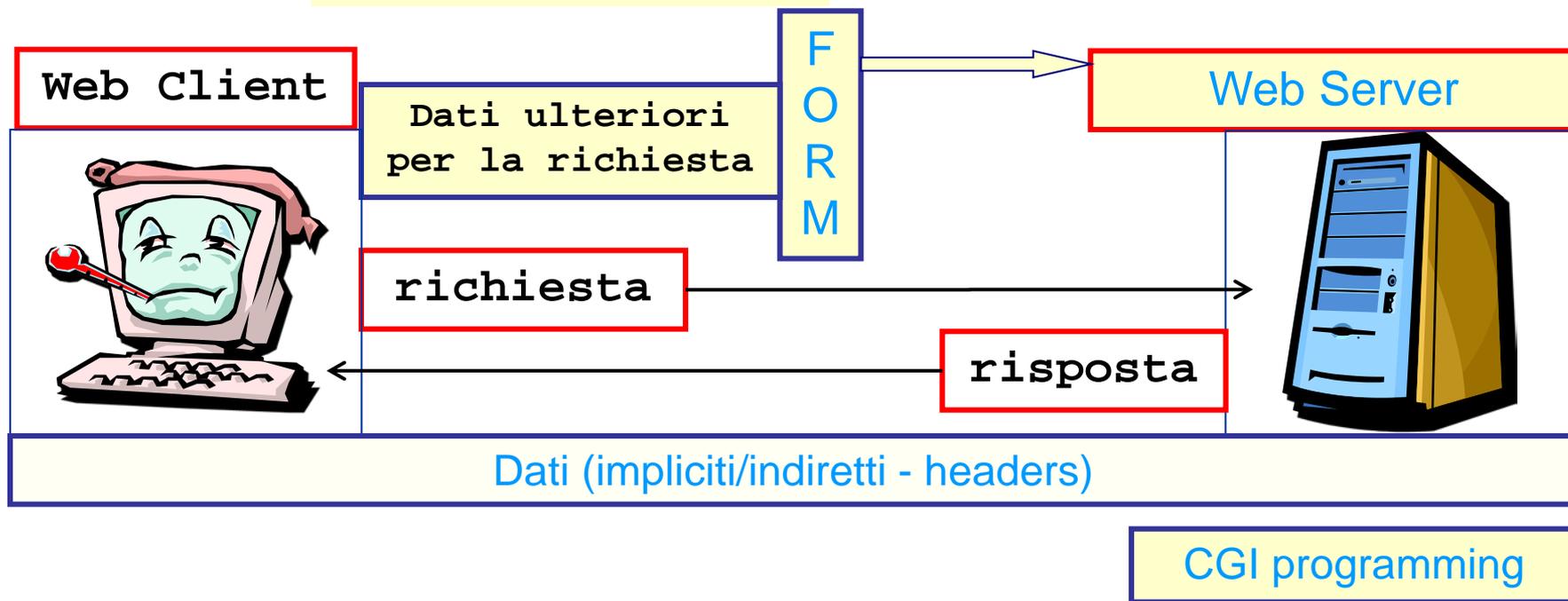


# Web Server; FORM; e prima Programmazione

`http://...it/scr.pl?dati`



(Client Side)

Programming

(Server Side)

la richiesta e la risposta possono essere “prodotti” del calcolo eseguito da opportuni programmi;

il client deve poter aggiungere dei dati alla sua richiesta, per permettere il calcolo;

i programmi possono essere in funzione sia sul server (server side) che sul client (client side)

# Common Gateway Interface

Meccanismo per la gestione, da parte del web server, di programmi/script eseguiti su server.

Dopo la richiesta da parte del browser, la risposta potrebbe essere una risorsa prodotta da un programma (risorsa dinamica) invece che una risorsa statica.

perche'? tecnologia stabile e essenzialmente browser independent; molti linguaggi tradizionali: shell-unix, batch-dos/win, Tcl, C,C++, java, javascript, Visual Basic, python, Ruby ...

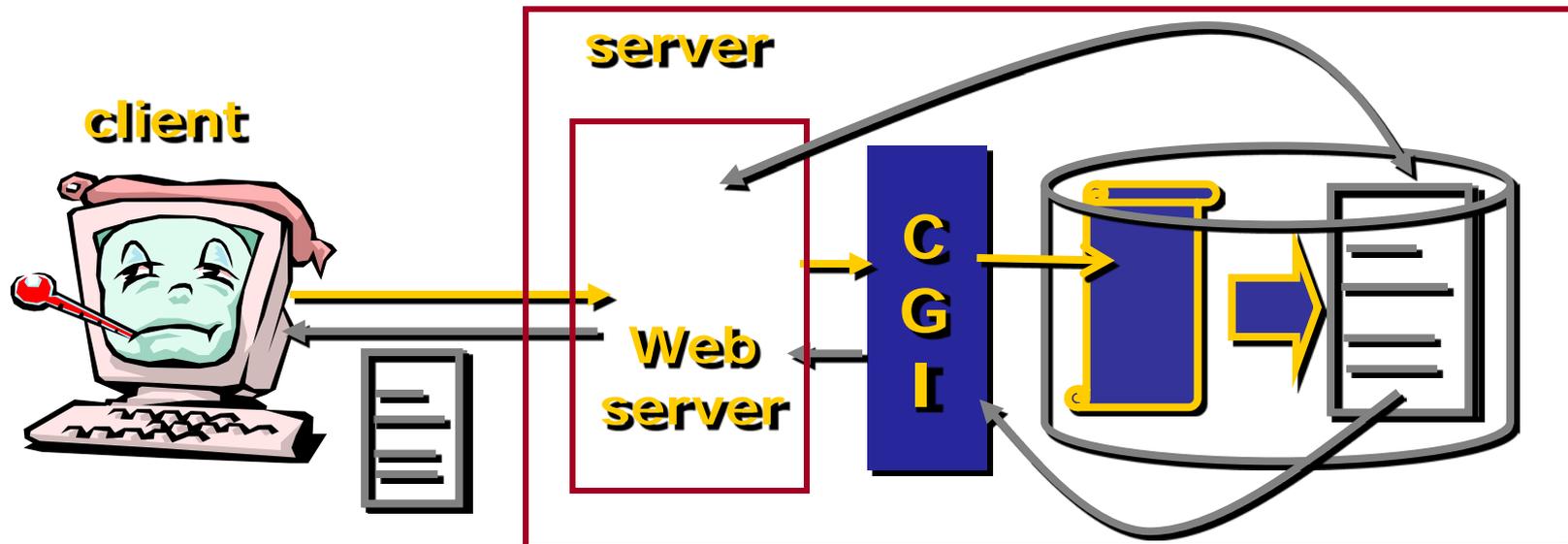
Scopi? searching, db access, form handling

Alternative? .NET, PHP, Perl, Python ... java (jsp)

Il programma cgi gestisce informazioni quali

- **get/post data**
- **cookies**
- **var ambiente client**
- **var ambiente server**
- risorse del server (file system, db)

e produce un output: la risorsa che viene inviata con la risposta



# Configurazione server web (apache)

```
ServerRoot "C:/Apache"  
Port 80  
ServerAdmin admin@localhost
```

Dove sono i file del server; quale porta e` tenuta in osservazione per ricevere richieste; indirizzo web master

```
ServerName localhost  
DocumentRoot "C:/Apache/htdocs"
```

http://localhost sara` l'indirizzo (solo per usi locali);  
http://127.0.0.1 = accesso con ip (sempre usi locali)  
con questi accessi, la directory di default in cui cercare i documenti e` la DocumentRoot

```
UserDir "\un\certo\path\DocumentDir"  
DirectoryIndex index.html index.php default.php
```

http://localhost/~marte  
accesso nella  
C:\...\Documents\marte\

```
AccessFileName .htaccess  
Alias /icons/ ".../.../icons/"
```

File che, se e` in una dir, ne governa i diritti di accesso;  
http://localhost/icons/ e` un alias per la dir C:.../icons/

```
ScriptAlias /cgi-local-bin/  
".../Documents/localhtdocs/LOCALCGI/"  
ScriptAlias /cgi-bin/ "C:/apache/cgi-bin/"
```

Dove sono gli script che potranno essere eseguiti (che devono essere localizzati in precise dir. per questioni di "sicurezza")

```
IndexOptions FancyIndexing
```

cgi-local-bin (dir aggiuntiva)  
cgi-bin default

```
AddIconByType (TXT,/icons/text.gif) text/*  
DefaultIcon /icons/unknown.gif
```

uno script si attivera` alla richiesta

<http://localhost/cgi-bin/nome.ext>

`httpd.conf` (e sue scomposizioni)

# Configurazione server web (apache)

```
ServerRoot "C:/Apache"  
Port 80  
ServerAdmin admin@localhost  
  
ServerName localhost  
  
DocumentRoot "C:/Apache/htdocs"  
  
UserDir "\un\certo\path\DocumentDir"  
  
DirectoryIndex index.html index.php default.php  
  
AccessFileName .htaccess  
  
Alias /icons/ ".../.../icons/"  
  
ScriptAlias /cgi-local-bin/  
".../Documents/localhtdocs/LOCALCGI/"  
ScriptAlias /cgi-bin/ "C:/apache/cgi-bin/"  
  
IndexOptions FancyIndexing  
  
AddIconByType (TXT,/icons/text.gif) text/*  
DefaultIcon /icons/unknown.gif
```

directory di installazione/configurazione

apache2.conf	general conf
conf-available/	available additional ...
conf-enabled/	symlinks to some additional conf files
mods-available/	available modules
mods-enabled/	active modules among the available
ports.conf	ports listening mngmt
sites-available/	conf for special sites on the ws
sites-enabled/	active ...

[httpd.conf](#)

(e sue scomposizioni)

# Configurazione server web (esperienze sparse) ...

con la propria installazione del web server, e seguendo la traccia data dal file `config.apache.pillole.txt`, fare un po' di esperienze con la configurazione del web server

*(vedi in fondo alle slide per possibili scelte per il web server apache)*

Dopo aver installato il webserver, inserire i propri esercizi XHTML/CSS nella document root (es `htdocs`); poi accedere alla document root via web, o con il nome simbolico (`localhost`) o con il numero ip (`127.0.0.1`) predefinito per gli usi locali

Inserire i propri esercizi XHTML-CSS nella dir. `htdocs` del server (e/o sottodirectory, come faccio io con `MARCO-HTDOCS` ed accedervi via web con `http://localhost/...`

**Qui ci vuole un po' di pazienza ...** Creare una propria directory `localhtdocs`, alternativa a quella di default (cioè quella predefinita come `"htdocs"`, o `"www"`) e raggiungerla con il browser (il web server deve essere attivo ...)

**Anche qui ci vuole un po' di pazienza**

Sperimentare l'uso della protezione basata su `.htaccess`

Idem per `.htpasswd` (directory privato)

**Se fate questo siete al top ;)** Definire una directory `cgi-local-bin`, in cui possano funzionare script cgi, in alternativa alla directory standard `cgi-bin` (in effetti la nuova definizione si aggiunge esattamente lì dove quella standard è già presente ...)

# FORM

Sono lo strumento per fornire informazioni esplicite al server (inserite dall'utente per personalizzare la richiesta)

Se si vuole una risorsa statica, basta indirizzare una richiesta al server specificando la uri della risorsa (una pagina statica, una figura, un suono).

Se quel che si vuole e` invece qualcosa di "piu` personalizzato" (ad es. una pagina con informazioni diverse a seconda di chi e come le richiede) allora la richiesta deve essere corredata da informazioni che permettano al server di rispondere secondo le attese.

Ogni interazione http (richiesta-risposta) puo` coinvolgere informazioni di due tipi

- informazioni **implicite** (dati di ambiente) ad es.

```
REQUEST_URI = /cgi-local-bin/bat/provabat2.bat
REMOTE_ADDR = 127.0.0.1          REQUEST_METHOD = GET
CONTENT_TYPE =          CONTENT_LENGTH =          HTTP_HOST = localhost
HTTP_USER_AGENT = Mozilla/5.0 ... Gecko/20030208 Netscape/7.02
HTTP_ACCEPT =text/xml,...,text/html; image/png,image/jpeg,
HTTP_ACCEPT_CHARSET = ISO-8859-1, utf-8;q=0.66, *;q=0.66
```

- informazioni **esplicite** (dati tout court)

QUERY\_STRING

stdin

In caso di richiesta **POST**, il programma che gira sul server avra` i medesimi dati url-encoded leggendoli dallo stream di input standard. **stdin!!**

In caso di richiesta **GET**, la var. d'ambiente QUERY\_STRING contiene i dati trasmessi dal client in formato *url-encoded*

nome=valore&nome=valore&nome=valore

es.

primonome=Marco&cognome=Temperini&S1=In  
via+nome+e+cognome+%28e+riottienili%29

(	space	)	=	+	/
%28	+	%29	%3D	%2B	%2F

L'elemento form contiene elementi per esprimere coppie <NOME\_DEL\_DATO, DATO> in modo che sia possibile

- al browser visualizzare l'elenco di campi da "riempire" (ognuno associato ad un nome)
- all'utente inserire il dato in corrispondenza del campo (e così il dato è associato al nome)
- al browser inviare il pacchetto di dati (con i nomi associati) al server, in una richiesta creata dal "submit" (Invio). La richiesta in pratica chiede l'esecuzione di un programma (es. cgi) sui dati inviati ...

**<form ...>** elementi della form **</form>**

**action** la URI del programma cgi (o, in seguito, dello script PHP per noi) che processerà i dati sul server

**method** specifica come i dati verranno trasmessi

**get:** dati appesi alla URI dell'action, dopo un ?

**post:** dati inviati allo stdin del server

**enctype** (`application/x-www-form-urlencoded`),  
**accept-charset...**

```
<form action="http://localhost/cgi-bin/bat/cgiFile/provabat2.bat" method="get">
<p>Nome :
...
</form>
```

una form da gestire

file:///C:/Users/marco/Desktop/MARCO/MARCO/...

Nome:

Cognome:

Preferenze calcistiche:

- Roma
- Latina
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

- Rosa
- Agliola delle bisce
- Anemone Epatica
- Agatea
- Amaranto

Annulla le scelte    Invio1



```
secondo programmino batch d X +
localhost/cgi-bin/bat/cgiFile/provabat2.bat?nome=Susanna&nome=Susanna&calc=Roma&gen

Inizio secondo programma batch di prova

Un po' di variabili d'ambiente.

HTTP_USER_AGENT = "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0"
HTTP_ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_ENCODING = gzip, deflate
HTTP_ACCEPT_LANGUAGE = en-US,en;q=0.5
HTTP_ACCEPT_CHARSET =

HTTP_CONNECTION = keep-alive
HTTP_HOST = localhost

SERVER_ADMIN = postmaster@localhost ----- SERVER_URL =
SERVER_NAME = localhost ----- SERVER_PORT = 80
SERVER_PROTOCOL = HTTP/1.1
SERVER_SOFTWARE = Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
DOCUMENT_ROOT = C:/xampp/htdocs
```

**GET** Alla pressione di Inviol, viene attivata la richiesta con URI  
`http://localhost/cgi-bin/ ...`  
`?nome=Susanna&nome=Susanna&calc=Roma&generi=Gialli&generi=Fantascienza&fiori=Anemo`  
`ne&fiori=Agatea&invio=Inviol`

**POST** Alla pressione di Inviol, viene attivata la richiesta con URI  
`http://localhost/cgi-bin/...`

# FORM - <form>

```
<form action="http://localhost/cgi-bin/printenv.pl" method="get">
<p>Nome :
<input type="text" name="nome" value="Susanna" size="30">
</p>
<p>Cognome :
<input type="password" name="cognome" size="30">
</p>
...
...
<p>
<input type="reset" value="Annulla le scelte">
<input type="submit" name="invio" value="Invio1">
</p>
</form>
```

una form da gestire

Nome: Susanna

Cognome: ●●●●●●

Pre assegnata ma modificabile

Preferenze calcistiche:

- Roma
- Latina
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

- Rosa
- Agliola delle bisce
- Anemone Epatica
- Agatea
- Amaranto

Annulla le scelte Invio1

QUERY\_STRING  
nome=Susanna&cognome=Abbacchi&...&invio=Invio1

# FORM - <input>

Un elemento <input> permette di specificare diversi tipi di dati:

- text** casella di testo
- password** casella testo, con caratteri mascherati
- radio** opzioni (a scelta singola)
- checkbox** opzioni a scelta multipla
- select/option**
- reset** un bottone per azzerare i campi
- submit** e' il bottone di azione

File Edit View History Bookmarks Tools Help

una form da gestire

file:///C:/xampp/htdocs/LWEB/ESPERIENZE-CGI/altro/form di prova con batch/formex.html

Nome: Susanna

Cognome: ●●●●●●

Preferenze calcistiche:

- Roma
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

- Rosa
- Anemone
- Viola
- Azalea
- Margherita

Annulla le scelte Invio1

**hidden** non visualizzato dal browser: nasconde dati che arriveranno al server

# Cosa vediamo

un po' di introduzione sulle FORM

Alcuni esempi pratici di pagine web che fanno uso di script o programmi tramite CGI

(sta tutto nella directory ESPERIENZE-CGI)

- SuBatch.html
- SuC.html
- minisito indovinelli
- forms (anche questa a fianco)

versione 2 della form da gestire con

quando la form viene inviata dall'utente, la funzione di gestione di headers e permette di

qui la form e' un po' piu' curata ... vale la pena di vederla

Nome:

Cognome:

Preferenze calcistiche:

- Roma
- Latina
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

# FORM - <input> - 2

```
...<p>Preferenze calcistiche:<br />
<input type="radio" name="calc"
      value="Roma"> Roma<br />
<input type="radio" name="calc"
      value="Inter"> Inter<br />
...</p>

<p>Preferenze letterarie:           <br />
<input type="checkbox" name="lettere"
      value="Gialli"> Gialli           <br />
<input type="checkbox" name="lettere"
      value="Fantascienza"> Fantascienza<br />
<input type="checkbox" name="lettere"
      value="Horror"> Horror           <br />
</p>
<p>
Scegliere tra i seguenti fiori (anche pi&ugrave;
uno):
<select name="fiore" size="5" multiple="multi
<option value="Rosa">Rosa</option>
... 6 altre ...
<option value="Lilium">Lilium</option>
<option value="Nasturzio">Nasturzio</option>
<option value="Papavero">Papavero</option>
</select>
</p>...
```



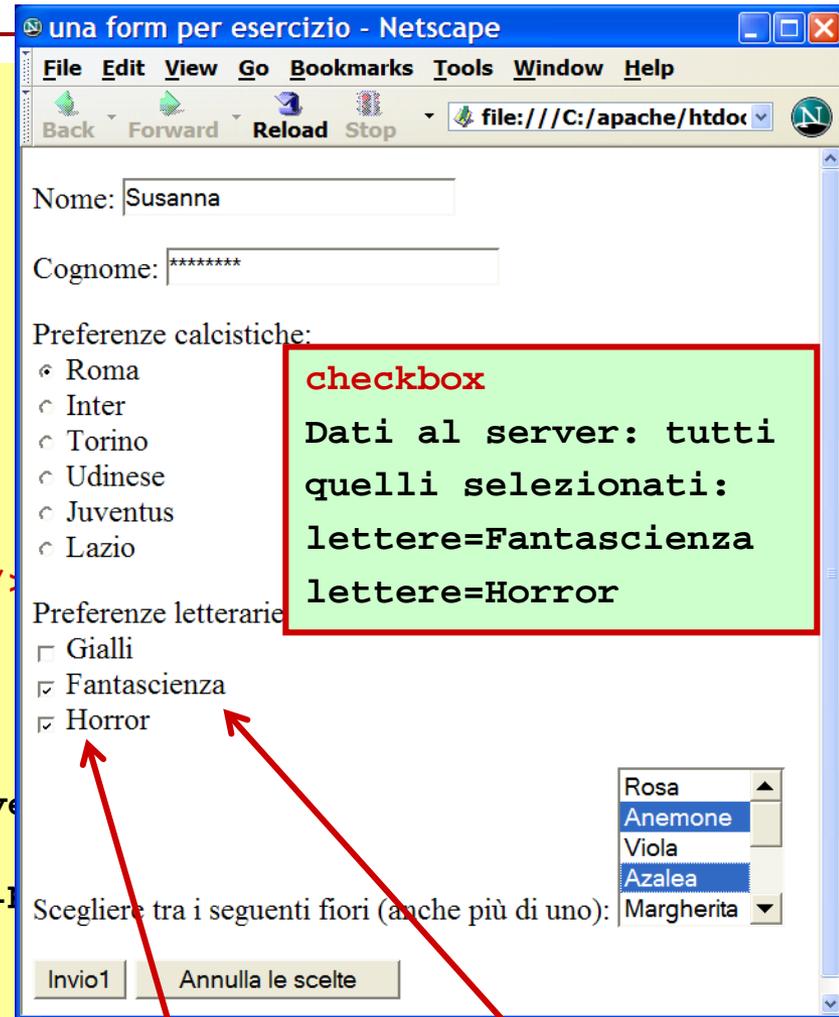
## QUERY\_STRING

```
...&calc=Roma&lettere=Fantascienza
&lettere=Horror&fiore=Anemone&fior
e=Azalea&invio=Invio1
```

# FORM - <input> - 3

```
...<p>Preferenze calcistiche:<br />
<input type="radio" name="calc"
      value="Roma"> Roma<br />
<input type="radio" name="calc"
      value="Inter"> Inter<br />
...</p>

<p>Preferenze letterarie:           <br />
<input type="checkbox" name="lettere"
      value="Gialli"> Gialli           <br />
<input type="checkbox" name="lettere"
      value="Fantascienza"> Fantascienza<br />
<input type="checkbox" name="lettere"
      value="Horror"> Horror           <br />
</p>
<p>
Scegliere tra i seguenti fiori (anche pi&ugrave;
uno):
<select name="fiore" size="5" multiple="multi
<option value="Rosa">Rosa</option>
... 6 altre ...
<option value="Lilium">Lilium</option>
<option value="Nasturzio">Nasturzio</option>
<option value="Papavero">Papavero</option>
</select>
</p>...
```



**checkbox**  
Dati al server: tutti  
quelli selezionati:  
lettere=Fantascienza  
lettere=Horror

**QUERY\_STRING**  
...&calc=Roma&lettere=Fantascienza  
&lettere=Horror&fiore=Anemone&fiore  
=Azalea&invio=Invio1

# FORM - <input> - 4

Un'immagine come submit button

```
<p>
<input type="image"
      src="img/ruote.dentate.gif" name="elaborazione">
</p>
```

**image**

oltre all'attivazione della action: al server le coordinate del click

bottone generico

```
<p>
<input type="button" value="spingiQui" onclick= ... >
</p>
```

**button**

non attiva action (e allora, evento onclick + javascript)

```
<input type="hidden" name="titoloCorso"
      value="Fond.Inf.II">
```

**Campo hidden**

serve a far arrivare al server informazioni incluse nel

documento form ma non visualizzate sul browser ("dati sulla form"). Chi ha incluso queste informazioni? Magari il server stesso, quando ha fornito al browser il documento form, inserendoci dentro alcune "informazioni di contesto" da consultare quando la form verra` attivata (ad es. perche' l'interazione tra client e server avviene con lo scambio di diverse form e bisogna mantenere delle informazioni di stato tra l'una e l'altra)

# FORM - <select>

Un elemento <select> contiene una sequenza di opzioni, rese come un menu` (scrollabile o a tendina). L'attributo **nome** fornisce il nome per i dati prodotti. Con **multiple="multiple"**, si possono selezionare piu` valori del menu`.

Una opzione e` un elemento <option>, il cui attributo **value** specifica il valore da dare a **nome** se quella opzione e` selezionata. Per <option>,  
**selected="selected"** rende selezionata per default l'opzione (similmente a **checked="checked"** delle checkbox o radio button);  
**disabled="disabled"** disabilita l'opzione (come per gli elementi <input>).

```
<p>  
Scegliere tra i seguenti fiori (anche  
piu` di uno):  
<select name="fiore"  
      size="5" multiple="multiple">  
<option value="Rosa">Rosa</option>  
... 6 altre ...  
<option value="Lilium">Lilium</option>  
<option value="Nasturzio">Nasturzio</option>  
<option value="Papavero">Papavero</option>  
</select>  
</p>
```

una form per esercizio - Netscape

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop file:///C:/apache/htdocs

Nome:

Cognome:

Preferenze calcistiche:

- Roma
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

- Rosa
- Anemone
- Viola
- Azalea
- Margherita

Invio1 Annulla le scelte

Dati al server: tutti  
quelli selezionati:  
fiore=Anemone  
fiore=Azalea

## QUERY\_STRING

```
...&calc=Roma&lettere=Fantascienza  
&lettere=Horror&fiore=Anemone&fiore=Azalea&invio=Invio1
```

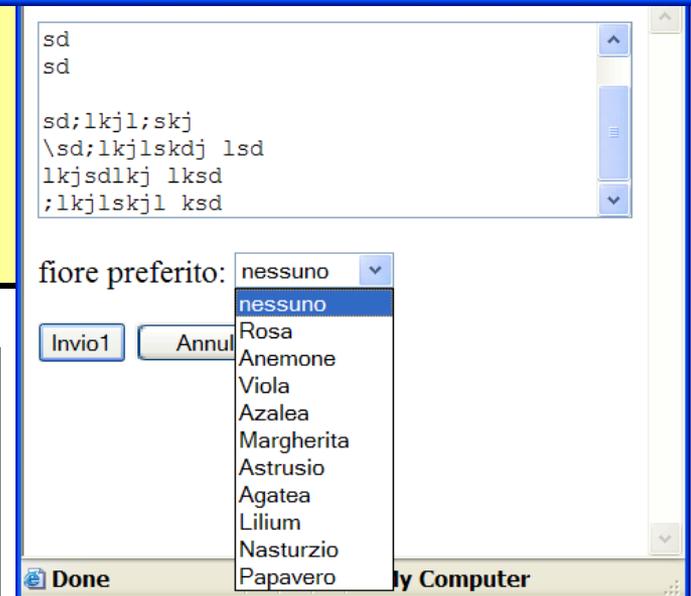
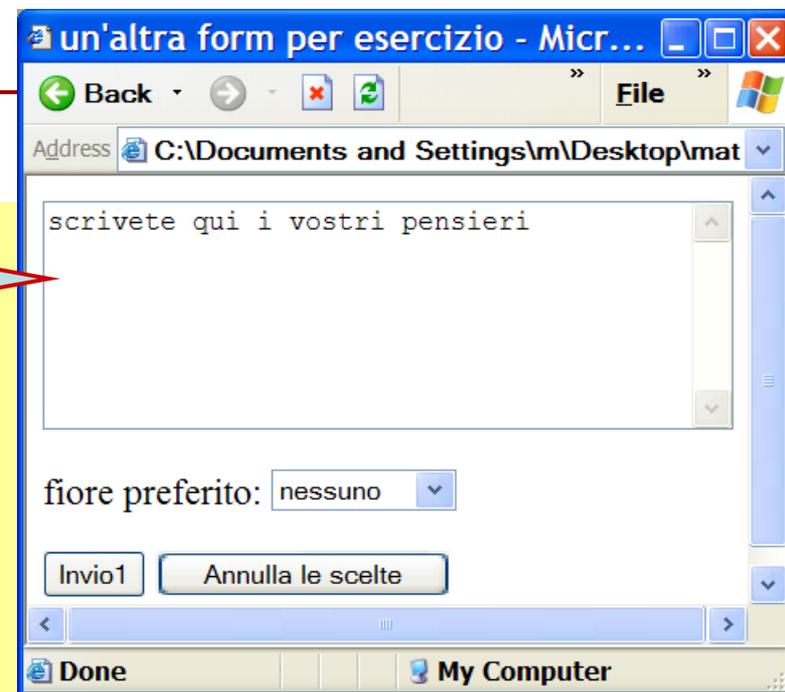
# FORM - <select> + <textarea>

```
<textarea rows="7" cols="40" name="pensieri">
scrivete qui i vostri pensieri
</textarea>
```

```
<p>
fiore preferito:
<select name="fiore">
<option value="snarl">nessuno</option>
<option value="Rosa">Rosa</option>
... 6 altre ...
<option value="Lilium">Lilium</option>
<option value="Nasturzio">Nasturzio</option>
<option value="Papavero">Papavero</option>
</select>
</p>
```

## QUERY\_STRING

```
http://localhost/cgi-local-bin/c/PROVA2C
.EXE?pensieri=scrivete+qui+i+vostri+pensie
ri...&fiore=snarl&invio=Invio1
```



# Risorse

## web server

- **www.apache.org** <http://httpd.apache.org>,  
es. <http://httpd.apache.org/docs/2.4/>  
<https://httpd.apache.org/docs/2.4/howto/cgi.html>  
<http://httpd.apache.org/docs/current/mod/core.html> (apache core features, per il  
significato di ciascuna parola che appare nell'httpd.conf ...)

- **XAMPP, easyphp, wamp; mamp**  
(applicazioni che permettono di installare e gestire “in un colpo solo” apache, php con  
mysql/Mariadb e magari un smtp o ftp server – verificare le configurazioni di default)

## Directory pubblica

- **config.apache.pillole.txt**
- **README-ESPERIENZE-CGI.txt**
- **ESPERIENZE-CGI/bat/**                    1) **SuBatch.html**                    2) **cgiFile**
- **ESPERIENZE-CGI/c/**                    1) **SuC.html**                    2) **cgiFile**
- **ESPERIENZE-CGI/formex/**
- **ESPERIENZE-CGI/cgi-indovinelli/**
- **un\_esempio\_di\_htaccess/**

tornando indietro per un momento ... una lettura interessante e` qui:

<http://www.webdevout.net/articles/beware-of-xhtml>

Come viene usato XHTML, benefici dell'approccio XML, compatibilita` con HTML,  
problem con browsers ...

Per affezionati

# Pratica (anche per linux-unix oriented students)

## *Usa dei file di esempio visti a lezione (repetita juvant)*

- sono disponibili in ESPERIENZE-CGI.zip
- se si usa la propria macchina, e il web server in essa installato, i programmi possono essere messi nella dir cgi-bin (nella directory di installazione del web server).

*In tutti i casi, gli esempi contenuti in SuC.html, SuBatch.html, indovinelli, vanno modificati, per niente/poco/quite, in modo che si faccia riferimento ai programmi usando i path giusti rispetto al proprio file system.*

*A lezione i percorsi codificati nei programmi C, ad esempio, erano relativi a MARCO\_HTDOCS/ESPERIENZE\_CGI/... e ovviamente vanno cambiati.*

*Riguardo agli script "bat", questi evidentemente funzionano solo nei sistemi basati su windows.*

*Ma si possono sempre modificare, scrivendoli nel linguaggio di scripting del sistema linux (es. bash), o dello unix sottostante al macOS.*

*In proposito, qualche nota segue:*

## *Compilazione di un programma c sotto linux*

```
cc prova1c.c -o prova1c
```

*compila il file prova1c.c e produce il file eseguibile prova1c*

## *Script unix*

*- Iniziano con #!/bin/sh*

*-si possono chiamare nome.cgi (o anche nome.sh. Magari evitiamo di chiamarli ".bat", ok?)*

## *Privilegi di esecuzione*

*Puo` essere che file eseguibili e script non possano essere eseguiti per carenza di permesso.*

```
chmod o+x nomefile.ext
```

# Attività pratica

**CGI-0** (facoltativo, e fattibile anche dopo gli esercizi della pagina successiva)

sperimentare con l'installazione casalinga di apache: eseguire la definizione delle directory locali viste a lezione (terza slide) e sperimentarne l'uso;

aggiungere in httpd.conf le definizioni necessarie a far apparire un index di una delle directory definite similmente all'esempio in figura:

aggiungere tipi di file non considerati nella configurazione di default;

aggiungere o riusare icone corrispondentemente a tipi di file aggiuntivi; aggiungere o modificare la descrizione testuale corrispondente a certi tipi di file.

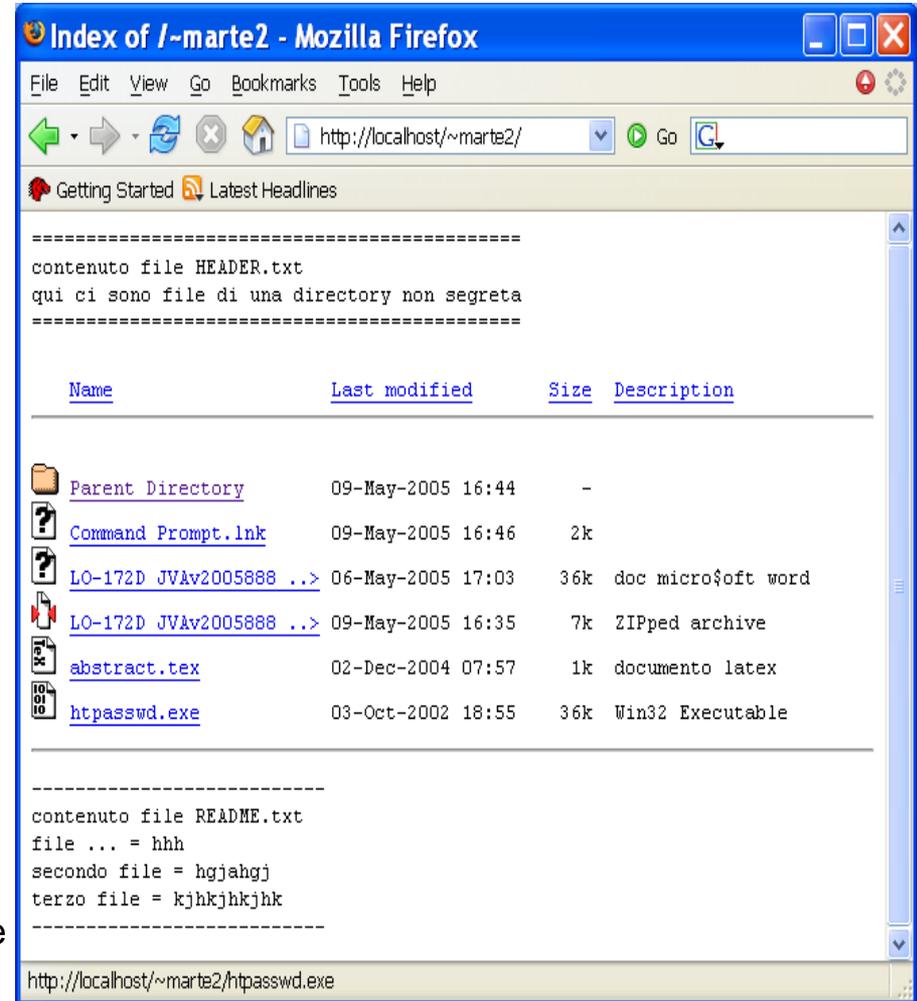
Descrivere in una pagina web contenuta nella directory le direttive ed opzioni usate nella definizione delle directory

**CGI-0bis** (facoltativo, e fattibile anche dopo gli esercizi della pagina succ.)

Proteggere una directory (definita come un alias in httpd.conf) mediante un file .htaccess; provvedere a che solo tre utenti di propria scelta, autenticati da password, possano accedere al listing della dir e ai suoi file.

Descrivere in una pagina web contenuta nella directory le direttive ed opzioni usate nella definizione della directory ed in quella del file di controllo accesso.

Provare a rendere impossibile l'accesso al listing della directory, lasciando però possibile l'accesso alla pagina web descrittiva di cui sopra



# Attività pratica

## CGI-1

sperimentare e modificare gli esercizi visti a lezione (corrispondenti a SuBatch.html, SuC.html).

Produrre almeno un nuovo file `eser.cgi.1.html` con un uso personale della cgi batch (o shell per unix) (Modificare solo "copie" dei file di partenza ...)

## CGI-2

Scrivere un documento `xhtml eser.cgi.2.html` in cui sia replicata la struttura della form a destra in alto; il campo cognome è di tipo password; azalea e anemone sono selezionati per default; sono possibili multiple selezioni per i fiori.

Sperimentare l'invio con metodo post e get con un programma cgi in c scritto appositamente (o `provac2.c`).

## CGI-3

ripetere il precedente ma con il layout mostrato a destra in basso

## CGI-4

replicare l'esercizio sugli indovinelli, permettendo di prendere gli indovinelli da un file testuale in modo randomico. Il file testuale contiene molti indovinelli, nel format seguente (dove R indica una risposta possibile all'indovinello - giusta; se ne danno 3 per ogni indovinello; quindi un indovinello è definito su 4 righe: una per il testo e tre per le risposte. La prima riga del file dice quanti indovinelli sono definiti nel file.

```
numero risposte nel file
<testo indovinello>
<R: prima risposta possibile>
<R: seconda risposta possibile>
<R: terza risposta possibile>
```

...

una form da gestire - Pale Moon  
http://localhost/MARCO-HTDOCS/ESPERIENZE-CGI/altro/formB...

una form da gestire x +

Nome: Susanna

Cognome: .....

Preferenze calcistiche:

- Roma
- Latina
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

- Anemone Epatica
- Agatea
- Amaranto
- Viola
- Azalea

Annulla le scelte Invio1

Nome: Susanna

Cognome: .....

Preferenze calcistiche:

- Roma
- Latina
- Inter
- Torino
- Udinese
- Juventus
- Lazio

Preferenze letterarie:

- Gialli
- Fantascienza
- Horror

Scegliere tra i seguenti fiori (anche più di uno):

Agatea

Annulla le scelte Invio1