOM in memoria, sia modificandone parti che creandone di nuove. Ad ogni fase ciclica del *rendering engine*, tutte le parti che vengono rappresentate nella *viewport* vengono dichiarate *clean*, in modo che variazioni introdotte da parti di JavaScript possano essere marcate per un *update*.

Questo processo di marcatura e di update selettivo prende il nome di **reflow** e la sua efficacia ed efficienza dipendono dalla particolare implementazione del meccanismo: nel caso di Firefox questa è molto buona, ma nel caso di Chrome è veramente eccezionale e rapida.

Il motore di Firefox si chiama Geko ed è molto ben documentato:

<https://developer.mozilla.org/en-US/docs/Mozilla/Gecko>

Una conseguenza dell'esistenza di questo modello di **reflow**, è il fatto che diventa possibile prevedere un comportamento dinamico mediante il quale un utente sia in grado di modificare selettivamente parti di una pagina web. È possibile infatti attivare uno script che, ad un certo istante, richieda dei nuovi contenuti al server, questi li fornisca e il DOM in memoria venga aggiornato con essi. Il reflow provvederà automaticamente a renderli disponibili determinando il comportamento dinamico richiesto.

Quante e quali sono le componenti che concorrono alla creazione di una pagina web dal comportamento dinamico articolato? La risposta è: **tante** (anche se dipende dalla particolare funzionalità di cui si vuole dotare la pagina).

- A livello minimale, **HTML** (<http://www.w3.org/html/>)
- Per una gestione razionale dell'aspetto estetico delle pagine, **CSS** (Cascading Style Sheets) (<http://www.w3.org/Style/CSS/Overview.en.html>)
- Per dotare le pagine di comportamento dinamico, ad un livello medio-basso:  
**JavaScript** in modo nativo (senza l'uso di frameworks o librerie)  
(<https://developer.mozilla.org/en/docs/Web/JavaScript>)
- L'uso di JavaScript comporta la conoscenza, a qualche livello, di tecnologie quali
  - ✓ **XHTML** (<http://www.w3.org/TR/xhtml1/>) (deprecato a favore di HTML5)
  - ✓ **XML** (<http://www.w3.org/XML/>)
  - ✓ **JSON** (<http://json.org/>)
  - ✓ **AJAX** (<https://developer.mozilla.org/en/docs/AJAX>)
  - ✓ **SVG** (<https://developer.mozilla.org/en/docs/Web/SVG>)
  - ✓ **Node.js** (<http://nodejs.org>)

Torniamo ora al nostro esempio e vediamo di dotare la pagina di una qualche forma di dinamicità (variabilità nel tempo).

Vorremmo, ad esempio, che al posto della scritta **Welcome!** vi fosse un orologio digitale, che indichi lo scorrere del tempo e che, 15 secondi dopo aver caricato la pagina questo si fermi e compaia infine al suo posto la scritta di benvenuto.

Per fare ciò dovremo usare delle componenti di JavaScript che ci permettano di identificare parti della pagina WEB nel DOM che lo rappresenta in memoria (ad esempio l'oggetto in cui la scritta è memorizzata), e parti che sappiano poi manipolare quella zona di memoria sostituendone il contenuto con un altro. Vorremmo inoltre che questa operazione fosse schedulata avvenire periodicamente per i primi 15 secondi soltanto, andando poi a termine.

Per fare ciò introduciamo nella nostra pagina HTML delle parti che sono codice JavaScript da eseguire.

File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_4.html 1351 bytes L: --- C: ---

```
6     display : block ;
7     top      : 10px ;
8     left     : 20px ;
9     width    : 100px ;
10    position : relative ;
11    border   : 1px solid black ;
12    padding   : 10px ;
13    text-align: center ;
14    font-weight: bold ;
15    font-family: sans-serif ;
16    color     : #0000ff ;
17 }
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23 This is page one... (file one_4.html)
24 <div style="position: relative; top: 30;">
25   <hr/>
26   Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
27   </div>
28 </div>
29
30
31 <div id="menu" style="float: top;">
32   Welcome!!!
33 </div>
34
35 <script>
36   var divMenu = document.getElementById("menu") ;
37   var a = 0 ;
38 </script>
39
40 <div style="position: relative; top: 900;">-</div>
41 </body>
42 </html>
```

Utilizziamo un nuovo tag, **script**, che ci permette di introdurre un programma in JavaScript nella pagina HTML stessa.

La variabile **divMenu** conterrà una referenza al contenuto dell'oggetto che rappresenta il blocco **menu** in memoria.

Two Mozilla Foundation (US) https://addons.mozilla.org/en-US/firefox/addon/firebu mozilla firebug

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Register or Log in Other Applications mozilla

**ADD-ONS**

EXTENSIONS | THEMES | COLLECTIONS | MORE...

search for add-ons

Extensions » Firebug

## Firebug 1.12.8 NO RESTART

by Joe Hewitt, Jan Odvarko, robcee, FirebugWorkingGroup

Firebug integrates with Firefox to put a toolbar in the browser window while you browse. You can edit, debug, and analyze any web page...

+ Add to Firefox

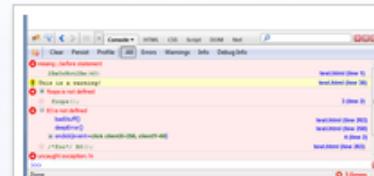


1,627 user reviews

Prima di procedere oltre è utile scaricare ed installare un plugin di Firefox **fondamentale** per lo sviluppo di codice interattivo su WEB:

## Firebug

(anche se ne esiste una versione simile in **WebDeveloper** che abbiamo già installato)





Two



Google



mobile.mib.infn.it/~menasce/esempi/one\_3.html

This is page one... (file one\_3.html)

Welcome!!!

Attiviamo il debugger...

Vediamo ora come esaminare l'esecuzione  
del programma passo dopo passo e accedere  
al valore delle variabili ad ogni istante

liqua.

e irure

non

proident,

sunt in culpa qui officia deserunt mollit anim id est laborum."

;

© Peter Cui Bide | petercui.deviantart.com

File Edit View History Bookmarks Tools Help

Two +

mobile.mib.infn.it/~menasce/esempi/one\_4.html

Google

This is page one... (file one\_4.html) **Welcome!!!**

© Peter Cat Tide | petercat.deviantart.com

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT, SED DO EIUSMOD TEMPOR INCIDIDUNT UT LABORE ET DOLORE MAGNA ALIQUA. UT ENIM AD MINIM VENIAM, QUIS NOSTRUD EXERCITATION ULLAMCO LABORIS NISI UT ALIQUIP EX EA COMMODO CONSEQUAT. DUIS AUTEM IRURE DOLOR IN REPREHENDERIT IN VOLUNTATE VELIT ESSE CILLUM DOLORE EU FUGIAT NULLA PARIATUR. EXCEPTEUR SINT OCCAECAT CUPIDATAT NON

Firebug - Two

Console HTML CSS Script DOM Net Cookies

Edit html

<html>

Style Computed Layout DOM

This element has no style rules. You can [create a rule](#) for it.

Modifichiamo ora il codice per cambiare periodicamente valore al testo di saluto

File Edit Search Preferences Shell Macro Windows Help

/user/gr1/e831/menasce/public\_html/esempi/one\_4.html byte 1496 of 1496 L: 51 C: 0

```
10      position : fixed ;
11      border   : 1px solid #0f0;
12      padding  : 10px;
13      text-align: center;
14      font-weight: bold;
15      font-family: arial;
16      color    : #f0f;
17  }
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23   This is page one... (file one_4.htm
24   <div style="position: relative; top
25     <hr/>
26     Lorem ipsum dolor sit amet, conse
27     elit, sed do eiusmod tempor incidi
28   </div>
29 </div>
30
31 <div id="menu" style="float: top;">
32   Welcome!!!
33 </div>
34
35 <script>
36   var divMenu = document.getElementById("menu") ;
37   setInterval(
38     function()
39     {
40       console.log("Stepping inside loop...") ;
41     },
42     1000
43   ) ;
44
45 </script>
46
```

La funzione **setInterval** (puro JavaScript) esegue la funzione specificata in *calling sequence*, ripetendola periodicamente con una certa frequenza, che è possibile specificare come secondo argomento (in millisecondi).

La funzione **console** da accesso alla console di **Firebug** (il metodo log scrive la corrispondente stringa sulla console medesima).

File Edit View History Bookmarks Tools Help

Two mobile.mib.infn.it/~menasce/esempi/one\_4.html Google

This is page one... (file one\_4.html) Welcome!!!

Firebug - Two

Console HTML CSS Script DOM Net Cookies

Use #<number> to go to line

all | one\_4.html | one\_4.html

```
13     text-align : center;
14     font-weight : bold;
15     font-family : arial;
16     color      : #f0f;
17   }
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23   This is page one... (file one_4.html)
24   <div style="position: relative; top: 30;">
25     <hr/>
26     Lorem ipsum dolor sit amet, consectetur adipisci
27     elit, sed do eiusmod tempor incididunt ut labor
28   </div>
29 </div>
30
31   <div id="menu" style="float: top;">
32     Welcome!!!
33   </div>
34
35   <script>
36     var divMenu = document.getElementById("menu");
37     var a = 0;
38   </script>
39
40   <div style="position: relative; top: 900;"></div>
41 </body>
42 </html>
```

Watch Stack Breakpoints

previousElementSibling  
remove  
baseURI  
childNodes  
data  
firstChild  
lastChild  
length  
nodeName  
nodeType  
nodeValue  
parentElement  
parentNode  
prefix  
previousSibling  
textContent  
wholeText  
addEventListener

null  
remove()  
"http://mobile.mib.infn....nasce/est/one\_4.html"  
NodeList[ ]  
"  
"Welcome!!!"  
"  
null  
null  
15  
div#menu  
null  
null  
"\n" "Welcome!!!\n" "  
"\n" "Welcome!!!\n" "  
addEventListener()

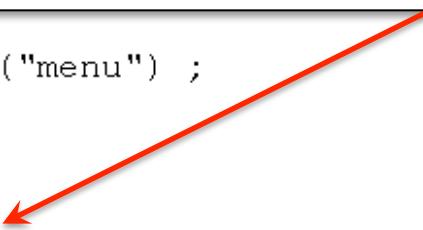
Torniamo al codice e, nella parte ciclica istanziamo il codice che ci interessa (data ed ora ripetuti ogni secondo per i soli primi 15 secondi).

>>>

```
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23   This is page one... (file one_4.html)
24   <div style="position: relative; top: 60;">
25     <hr/>
26     Lorem ipsum dolor sit amet, consectetur adipisicing
27       elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut eni
28   </div>
29 </div>
30
31   <div id="menu" style="float: top;">
32     Welcome!!!
33   </div>
34
35 <script>
36   var divMenu = document.getElementById("menu") ;
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53   </script>
54
```

Spostiamo il testo di **lorem ipsum** un po' più in basso...

All'interno del timer istanziamo un nuovo oggetto **Date** ad ogni iterazione temporale



```
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23 This is page one... (file one_4.html)
24 <div style="position: relative; top:
25 <hr/>
26 Lorem ipsum dolor sit amet, consectetur
27 elit, sed do eiusmod tempor incididunt
28 </div>
29 </div>
30
31 <div id="menu" style="float: top;">
32 Welcome!!!
33 </div>
34
35 <script>
36 var divMenu = document.getElementById("menu") ;
37
38 var interval = setInterval(
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54 ) ;
```

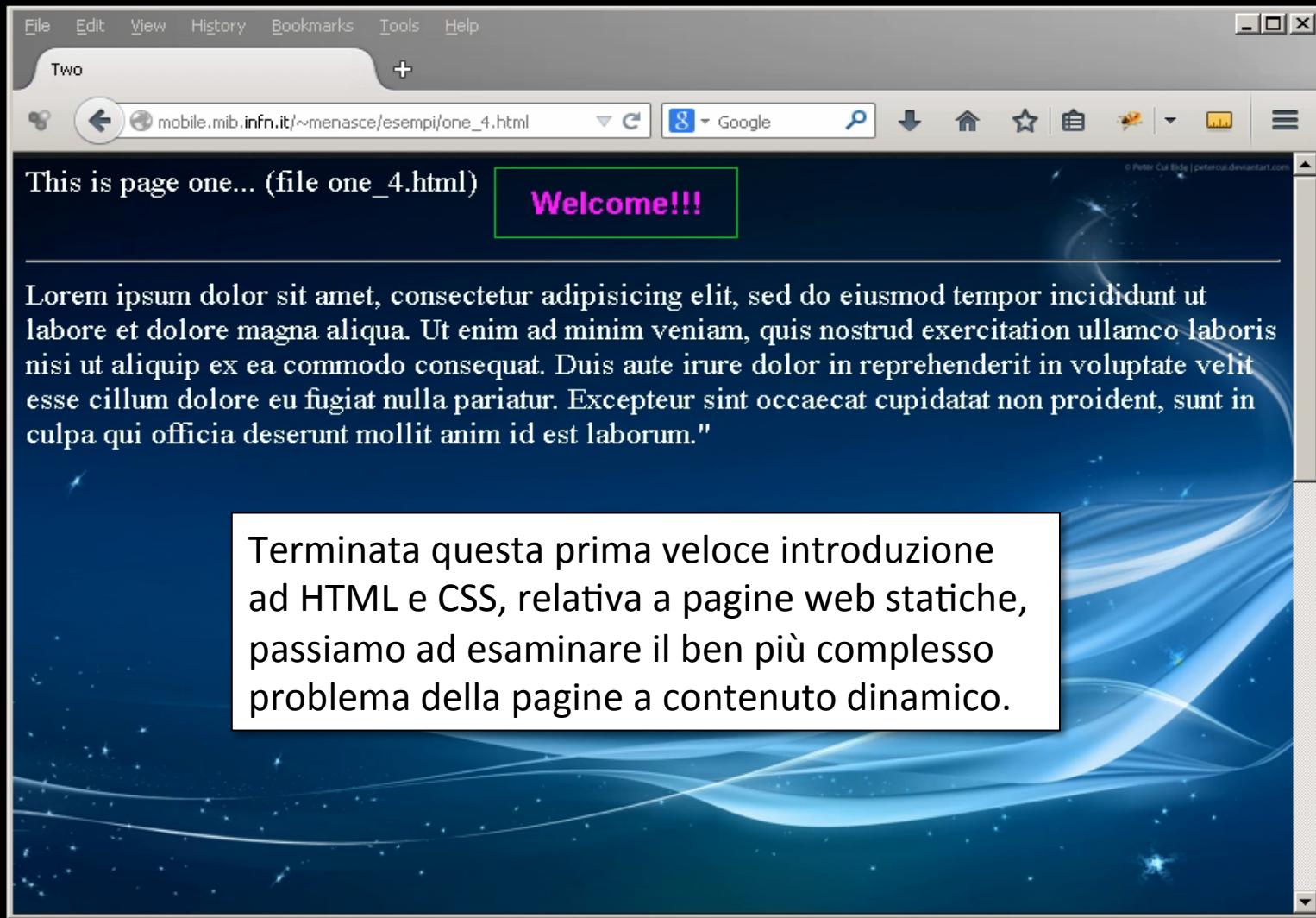
Questo è il punto centrale: la variabile **innerHTML** contiene il blocco dati del **div menu**. Con questa operazione lo sostituiamo periodicamente. Il *rendering engine* di Firefox provvede poi al refresh sullo schermo.

```
        function()
        {
            var date = new Date() ;
            divMenu.innerHTML = " ] " + date ;
```

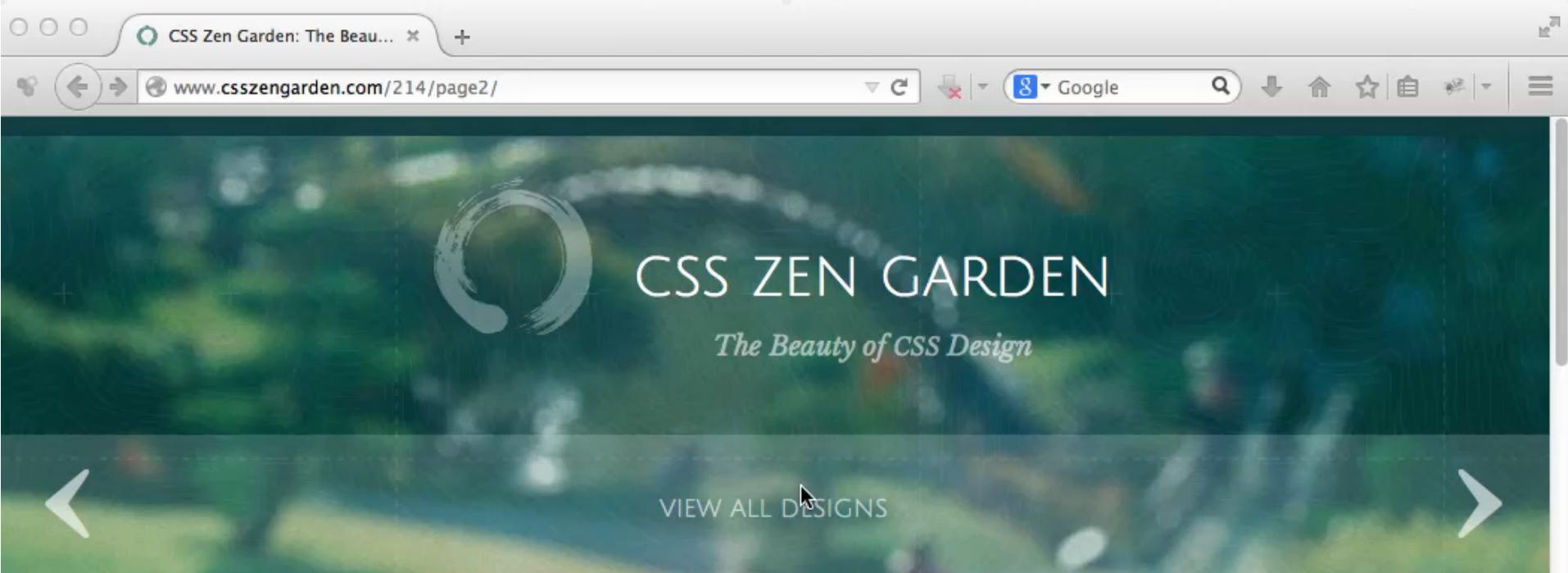
Questo iteratore però prosegue eternamente: se lo vogliamo fermare dopo un certo numero di iterazioni lo dovremo fornire di un meccanismo adeguato allo scopo:

```
18 </style>
19
20
21 <body background="icons/image.png">
22 <div style="color: white">
23   This is page one... (file one_4.html)
24   <div style="position: relative; top: 60;">
25     <hr/>
26     Lorem ipsum dolor sit amet, consectetur adipisicing
27     elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut eni
28   </div>
29 </div>
30
31 <div id="menu" style="float: top;">
32   Welcome!!!
33 </div>
34
35 <script>
36   var divMenu = document.getElementById("menu") ;
37   .
38   var interval = setInterval(
39     .
40     .
41     function()
42     {
43       var date = new Date() ;
44       divMenu.innerHTML = start + "] " + date ;
45     }
46     ,
47     1000
48   ) ;
49
50
51
52
53 </script>
54
```

Questo iteratore però prosegue eternamente:  
se lo vogliamo fermare dopo un certo numero  
di iterazioni lo dovremo provvedere con un  
adeguato meccanismo:



Terminata questa prima veloce introduzione ad HTML e CSS, relativa a pagine web statiche, passiamo ad esaminare il ben più complesso problema delle pagine a contenuto dinamico.



A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [HTML FILE](#) and [CSS FILE](#)

## THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

# Fine prima parte

*“That's all Folks!”*

Isberg<sup>®</sup>

Vai a:

[Parte 2](#)