

STAGES PER STUDENTI DI SCUOLA MEDIA SUPERIORE



9 / 27 GIUGNO 2003



SESSIONE : **INFORMATICA**

TEMA : **Conversione e ottimizzazione
delle pagine HTML dinamiche,
partendo da documenti XML**

STUDENTE : **Marco Mariani**

ISTITUTO : **Liceo Scientifico Statale "FARNESINA"**

TUTOR : **Igor Sfiligoi**

27/06/03

SCOPO DELLO STAGE

Lo scopo dello stage era lo sviluppo di pagine HTML dinamiche partendo da dati in formato XML.

STRUMENTAZIONE USATA

Durante tutto il corso ci siamo serviti di un computer fornito di Linux connesso ad internet con possibilita' di accedere sia a macchine della rete locale, che della rete al di fuori dei laboratori.

Abbiamo utilizzato Xemacs ed Emacs per scrivere i nostri programmi e Mozilla o Netscape per visualizzarli in formato HTML.

Abbiamo usufruito, inoltre, di un account personale dove salvare i nostri lavori.

Poiche' nessuno di noi aveva mai lavorato in Linux, soprattutto durante la prima settimana sono stati utilizzati libri finalizzati all'apprendimento e all'utilizzo dello stesso.

Il libro "Linux in a NutShell" di "Jessica Perry e O'Reilly Staff" (ISBN 1565921674) ci ha insegnato a muoverci in Linux con comandi quali "ls" o "cd" e ad usare alcuni programmi come emacs, xemacs o mozilla.

Dopo aver imparato a muoverci in Linux cominciava il nostro vero corso di programmazione in PHP. Nei primissimi giorni abbiamo seguito due interessanti lezioni on-line sul linguaggio PHP che ci hanno insegnato le funzioni base per scrivere qualche programma.

Lezione PHP di "Antonio Corsi"

http://web.tiscali.it/no-redirect-tiscali/antonio_corsi/AA20022003/php_lez_indice.htm

Guida di PHP di "A.Bergamasco"

<http://www.risorse.net/php/index.php>

Durante l'esecuzione degli esercizi richiesti in queste lezioni il libro "Guida a PHP (seconda edizione)" di "Tim Converse e Joyce Park" (ISBN 8838643202) e' stato molto utile aiutandoci ad imparare le varie formule e a ricordare i tanti comandi del PHP.

Oltre ai libri, durante queste lezioni ci siamo serviti di altra documentazione: i manuali e i tutorial on-line.

Manuale PHP

<http://www.php.net/manual/it/>

Tutorial FORMS di "Joe Burns"

<http://www.htmlgoodies.com/tutors/forms.html>

Questi, a mio parere, sono risultati piu' veloci per quanto riguarda la consultazione o la ricerca di una particolare funzione, e piu' esplicitivi grazie ad ottimi esempi che rendevano piu' chiaro il funzionamento del comando in questione.

Una volta acquisito il linguaggio PHP i nostri esercizi sono evoluti fino all'introduzione dei file in formato XML nell'ultima settimana del corso. Il libro "XML la guida completa" di "Heather Williamson" (ISBN 8838642125) e' servito a conoscere questo formato e comprenderne le dinamiche.

METODOLOGIA DEL LAVORO

Il nostro corso puo' essere diviso fondamentalmente in tre parti equivalenti alle tre settimane di permanenza:

- Nella prima settimana abbiamo imparato a conoscere il linguaggio PHP, la sua traduzione in HTML e le sue funzioni principali. Con l'aiuto dei manuali scritti e di quelli on-line abbiamo svolto alcuni esercizi applicativi sempre sotto il controllo del tutore che ci ha lasciato molta liberta' quando si trattava di apprendere qualche nuova funzione e che, invece, ci ha aiutato sempre quando avevamo dei problemi negli esercizi. Tutta la prima sessione di esercizi e' servita a familiarizzare con il PHP e le sue tante funzioni. Abbiamo seguito le lezioni on-line e abbiamo sviluppato esempi di ogni funzione per capirne i meccanismi ed assimilarli. Qualche problema l'abbiamo avuto sul For, sul ForEach (vedi esempio 1) e sugli Array ma tutti brillantemente superati grazie ai consigli del nostro tutore e ai manuali sia testuali che on-line. Questi esercizi sono stati molto importanti perche' hanno fondato la base sulla quale poi ci siamo mossi e dalla quale siamo partiti per scrivere tutti gli altri programmi.

- Nella seconda settimana, dopo aver assimilato bene le funzioni principali, le abbiamo messe in pratica con l'aggiunta di nuove tecniche di rappresentazione quali la costruzione di tabelle in HTML o la struttura dei FORM (vedi esempio 2) grazie ai quali abbiamo iniziato a realizzare pagine WEB collegate fra loro, il tutto partendo da file di dati (solitamente denominati come file.DAT). Quindi abbiamo cominciato a creare programmi specifici, capaci di leggere dati e di creare una pagina HTML visualizzando una tabella contenente i valori precedentemente letti. Finito questo primo esercizio lo abbiamo adattato per un file di dati piu' lungo cercando di separare le varie tabelle e dando cosi' la possibilita' all'utente di accedere ad una pagina HTML contenente solo i dati da lui richiesti, tutto tramite l'uso di FORMS. Poi abbiamo migliorato lo stesso file aggiungendo nuove funzionalita' quali ForEach, Chop, If-Else e Array per renderlo piu' dinamico, pulito e ottimale. In particolare abbiamo speso molto tempo cercando di imparare a creare Array multidimensionali che riassumono i file in formato dati semplici (vedi esempio 3).

- Nell'ultima settimana l'evento clou e' stato l'uso di file dati in formato XML (vedi esempio 4) e non piu' in file.DAT come eravamo abituati. Questo ha comportato la necessita' di documentarsi sul formato XML e su come leggerlo. Con l'utilizzo dell'XML molte cose sono cambiate, a partire dal ruolo centrale delle Funzioni che sino ad ora non avevamo preso in grande considerazione. Inizialmente si trattava solo di prendere il nostro file di dati in formato testo semplice e tradurlo in

XML, per cominciare a familiarizzare con questo nuovo formato. Una volta imparato a leggere e scrivere questo nuovo tipo di dati, i programmi seguenti non sono stato altro che l'adattamento dei precedenti dalla lettura di file.DAT a quella di file.XML. Infatti con impegno costante e qualche aiuto da parte del tutore siamo riusciti: in primo luogo a creare un lettore di file per formato XML, poi a far riscrivere lo stesso file.XML in un Array multidimensionale come quello degli esercizi precedenti, ed quindi usare questo Array per sviluppare una tabella HTML dinamica, dando all'utente la possibilita' di scegliere sia lo schema che la tabella/view da visualizzare (vedi esempio 5).

RELAZIONI, FORMULE ED EVENTUALI SOFTWARE UTILIZZATI

Una Pagina HTML

I Tag

Una nozione fondamentale che abbiamo imparato durante questo stage e' che le pagine HTML sono ordinate da TAG. Per esempio un file HTML avra' sempre i tag

```
<html>  
.....  
.....  
</html>
```

dove i puntini indicano tutto cio' che verra' riportato in output.

Questo e' molto importante per iniziare a dare una struttura al programma.

In una pagina vi possono essere molti TAG diversi a seconda della funzione, del colore, della posizione o del carattere della stringa che racchiudono.

Ad esempio, questi consentono di scrivere in carattere la stringa in carattere formato titolo

```
<h2> ..... </h2>
```

Questi altri consentono l'apertura di una tabella

```
<table> ..... </table>
```

Come e' facile intuire tutte le tag sono comprese dai segni <..> ed in particolare quelle di chiusura contengono il segno /.

Composizione

Una pagina in genere consta di due TAG fondamentali: un <HEAD> e un <BODY>.

La HEAD comprende il titolo del programma e tutto cio' che l'autore considera importante ma che non verra' mai visualizzato in output.

Il BODY, invece, comprende tutta la parte di testo finalizzata al funzionamento del programma.

Un Programma PHP

All'interno di una pagina HTML dinamica sono presenti le TAG

```
<?php
```

```
.....
```

```
?>
```

che stanno ad indicare l'inizio (e la fine) di un programma PHP.

La prima cosa che abbiamo imparato sulla programmazione in PHP e' stata la denominazione di un comando: in PHP ogni comando termina con il punto e virgola.

Per quanto riguarda le funzioni di scrittura in output:

- Se si vuol avere una stringa come risultato in uscita si utilizza

```
ECHO "string"; oppure PRINT ("string");
```

- Se si vuole scrivere una variabile (indicata in PHP da \$nome_della_variabile) si utilizza

```
ECHO $nome_della_variabile; oppure PRINT ($nome_della_variabile);
```

- Se, invece, si vuole visualizzare un Array si utilizza

```
PRINT_R ($nome_dell'array);
```

L'assegnazione di un valore ad una variabile avviene tramite operatori: o semplicemente per assegnazione (\$var = "valore") oppure tramite operazioni piu' complesse quali For o ForEach.

I commenti (stringhe di testo che non vengono visualizzate in output) in PHP si ottengono con //..... oppure con /*.....*/.

Per gli input dati dall'utente, ovvero per rendere interattiva la pagina in costruzione, abbiamo imparato ad utilizzare i FORM e i pulsanti di INPUT che ci hanno permesso di cominciare a realizzare pagine collegate tra loro.

Il linguaggio PHP e' regolato dagli operatori di controllo; per esempio If-Else.

Ma gli operatori piu' complicati, ma anche i piu' utili, sono i Loop.

I loop esistenti sono vari: i piu' frequenti sono For, ForEach, While.

La loro funzione e' quella di eseguire operazioni ripetitive all'interno di un testo.

Oltre ai loop vi sono le Funzioni che permettono di memorizzare una serie di istruzioni e ripeterle senza bisogno di riscriverle.

Il PHP ha una grande dispensa di funzioni e operatori.

Nel corso del nostro stage, per ultimare gli esercizi, e' stato necessario prendere in considerazione funzioni quali File o Fopen, mentre per rendere possibile la realizzazione di tabelle abbiamo usato le funzioni da Array ed Explode.

Se si vuole utilizzare il formato XML le funzioni che leggono e rielaborano questo nuovo linguaggio sono i Parser (\$xml_parser).

RISULTATI DELLO STAGE

Buona parte dello stage e' stata dedicata all'acquisizione delle nozioni base di PHP delle quale eravamo privi.

Dopo averle acquisite abbiamo cominciato a realizzare programmi semplici in grado di visualizzare tabelle in pagine HTML.

Da questi siamo passati a pagine dinamiche ed interattive con la possibilita' di scegliere la tabella da visualizzare.

Anche con i file dati in formato XML gli esercizi erano piu' o meno gli stessi: e tutti si basavano sul creare array multidimensionali (ad esempio array contenenti sottoarray "schema" che a loro volta comprendono sottoarray "table" che sono composti da "field" che comprendono le stringhe "name", "type" e "nullable" che poi vengono riportate in tabella) dai quali e' possibile estrarre, senza grande difficolta', e con velocita', le informazioni richieste.

Il risultato finale dello stage e' stata la creazione di pagine HTML dinamiche partendo da file dati in formato XML (vedi esempio 5).

1) ESEMPIO SUL FOREACH

```
<?php
$a1 = array("abc"=>"def", "ghk"=>"iuy");
$a2 = array("zxc"=>"vbn", "123"=>"456");
$c1 = array("a1"=>$a1, "a2"=>$a2);
$a12 = array("abc2"=>"def2", "ghk2"=>"iuy2");
$a22 = array("zxc2"=>"vbn2", "1232"=>"4562");
$c2 = array("a12"=>$a12, "a22"=>$a22);
$d1 = array("c1"=>$c1, "c2"=>$c2);

foreach ($d1 as $e=>$f)
{ echo "<schema name=\"\" . $e . \">\" . \"\n\";
  foreach ($f as $g=>$i)
  { echo "\t<table name=\"\" . $g . \">\" . \"\n\";
    foreach ($i as $h=>$k)
    { echo "\t\t<field \" . $h . \"=\" . $k . \">\n\";}
    echo "\t</table>\n\";}
  echo "</schema>\n\";}
?>
```

2) ESEMPIO SUI FORM

```
<html>
<head>
<title> NuovoEsempio </title>
</head>
<body>
<p>
Inserisci tre numeri a tua scelta e ne avrai delle relazioni
</p>
<FORM action="NuovoEsempioCorrelato.php" method="get">
<p>
Numero A
<input type="NUMBER" name="a">
<br>
</p>
<p>
Numero B
<input type="NUMBER" name="b">
<br>
</p>
<p>
Numero C
<input type="NUMBER" name="c">
<br>
</p>
<input type="submit" name="submit" value="INVIA">
<input type="reset" name="reset" value="RIPRISTINA">
</FORM>
</body>
</html>
```

3) ESEMPIO SUGLI ARRAY MULTIDIMENSIONALI

```
<?php
$dat = file ('prova2finale.dat');
$primo = 0;
$tabelle = array();
foreach ($dat as $a=>$c)
    { if ($a == "---\n")
        { $titolo = explode(".", chop($dat[$primo]));
          $schema = $titolo[0];
          $tabella = $titolo[1];
          $field = array();
          for ($b=$primo+1;$b<$a;$b++)
              { $field[] = explode (" ", $dat[$b]);
                $tabelle[$schema][$tabella] = $field;
                $primo = $a+1; }}
    }
?>
```

4) ESEMPIO DI FILE.XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE dbfdesc PUBLIC "-//Sfiligoi Igor//DTD dbfields//EN" "dbfields.dtd">
<dbfields>
<schema name="cndrng">
    <table name="dc_bads">
        <field name="start_of_validity" type="integer" nullable="No" />
        <field name="end_of_validity" type="integer" nullable="Yes" />
        <field name="layer" type="integer" nullable="No" />
        <field name="wire" type="integer" nullable="No" />
        <field name="status" type="integer" nullable="No" />
    </table>
    <table name="dchtrg_calib">
        <field name="start_of_validity" type="integer" nullable="No" />
        <field name="end_of_validity" type="integer" nullable="Yes" />
        <field name="slope_t0d" type="double" nullable="No" />
        <field name="slope_t1d" type="double" nullable="No" />
        <field name="slope_tcrd" type="double" nullable="No" />
        <field name="slope_t2d" type="double" nullable="No" />
        <field name="offset_t0d" type="integer" nullable="No" />
        <field name="offset_t1d" type="integer" nullable="No" />
        <field name="offset_tcrd" type="integer" nullable="No" />
        <field name="offset_t2d" type="integer" nullable="No" />
    </table>
    <table name="emc_bads">
        <field name="start_of_validity" type="integer" nullable="No" />
        <field name="end_of_validity" type="integer" nullable="Yes" />
        <field name="subdet_id" type="integer" nullable="No" />
        <field name="module" type="integer" nullable="No" />
        <field name="plane" type="integer" nullable="No" />
        <field name="column" type="integer" nullable="No" />
    </table>
</schema>
</dbfields>
```



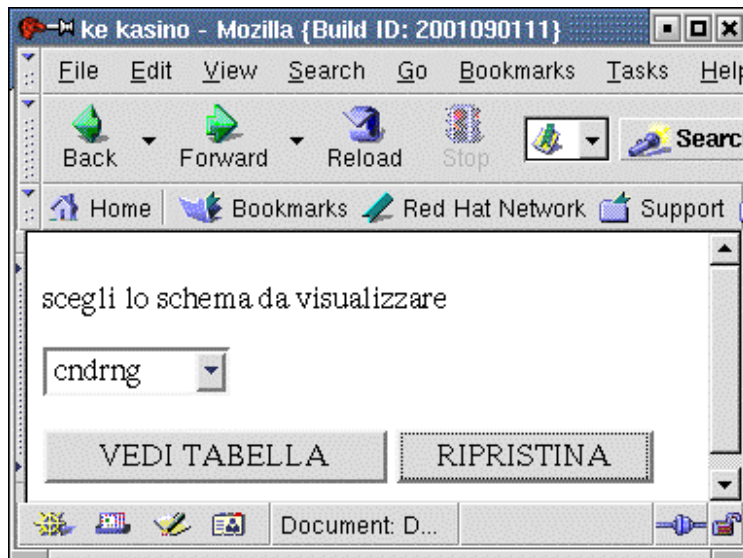
```

        <field name="side" type="integer" nullable="No" />
        <field name="board_id" type="integer" nullable="No" />
        <field name="status" type="integer" nullable="No" />
    </table>
</schema>
<schema name="cndrun">
    <table name="dc_eff">
        <field name="run_nr" type="integer" nullable="No" />
        <field name="layer" type="integer" nullable="No" />
        <field name="hw_eff" type="double" nullable="No" />
        <field name="hw_eff_err" type="double" nullable="No" />
        <field name="sw_eff" type="double" nullable="No" />
        <field name="sw_eff_err" type="double" nullable="No" />
    </table>
</schema>
<schema name="crossref">
    <table name="torta_run">
        <field name="run_nr" type="integer" nullable="No" />
        <field name="start_of_validity" type="integer" nullable="No" />
        <field name="insert_time" type="timestamp" nullable="No" />
    </table>
</schema>
<schema name="daq">
    <table name="test1">
        <field name="filename" type="varchar(64)" nullable="No" />
        <field name="type" type="char(1)" nullable="No" />
        <field name="data" type="integer" nullable="No" />
        <field name="id" type="char(13)" nullable="No" />
    </table>
</schema>
</dbfields>

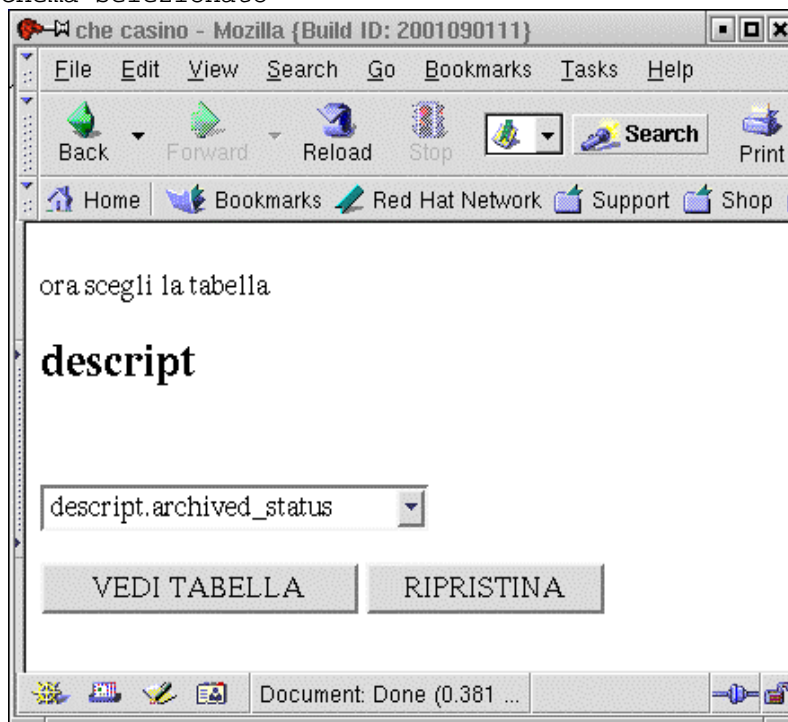
```

5) ESEMPIO DI PAGINA HTML DINAMICA INTERATTIVA

In questa pagina e' possibile scegliere lo schema da visualizzare



Nella seconda pagina la scelta ricade sulle tabelle presenti all'interno dello schema selezionato



Nella terza pagina si visualizzerà la tabella richiesta

