PHP

A. Lorenzi, E. Cavalli

INFORMATICA PER ISTITUTI TECNICI TECNOLOGICI

Atlas

Programmazione lato server

- PHP è un linguaggio che estende le funzionalità del Web server consentendo l'interpretazione di file con estensione *.php* contenenti il codice dell'applicazione.

Programmazione lato server

- Quando il client richiede una pagina con estensione *.php*, il server Web non spedisce al browser direttamente il file.
- Prima interpreta le istruzioni scritte in PHP, recupera gli eventuali dati richiesti prelevandoli dai file o dai database del server e in seguito restituisce una pagina Web visualizzabile dal browser.

Programmazione lato server

Linguaggio PHP

- **PHP** (acronimo di PHP: *Hypertext Preprocessor*, preprocessore di ipertesti) è un software che può essere liberamente installato e utilizzato.
- blocco di codice contenente le istruzioni in PHP:

```
<?php
    // elenco di istruzioni in PHP
?>
```

Linguaggio PHP

- Per controllare che il Web server possa supportare il linguaggio PHP:
- file *informazioni.php*

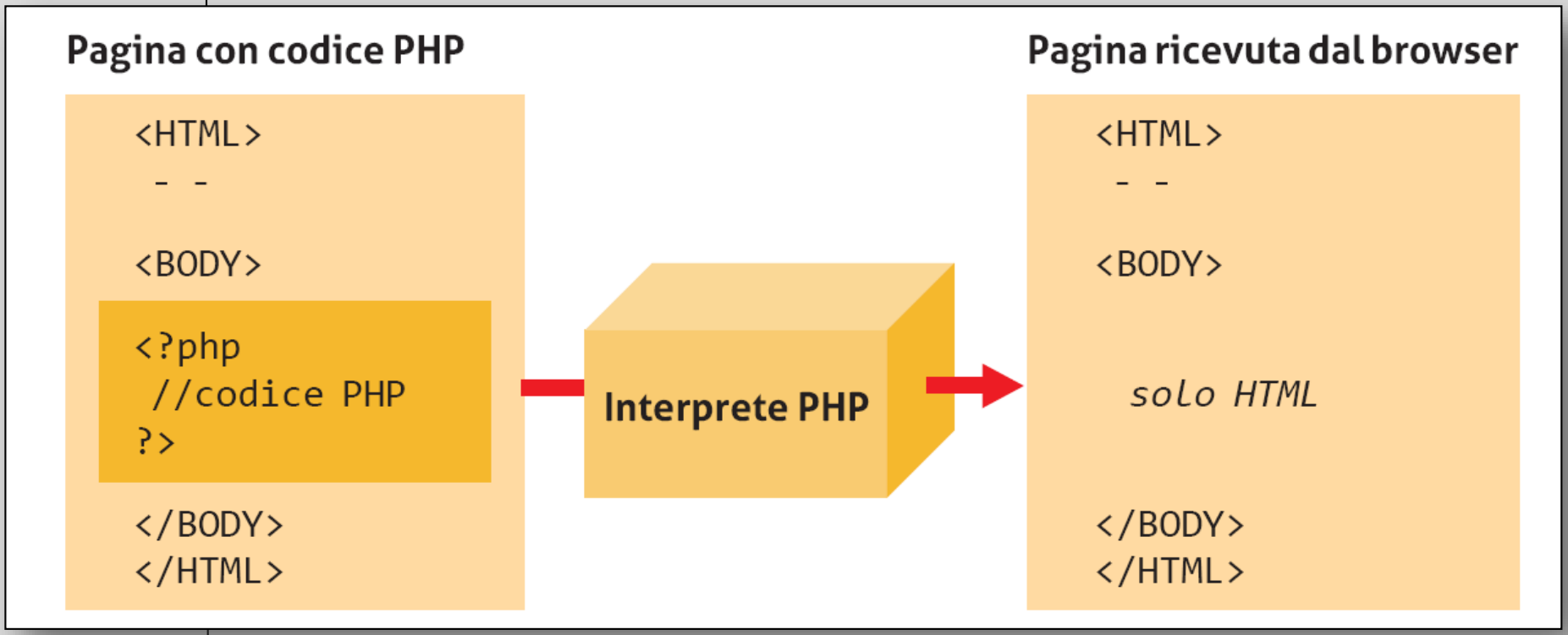
```
<?php
phpinfo();
?>
```

http://*nomeserver*/informazioni.php

- viene generata una pagina Web contenente le informazioni sull'interprete PHP

La pagina PHP

- La pagina Web ricevuta dal browser non contiene codice PHP, ma solo il codice HTML generato dal server sulla base delle istruzioni PHP.



Variabili

- Le variabili sono identificate da un nome preceduto dal segno del dollaro $.
- I nomi delle variabili sono *case-sensitive.*
- Variabili implicitamente dichiarate in base ai valori loro assegnati:

```
<?php
$eta = 19;                          // contiene un intero
$titolo = "I promessi sposi ";      // contiene una stringa
$prezzo= 31.2;                      // contiene un floating point
?>
```

Variabili

- Variabili che memorizzano una stringa:

```
<?php
$nome = "Giovanni";
$saluto = "Buongiorno $nome"; // contiene: Buongiorno Giovanni
echo "$saluto <br />";
$saluto = 'Buongiorno $nome'; // contiene: Buongiorno $nome
echo "$saluto <br />";
?>
```

Operatori

- punto (.) come **operatore di concatenazione**:

```
<?php
$nome = "Giovanni";
$saluto = "Buongiorno " . $nome; // contiene: Buongiorno Giovanni
?>
```

Operatori

- **operatori aritmetici**

Operatore
+
–
*
/
%
++
––

Operatori

- operatore di **assegnamento:** il segno di uguale (=)
- operatori di **assegnamento combinati:**

Operatore
+=
-=
*=
/=
%=
.=

Operatori

- operatori di **confronto** e operatori **logici**

Operatore
==
!=
<
>
<=
>=

Operatore	
!	Not
&&	And
||	Or
xor	Xor

Array

- Un **array** è rappresentato da una variabile che contiene un insieme di valori identificati da un indice.

```
$voti[0]= 8.0;
$voti[1]= 6.5;
. . .
$dati[0] = 30;
$dati[1] = "Roma";
```

- Notazione equivalente, in forma più compatta:

```
$voti = array(8.0, 6.5, 5.5, 7.0);
```

Array associativi

- Negli array **associativi** l'indice è una stringa che è indicata tra doppi apici.

```
$persona["cognome"] = "Rossi";
$persona["nome"] = "Antonio";
```

La struttura *if*

- struttura di **selezione**

```
if ( condizione ) {
// istruzioni eseguite se condizione è vera
} else {
// istruzioni eseguite se condizione è falsa
}
```

La struttura *while*

- Struttura di ripetizione **while**

```
while ( condizione )
{
// istruzioni eseguite mentre condizione si mantiene vera
}
```

La struttura *for*

- Struttura di ripetizione **for**

```
for( inizializzazione; condizione; aggiornamento )
{
// istruzioni eseguite se condizione è vera
}
```

Variabili predefinite

Sono dette **superglobals** e sono raggruppate in array associativi:

- **$_GET**: contiene le variabili passate allo script tramite la modalità GET
- **$_POST**: contiene le variabili passate allo script tramite la modalità POST
- **$_SERVER**: contiene le variabili passate allo script dal server Web
- **$_COOKIE**: contiene le variabili passate allo script tramite i cookie
- **$_SESSION**: contiene le variabili utilizzate per implementare il concetto di session

I nomi sono scritti con i caratteri tutti **maiuscoli**

Invio di dati tramite un form HTML

- form HTML che contiene una casella di testo (*text*) e un pulsante per l'invio (*submit*)
- **action**: nome della pagina PHP
- attributo **method**
  - **Get**
  - **Post**
- I parametri vengono passati allo script PHP per mezzo dell'*array associativo* **$_GET** oppure **$_POST**

Get e Post

- nella modalità **get** i parametri vengono codificati automaticamente dal browser all'interno dell'indirizzo e vengono visualizzati insieme all'URL della pagina Web.
- la modalità **post** viene utilizzata per spedire grandi quantità di dati e in modo che non siano visibili all'utente.

Invio con metodo Get

Pagina HTML

```
<form action="cerca.php" method="get">
Parola: <input type="text" name="parola" />
<input type="submit" value="Cerca" />
</form>
```

Script PHP

```
$stringa = $_GET["parola"]
```

Invio con metodo Post

Pagina HTML

```
<form action="risposta.php" method="post">
Nome <input type="text" name="nome" size="20" />
<input type="submit" value="Cerca" />
</form>
```

Script PHP

```
$nome = $_POST["nome"];
```

Soluzione con unica pagina PHP

- caricata due volte, contiene il form HTML e la parte di visualizzazione:

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
. . .
<input type="submit" name="invio" value="Invia" />
</form>
```

- distinzione tra il primo e il secondo caricamento: struttura **if** e funzione predefinita **isset**, controlla se alla variabile *$_POST['invio']* è stato assegnato un valore:

```
if(isset($_POST['invio'])) {
. . . .
}
```

Parametri nell’indirizzo URL

http://localhost/pagina.php?nome=valore

http://localhost/registra.php?nome=Mario&cognome=Rossi&email=mario.rossi@gmail.com

File di testo

- **fopen** apre il flusso di dati
- (**r** = lettura, **w** = scrittura, **a** = append)

```
$file = fopen("http://www.mioserver.com/prove/dati.txt", "r");
```

- **fclose** chiude il flusso di dati:

```
fclose($file);
```

Lettura e scrittura

- **fgets** legge una linea dal file:

```
$line = fgets($file, 1024);
```

- **fwrite** scrive sul file:

```
fwrite($file, $line . "\n");
```

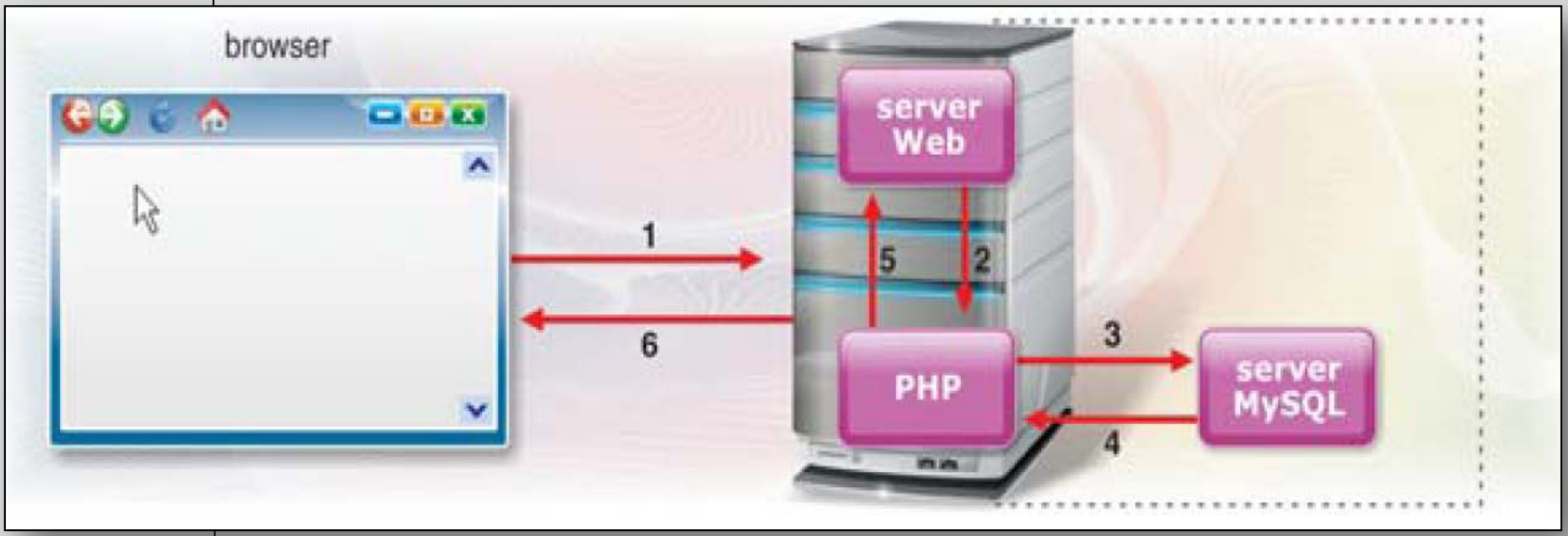
Lettura sequenziale

- Le linee del file vengono lette in modo sequenziale attraverso un'iterazione **while:**

```
while (!feof ($file)) {
    // lettura di una linea
}
```

Accesso ai database

- **MySQL** è un programma *server* che si occupa della gestione di basi di dati.
- Quando un utente richiede una pagina Web lato server:



Server e database

- Connessione al server MySQL: **mysql_connect**

```
mysql_connect($host, $username, $password) or
die('Impossibile connettersi al server: ' .
mysql_error());
```

- Selezione del database: **mysql_select_db**

```
mysql_select_db($db_nome) or die ('Accesso al
database non riuscito: ' . mysql_error());
```

Comandi SQL

- Esecuzione con funzione **mysql_query**
- restituisce un array *$result*

```
$sql = "SELECT * FROM $tab_nome";
$result = mysql_query($sql);
```

Righe della tabella

- La funzione **mysql_fetch_array** riceve come parametro la variabile $*result* e ritorna la riga successiva del risultato.

```
$row = mysql_fetch_array($result)
```

- Per esaminare tutte le righe:

```
while($row = mysql_fetch_array($result))
{
    // Operazioni sulla riga
}
```

Accesso ai dati XML

- libreria **SimpleXML**
- **simplexml_load_file** carica all'interno di un oggetto *SimpleXML* il contenuto di un file e restituisce l'elemento *root* (radice) del documento XML
- struttura **foreach**: esamina la struttura gerarchica del file
- funzione **children**: restituisce i figli di un nodo

Accesso ai dati XML

```
$xml = simplexml_load_file("Categorie.xml");
echo "<h2>" . $xml->getName() . "</h2>";
foreach($xml->children() as $cat)
{
     echo "<p>";
     foreach($cat->children() as $dato)
     {
          echo $dato->getName() . ": "
               . $dato . "<br />";
     }
     echo "</p>";
}
```

Espressione XPath

```
$codice = $_POST["codice"];
$xml =
simplexml_load_file("Prodotti.xml");
echo "Prodotto codice: $codice <br />";
$prod =
$xml->xpath
("/inventario/Prodotto[IDProdotto='$codice']");
echo $prod[0]->NomeProdotto . "<br />";
```