

Web Programming

Introduzione al corso



Università degli Studi
di Brescia

*Riccardo Cassinis
Marzia Tassi
Fabio Tampalini*

10 maggio 2005

Il sito del corso

- Queste trasparenze, e tutto il materiale relativo al corso, sono reperibili all'indirizzo web

<http://www.ing.unibs.it/~cassinis/Dida/2004/FSE/index.htm>

che contiene:

- Calendario e orari del corso, elenco dei partecipanti, nomi degli account individuali
- Materiale didattico



Sito utilizzabile per gli esperimenti

- Ogni allievo è stato dotato di un account sulla macchina `rifraff.ing.unibs.it` (sistema operativo Linux)
- Rifraff è raggiungibile da tutta l'internet mediante i protocolli ssh e http, ad esempio utilizzando PUTTY, PSCP e Internet Explorer
- Il nome dell'account è riportato sulla pagina <http://www.ing.unibs.it/~cassinis/Dida/2004/FSE/index.htm>
- La password, per tutti gli account, è inizialmente "fse" (senza le virgolette) e deve essere **immediatamente** cambiata usando il comando "passwd"
- L'accesso http è possibile all'indirizzo `http://rifraff.ing.unibs.it/~nomeaccount/nomefile`

La struttura della directory

- Inizialmente, ogni directory contiene la cartella `public_html`, che a sua volta contiene i file
 - [index.html](#)
 - [index.php](#)
- e la cartella
 - `cgi-bin`
- che contiene il file (eseguibile)
 - [test.cgi](#)



Introduzione

- Questo modulo si propone di integrare nozioni in parte già acquisite, applicandole alla realizzazione di un organico “programma”, potenzialmente distribuito sulla rete Internet su scala mondiale.
- Esso prevede una parte di introduzione al lavoro, e una parte realizzativa.

Prerequisiti

- Il corso si rivolge a chi ha già un minimo di esperienza di uso del linguaggio HTML e dei sistemi operativi Windows e Linux
- La documentazione necessaria può essere integralmente reperita sull'Internet. I link relativi sono indicati sul sito web segnalato in precedenza
- Le informazioni sulla Bourne-Again Shell possono essere ottenute direttamente da rifraff digitando *man bash*
- Lo stesso vale per tutti i comandi Linux



Obiettivi

- Acquisire “manualità” nell’uso di alcuni strumenti per la produzione del software, con particolare riferimento a
 - Pagine HTML statiche
 - PHP
 - Uso elementare di MySQL
 - Vari editor di testo
 - Client FTP, SSH, HTTP, etc.
- Rendersi conto delle problematiche reali che si incontrano durante lo sviluppo di un progetto software
- Evidenziare i vantaggi e le difficoltà dello sviluppo di moduli software cooperanti

Web programming

10 maggio 2005

7

Programma di lavoro

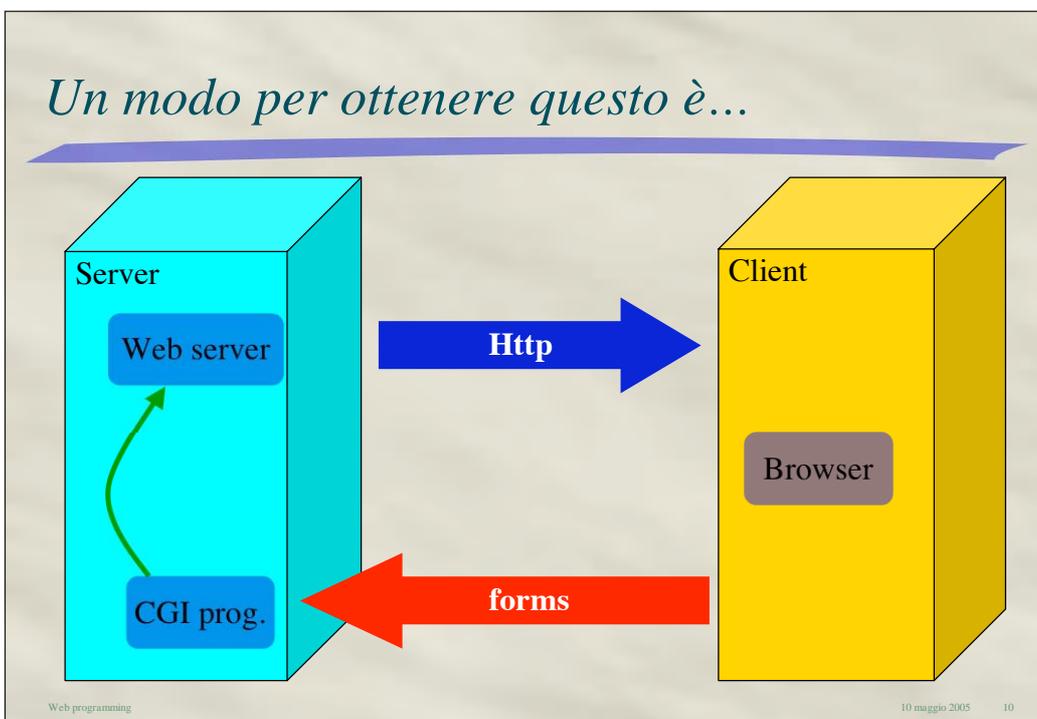
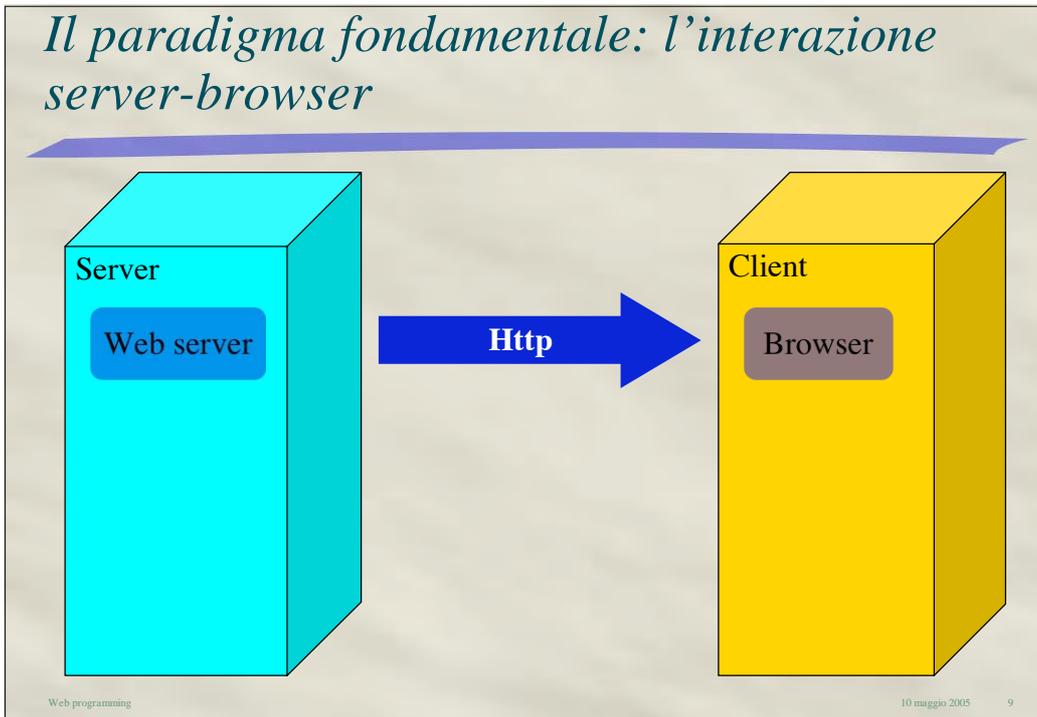
- Lezione: 2 ore
 - Organizzazione del lavoro
 - Presentazione e discussione degli strumenti
 - Introduzione al linguaggio PHP, con esercizi
- Esercitazione pratica 1: 4 ore
 - Esame delle caratteristiche della rete: numeri di IP, routing
 - Introduzione a MySQL, con esercizi
 - Presentazione del problema da risolvere e della distribuzione del lavoro
 - Preparazione e collaudo delle pagine HTML
 - Preparazione di alcuni script
- Esercitazione pratica 2: 4 ore
 - Preparazione dei rimanenti script
 - Integrazione e collaudo del sistema
- Valutazione e discussione dei risultati: 2 ore

Web programming

10 maggio 2005

8





Ma un altro modo potrebbe essere...

- Scrivere del codice HTML, in cui alcuni tag sono dei programmini
 - che vengono eseguiti quando il file viene servito
 - il cui (eventuale) output viene servito insieme al file HTML
 - che possono utilizzare file contenuti sul server...
 - ...magari organizzati con un sistema facile da usare.

Per finire

- Durante il corso, i docenti saranno a disposizione, tramite l'indirizzo di posta elettronica `nome.cognome@unibs.it`, per risolvere ogni perplessità relativa alle specifiche



Materiale utile per lo svolgimento del lavoro

- Le trasparenze che seguono contengono alcuni “pro memoria” utili per svolgere il lavoro

Le principali parole chiave relative ai form

```
<!-- This is the start tag for the form. -->
<form ACTION=MAILTO:riccardo.cassinis@unibs.it METHOD=POST>

<!-- This is a hidden field which sends a text so we know from which form we got this.
<input NAME="subject" TYPE="hidden" VALUE="Mail from ARL guestbook">
<!-- This is a standard text field which has the width of 56 characters. -->
<input TYPE="text" NAME="name" SIZE=56 MAXLENGTH=56>

<!-- This is a popup field where the user can choose an item. -->
<select NAME=from>
<option VALUE="mail message">I found the address in a mail message
<option VALUE="search engine">From a search engine
</select>

<!-- This is two radiobuttons where the user can choose an item, for example sex. -->
<input TYPE="radio" NAME="sex" VALUE="male">Male
<input TYPE="radio" NAME="sex" VALUE="female">Female

<!-- This is a textfield where the user can type in some comments about your page.. -->
<textarea NAME="comments" COLS=60 ROWS=8 WRAP="soft"></textarea>

<!-- At last, we have a 'sign' button and a 'reset' button which will clear the form. -->
<input TYPE="submit" VALUE="Sign">
<input TYPE="reset" VALUE="Reset">
<!-- Finally, don't forget to end the form using the end tag. -->
</form>
```



Script da inserire in tutte le pagine HTML

```
<script LANGUAGE="JavaScript">
<!-- Hide this script from incompatible Web browsers! -->
document.writeln("<i>Ultimo aggiornamento: ",
                  document.lastModified)
document.writeln("</font></i> ")
// Hide this script from incompatible Web browsers! -->
</script>
```

Uno script che potrebbe essere utile...(1)

```
#!/bin/bash
# La variabile IFS (Input Fields Separator) contiene tutti i caratteri che devono essere
# considerati separatori. Siccome nel nostro file i campi sono separati da <TAB>,
# definiamo il <TAB> come unico separatore possibile (il <CR> lo e' per
# definizione).

IFS=" "

echo IFS "$IFS"

NOMEFILE="provaruota"
echo *****
echo Apertura del File:$NOMEFILE
echo *****

# apre il file numero 3 usando un operatore di redirection
exec 3<>$NOMEFILE
```



Uno script che potrebbe essere utile...(2)

```
# legge la prima riga del file in tre variabili
read primo secondo terzo <&&3
# cicla con la lettura e la stampa fintanto che $primo contiene qualcosa
while [ $primo ]
do
    echo $terzo          # scrive su standard output
    echo $secondo
    echo $secondo >> fileuscita
    echo $primo >> fileuscita
    echo
    read primo secondo terzo <&&3 # legge la prossima riga
done

#chiude il file
3<&-
```

Uno script che potrebbe essere utile...(3)

```
# riapre il file numero 3 e fa le stesse cose di prima
exec 3<> $NOMEFILE
echo "File" $NOMEFILE "riaperto"

read primo secondo terzo <&&3
while [ $primo ]
do
    echo $terzo
    echo $secondo
    echo $primo
    echo $terzo >> fileuscita
    echo $secondo >> fileuscita
    echo $primo >> fileuscita
    echo
    read primo secondo terzo <&&3 # legge la prossima riga
done
3<&-

# termina regolarmente
exit 0
```



Non dimenticare...

- Di mettere due linee bianche dopo l'intestazione di ogni pagina html generata automaticamente
- Di scrivere correttamente l'intestazione:
"Content-type: text/html"
- Di salvare sempre il lavoro fatto a intervalli il più ravvicinati possibile

