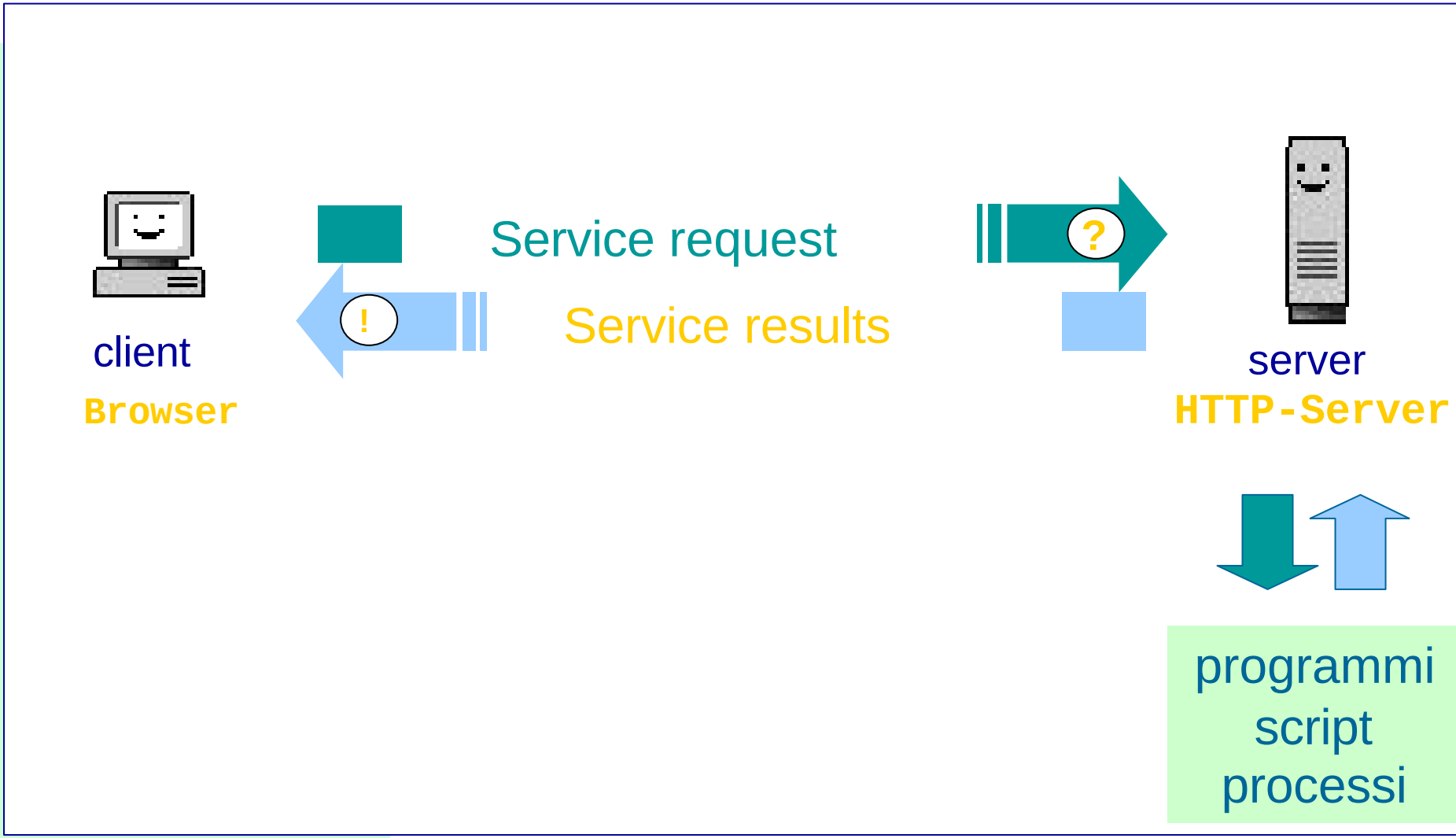
PHP

Paola Salomoni
salomoni@csr.unibo.it
mercoledì 20 dicembre 2000

Programmazione Server-Side



CGI

- CGI (Common Gateway Interfaces) definiscono come:
  - il client effettua il passaggio dei dati di input e la chiamata al programma sul server,
  - il server restituisce i dati di output del programma al client.

- La CGI deve funzionare all'interno del protocollo HTTP che è usato come supporto in entrambe le direzioni.

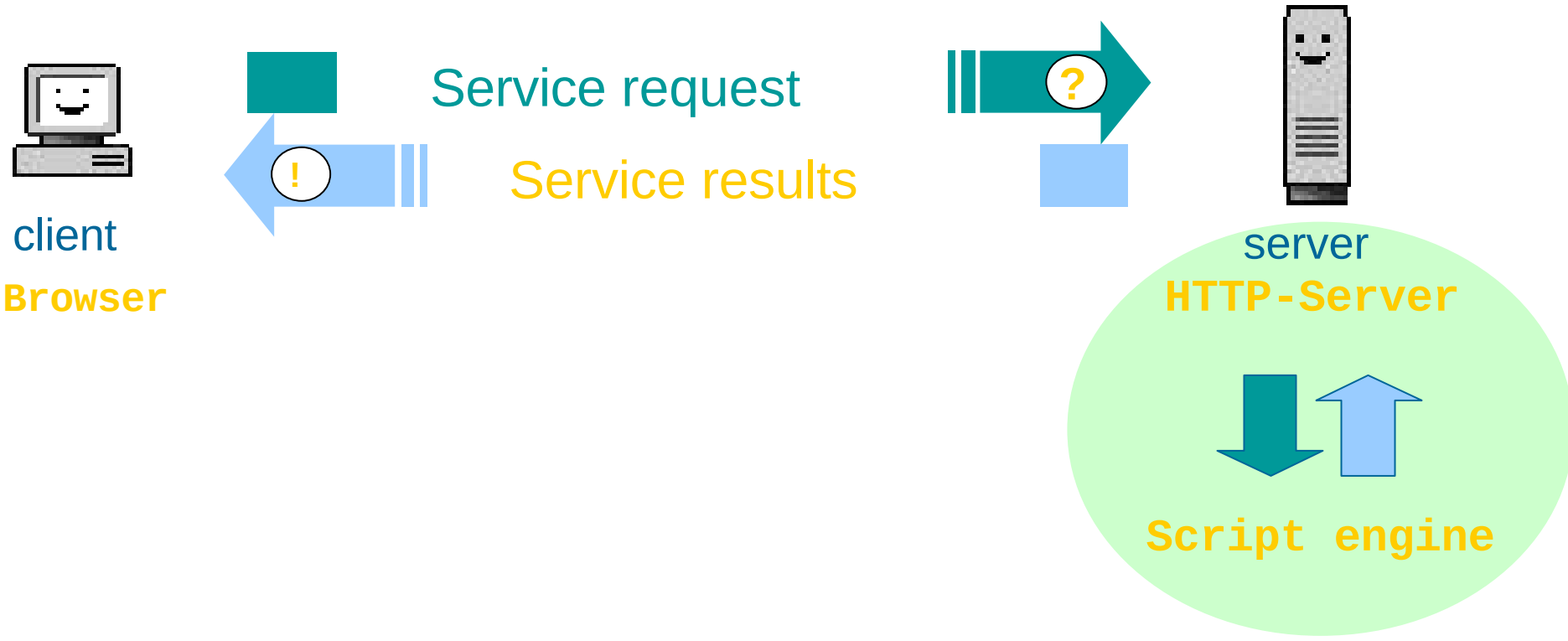
CGI: fasi

- Fasi:
  - ① (C): raccolta dati dal lato client mediante form
  - ② (C): invocazione del programma server e spedizione dei dati: IL PROGRAMMA E' in PERL, C,…..
  - ③ (S): lancio del programma
  - ④ (S): per prima cosa il programma decodifica i dati
  - ⑤ (S): produzione dell'output . IL PROGRAMMA produce stampe sullo standard output contenti la pagina HMTL da visualizzare.
  - ⑥ (C): visualizzazione dell'output (HTML)

PHP

- PHP (Personal Home Page Tool) è un sistema di integrazione di script all'interno del codice HTML.
- Richiedendo un file HTML che include codice PHP, il client chiede al server di interpretare il codice ed eseguirlo (HTML-embedded).
- Il server sostituisce al codice PHP il prodotto dell'esecuzione e il file HTML risultante viene restituito al client.

PHP


Service request
Service results
?
!
client
Browser
server
HTTP-Server
Script engine

PHP

- Lo script engine PHP si integra con il server HTTP (Apache) fornendo una alternativa alle soluzioni Microsoft basate su IIS+ASP+SQLS.

- Dimensioni dello sviluppo di PHP:
  - nel 1996 i siti basati su PHP erano circa 15.000
  - attualmente sono più di 150.000.

PHP

- PHP è:
  - OPEN SOURCE
  - MULTIPIATTAFORMA
- PHP fornisce supporto alla programmazione server side attraverso:
  - funzioni e costrutti di programmazione di base,
  - accesso all'*ambiente* del server,
  - primitive grafiche integrate,
  - supporto per i database più diffusi.

CGI vs PHP

- CGI vs PHP:
  - PHP è un linguaggio embedded che viene incluso nel codice HTML delle pagine e non necessita di file esterni per essere eseguito;
  - uno script PHP, di fatto, non ha bisogno di installazione, essendo il codice inserito direttamente nelle pagine, una volta che la pagina è caricata lo script è pronto per l'uso.
  - PHP consente di operare ad alto livello con:
    - i dati provenienti dalle form;
    - i DB sul lato server.

PHP e database

- In sostanza PHP consente di scrivere qualunque programma tipo CGI integrandolo nel codice HTML.
- La funzionalità di maggior successo è sicuramente l'integrazione con i principali motori/interfaccie SQL, tra cui: ODBC, Informix, Oracle, IBM DB2, PostgreSQL, MySQL, InterBase, dBase, ...

codice PHP

- Ogni porzione di codice PHP è inserito all'interno dei tag <?php e ?> (nel body dell' HTML).

```
<html>
  <body>
     <!-- HTML -->
     <?php . . . codice Php . . . ?>
     <!-- HTML -->
  </body>
</html>
```


codice PHP

- Alternativamente si può mettere il codice PHP tra i tag <script> e </script>.

```
<html>
   <body>
      <!-- HTML --->
      <script language="php">
      -- Codice PHP --
      </script>
      <!-- HTML --->
   </body>
</html>
```


sintassi base PHP

- Le istruzioni PHP sono concluse (e quindi separate da) punto e virgola (;).

```
<?php
   echo ”ciao mondo!";
?>
```

- PHP supporta sia i commenti tipici del C che quelli della shell Unix

```
<?php
   /* questo e’ un commento stile C*/
   # questo e’ un commento stile shell
 ?>
```


variabili PHP

- I tipi di variabili PHP fondamentali sono:
  - integer: numeri interi,
  - double: numeri in virgola mobile,
  - string: stringhe,
  - array: array scalari o associativi,
  - oggetti.
- Le variabili PHP non necessitano di dichiarazione e il nome della variabile comincia sempre con il simbolo $.

scope

- Lo scope di una variabile è limitato al contesto in cui sono definite (script, funzione, …).
- Per definire una variabile globale occorre dichiararla come tale facendo precedere la dichiarazione (o il primo uso) da global.

```
global $a;
```


array

- PHP supporta senza grosse differenze sia gli array scalari che gli array associativi.
- Le seguenti scritture hanno lo stesso effetto:

```
$a[] = “ciao”;
$a[] = “mondo”;
```

```
$a[0] = “ciao”;
$a[1] = “mondo”;
```

```
$a = array(
     0 => “ciao”,
     1 => “mondo”;
 );
```


array

- Gli array possono anche essere multi-dimensionali:

```
$a[0][0] = “ciao”;
$a[0][1] = “mondo”;
```

- Gli array associativi sono trattati in modo analogo agli array scalari:

```
$a[“nome”]= “Paola”;
$a[0][“nome”] = “Paola”;
```


classi

- Una classe è una collezione di variabili e funzioni che ne fanno uso.
- Una classe viene definita utilizzando la seguente sintassi:
  - dichiarazione della classe
  - dichiarazione delle variabili
  - dichiarazione delle funzioni (metodi)

oggetti

- In PHP possono essere definite classi di oggetti con:
  - la dichiarazione di classe (class).
  - la primitiva new che consente di creare una istanza dell'oggetto.

```
<?php
   class ciao {
      function saluto() {echo ”ciao mondo.";}
   }
   $a = new ciao;
   $a->saluto();
?>
```


variabili predefinite

- Con PHP si può accedere a tutte le variabili d'ambiente (come nelle CGI) e alle variabili del server.

```
<HTML>
<BODY>
<? echo("Ultima modifica:".
   date("d/m/Y",filemtime($PATH_TRANSLATED))); ?>
</BODY>
</HTML>
```


variabili provenienti dalle form

- Le variabili acquisite tramite una form HTML possono essere recuperate dallo script PHP in due array associativi che contengono le coppie variabile,valore già decodificate:
  - HTTP_GET_VARS: per il metodo GET;
  - HTTP_POST_VARS: per il metodo POST.
- Perché ciò sia possibile si deve:
  - configurare PHP con track_vars=On, oppure
  - inserire nello script la direttiva "php_track_vars".

espressioni

- Le espressioni sono quasi tutte, sia sintatticamente che semanticamente, mutuate dal C.
- Per esempio sono espressioni valide:

```
condizione ? valorevero : valorefalso;
variabile++; variabile--;
variabile += 10;
variabile= variabile1 = espressione;
```


operatori

- Gli operatori PHP sono classificabili in:
  - Arithmetic Operators
  - Assignment Operators
  - Bitwise Operators
  - Comparison Operators
  - Error Control Operators
  - Execution Operators
  - Incrementing/Decrementing Operators
  - Logical Operators
  - String Operators

error control

- PHP consente di intercettare gli errori prodotti da una espressione, facendo precedere l'espressione dal simbolo @.
- Il messaggio d'errore viene conservato nella variabile globale $php_errormsg.

```
<?php
 /* errore esterno da SQL */
 $res = @mysql_query ("select name,code from
    namelist") or die ("Query failed: error was
  '$php_errormsg'");
?>
```


Strutture di controllo

- Le strutture di controllo (C like) sono:
  - if
  - else
  - elseif
  - while
  - do..while
  - for
  - foreach
  - break
  - continue
  - switch

include e require

- Atre strutture di controllo sono: require(), e include(), che rimpiazzano se stesse con un file indicato nell'istruzione.
- Non sono funzioni ma costrutti e sono simili alle direttive al preprocessore C ottenute con #include.
- Ci sono alcune differenze:
  - include, include il file solo se viene eseguita,
  - require, include il file sempre.

funzioni

- Anche le funzioni sono molto simili al C.

```
function funzio ($arg_1,$arg_2, ...,$arg_n) {
   /*codice…. */
   return $valoreritorno;
}
```

- Gli argomenti per default sono passati per valore, per passarli per indirizzo occorre farli precedere dal simbolo &.

default argument

- Possono essere usati anche argomenti con default value in stile C++.

```
function preferito ($colore = ”blu") {
  return ”il mio colore preferito e’ $type.\n";
}
echo preferito ();
echo preferito (”giallo");
```

- L’output sarà:

```
il mio colore preferito e’ blu.
il mio colore preferito e’ giallo.
```


librerie di funzioni

- PHP fornisce librerie di funzioni per trattare con tipi dati avanzati:
  - funzioni matematiche
  - funzioni sulle RE compatibili con Perl
  - funzioni di encryption
  - funzioni su tipi dato come array, calendar, URL
  - funzioni di gestione degli errori
  - funzioni per supporto XML
  - …..

librerie di funzioni

- PHP fornisce librerie di funzioni che consentono di interagire con altri applicativi lato server:
  - il server HTTP (e in particolare Apache)
  - il server DBMS (di molti tipi, tra cui quelli elencati in precedenza)
  - gli altri servizi Internet (POP, IMAP, SMTP, FTP,..)

PostgreSQL

- Scegliamo come DBMS PostgreSQL
- Postgres, originariamente sviluppato dalla UC Berkeley, è un prodotto di dominio pubblico, disponibile gratuitamente ed Open Source
- Fornisce supporto al linguaggio SQL92/SQL3, all'integrità delle transazioni, e all'estensibilità del tipo.

PostgrSQL

- PostgreSQL fornisce una interfaccia via socket alle interrogazioni.
- Significa che per comunicare con il demone SQL occorre:
  - connettersi ad una opportuna porta e aprire la connessione,
  - spedire interrogazioni e ricevere risultati sulla connessione,
  - chiudere la connessione.

PHP e PostgreSQL

- Una connessione tipo inizia con:

```
int pg_connect (string host,
   string port, string options,
   string tty, string dbname)
```

- e termina con:

```
bool pg_close (int connection)
```


Altre istruzioni

- Tra le tante istruzioni le più utili sono:
  - pg_Exec: esegue una query
    ```
    int pg_exec (int connection, string query)
    ```
  - pg_Fetch_Row: ottiene un vettore enumerato da una riga della tabella.
    ```
    array pg_fetch_row (int result, int row)
    ```
  - pg_NumRows: restituisce il numero di righe(ad esempio ottunute da una select)
    ```
    int pg_numrows (int result_id)
    ```


Istruzioni SQL del PostgreSQL

- Con l'istruzione pg_Exec, PHP consente di eseguire qualunque query sul DB PostgreSQL. Tra queste:
  - data selection and relations (select, join,..)
  - data manipulation (update, insert, remove,…)
  - gestione delle viste
  - gestione degli indici
  - gestione delle transizioni

transizioni

- Un blocco di transazione inizia con un begin e se la transazione è stata valida termina con commit ed end.
- Se la transazione fallisce, essa deve venire chiusa con rollback e abort.

gestione del DB via Web

- Solitamente la gestione del DB via Web è fatta per fornire interfaccia Internet a:
  - inserimenti, rimozioni, e altre operazioni di aggiornamento del DB
  - ricerche sul DB
- E’ raro che via DB si cambi la struttura di una tabella. E’ invece frequente che se ne cambi il contenuto.

gestione del DB via Web

- Sono fasi eseguite quasi sempre dall'amministratore del sistema direttamente sul DBManager:
  - la creazione del DB
  - l’attribuzione della proprietà e dei permessi
  - la definizione delle tabelle
- Per queste fasi non useremo una interfaccia PHP.

esempio

- Proveremo un esempio molto semplice (mono-tabella) per fare una panoramica dei principali comandi
- DB e tabella vengono creati da :

```
createdb prova
CREATE TABLE amici (id serial, nome
    char(20), cognome char(30));
```


la connessione

- La connessione al DB da PHP viene realizzata dal seguente script:

```
<html>
   <body>
    <?php
     $db = pg_connect(“”, “”, “”, “”, “prova”);
     /* codice */
     pg_close();
    ?>
  </body>
```

- L’host (solitamente locale) e la porta (solitamente 5432) sono omessi.

inserimento

- L’inserimento avviene in due fasi (che corrispondono a due file HTML):
  - l’inserimento dei dati da parte dell’utente all'interno di una form.
  - Il submit della form richiama un altro file HTML che effettua l’inserimento vero e proprio dei dati nel DB

inserimento: la form

```
<html>
 <body>
  <form action="add.php" method="post">
   Nome: <input type="text" name=”nome"
   size=”20" length=”20"><BR>
   Cognome: <input type="text" name=”cognome"
   size=”30" length=”30"><BR>
   <input type="submit" value=”inserisci">
  </form>
 </body>
</html>
```


inserimento: il codice

```
<html>
 <body>
  <?php
   $db = pg_connect("","","","",”prova");
   $query = "INSERT INTO amici (id, nome,
    cognome) values (nextval(’amici_id_seq'),
   '$nome', '$cognome')";
   $result = pg_exec($db, $query);
   if (!$result) {printf ("ERROR"); exit;}
   printf (”inserito- %s %s", $nome, $cognome);
   pg_close();
  ?>
 </body>
</html>
```


select all

- La selezione e visualizzazione di tutti i dati di una tabella è realizzata mediante un file HTML.
- Nel codice PHP contenuto nel file devono esserci:
  - una select *
  - un meccanismo che stampi riga per riga, le righe estratte dalla tabella.

select all

```
<html>
 <body>
  <?php
   $db = pg_connect("","","","","prova");
   $query = "SELECT * FROM amici";
   $result = pg_exec($db, $query);
   if (!$result) {printf ("ERROR"); exit;}
   $numrows = pg_numrows($result);
   $row=0;
   printf ("<table border=1>\n");
   printf ("<tr><td>ID</td><td>Nome
   </td><td>Cognome</td></tr>\n");
. . .
```


select all (2)

```
. . .
 do{
  $myrow = pg_fetch_row ($result,$row);
  printf("<tr><td>%s</td><td>%s</td><td>%s</td>
   </tr>\n",
  $myrow[0], $myrow[1], $myrow[2]);
  $row++;
 }
 while ($row < $numrows);
 printf ("</table><br>\n");
 pg_close();
?>
```


cancellazione

- La cancellazione è fatta in due fasi:
  - vengono visualizzate tutte le righe (simile a select all) rendendo sensibili gli indici.
  - ciascun indice chiama (passando l'ID) un file HTML che fa la vera e propria cancellazione sulla tabella del DB.

delete: la selezione

```
. . .
do{
  $myrow = pg_fetch_row ($result,$row);
  printf("<tr>
   <td> <A HREF='cancella.php?id=%s'> %s </A> </td>
   <td> %s </td><td> %s </td></tr> \n",
   $myrow[0], $myrow[0], $myrow[1], $myrow[2]);
  $row++;
 }
 while ($row < $numrows);
```

➢ . . .

delete: la modifica del DB

```
<html>
 <body>
  <?php
   $db = pg_connect("","","","","prova");
   $query = "DELETE FROM amici where id='$id'";
   $result = pg_exec($db, $query);
   if (!$result) {printf ("ERROR"); exit;}
   printf (”cancellazione OK\n");
   pg_close();
  ?>
 </body>
</html>
```


ricerca

- La ricerca è fatta in due fasi:
  - viene richiesto all'utente l'inserimento della chiave di ricerca
  - viene effettuata la select con quella chiave dei ricerca e vengono visualizzati i risultati.

ricerca: form

```
<html>
 <body>
  <form action="cerca.php" method="post">
   Inserisci il nome da cercare
   <input type="text" name="nome"
   <input type="submit" value="cerca">
  </form>
 </body>
</html>
```


Ricerca: select

```
<html>
 <body>
  <?php
   $db = pg_connect("","","","","prova");
   $query = "SELECT * FROM amici WHERE nome=$nome";
   $result = pg_exec($db, $query);
   if (!$result) {printf ("ERROR"); exit;}
   $numrows = pg_numrows($result);
   $row=0;
   printf ("<table border=1>\n");
   printf ("<tr><td>ID</td><td>Nome
   </td><td>Cognome</td></tr>\n");
. . .
```


update

- L’update è fatto in tre fasi:
  - vengono visualizzate tutte le righe (come nella cancellazione) rendendo sensibili gli indici.
  - ciascun indice chiama (passando l’ID) un file che riempie una form col vecchio contenuto della riga. L’utente modifica i dati della riga e con il submit chiama.
  - un terzo file HTML che fa il vero e proprio update sulla tabella del DB.

coockie

- Il cookie (letteralmente “biscottino”) è una stringa contente una piccola quantità di dati specifici che vengono memorizzati sulla macchina client per lasciare traccia del sito.
- Il browser può cioè, senza violare i criteri di sicurezza della macchina client, memorizzare una stringa.

coockie

- I dati sono lasciati sulla macchina client per poter essere riutilizzati in un secondo passaggio dalla stessa pagina.
- Questo tipo di informazioni servono per:
  - lasciare dati sull'utente ma NON essenziali perché il coockie può essere cancellato in qualsiasi momento.
  - Fare statistiche sui propri accessi.

coockie

- I coockie sono parte di una estensione del protocollo HTTP per cui per registrare o leggere un coockie si può alternativamente:
  - fare eseguire al browser uno script Javascritpt che contiene l'azione di scrittura/lettura del coockie.
  - Utilizzare una CGI dal lato server che con appostiti pacchetti HTTP comanda al browser di scrivere/leggere il file di cookie.

coockie

- Il cookie è sostanzialmente un file di testo a cui il browser accede:
  - in lettura
  - in scrittura in append (per preservare i cookie lasciati da altre pagine)

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file!  Do not edit.

secure.webconnect.net    FALSE      /cgi-bin    FALSE  1234117898       4415
www.teknosurf2.com           FALSE  /cgi-bin    FALSE  1102750443       teknoacc    3098945170
.freefind.com                TRUE               /servlet    FALSE  1269673665       time        954316807052
secure.webconnect.net    FALSE      /cgi-bin    FALSE  1234117888       8983
www.familyeducation.com  FALSE      /article    FALSE  1577836800       HC          y
```


coockie

- La struttura di un cookie è standard e definita dall'HTTP con due tipi particolari di header:
  - Set-Cookie HTTP Response Header: il client riceve dal server il comando di assegnare il cookie.
  - Cookie HTTP Request Header: il server richiede all client il valore attuale del cookie.

Set-Cookie

- Sintassi del Set-Cookie:
  - **`NAME=`** VALUE: valore del cookie in cui si memorizzano le informazioni “personalizzate” che il sito vuole lasciare sul client
  - **`expires=`** DATE: data in cui il cookie può essere cancellato
  - **`path=`** PATH; URL della pagina che ha lasciato il cookie, comprensivo della componente dell’URL che rappresenta il path locale

Set-Cookie

- Sintassi del Set-Cookie:
  - **`domain=`** DOMAIN_NAME: se il cookie vuole "rappresentare" più pagine si può:
    - settare un path comune (es: solo l'host)
    - usare domain (meglio se ci sono più macchine server con lo stesso tipo di cookie)
  - **`secure`**: specifica in che se il cookie attraversa la rete in modo sicuro (TRUE) o no (FALSE).

Request

- La richiesta via HTTP produce una riposta che contiene l'elenco dei cookie il cui URL corrisponde a quello della richiesta:

```
Cookie: NAME1=OPAQUE_STRING1;
  NAME2=OPAQUE_STRING2 ...
```


cookie

- Ci sono alcuni numeri che limitano la disponibilità del client ad accettare cookie (e di conseguenza le dimensioni del file di cookie):
  - 300 cookie in totale,
  - 4 kilobyte per cookie,
  - 20 cookie per server o dominio.
- La rimozione avviene in conformità alla expiration date specificata oppure con un criterio del tipo least recently used.

PHP e i cookies

- Anche in questo caso ci sono due possibili operazioni:
  - inserimento di un nuovo cookie,
    - mediante la funzione setcookie(Nome, Valore, Espirazione, Percorso, Dominio, Secure);
  - lettura del cookie.
    - HTTP_COOKIE_VARS: un array associativo contenente le variabili passate allo script tramite i cookies HTTP;

setcookie

- setcookie(Nome, Valore, Espirazione, Percorso, Dominio, Secure) dove:
  - Nome è il nome del cookie (arbitrario);
  - Valore è il valore (arbitrario);
  - Espirazione è la data di espirazione del cookie;
  - Percorso è la directory, a partire dal dominio (vedi sotto) per la quale il cookie è valido;
  - Dominio è il dominio per il quale il dominio è valido;
  - Secure è un valore che imposta se il cookie debba essere inviato tramite una connessione HTTPS.

Riferimenti

- *PHP Home Page*, http://www.php.net/
- *PostgreSQL Home Page*, http://www.PostgreSQL.org
- S.S. Bakken e altri, *Manuale PHP*, http://www.php.net/manual/it/manual.php