

# Laboratorio Progettazione Web

## **Il linguaggio PHP – Lezione 5**

Andrea Marchetti – IIT-CNR  
andrea.marchetti@iit.cnr.it  
2011/2012

# Outline

- Linguaggi di programmazione
  - @ richiami generali
- Introduzione al PHP
- Variabili
  - Stringhe e funzioni su stringhe
- Istruzioni di assegnamento

# I Linguaggi di programmazione

- Un linguaggio di programmazione gestisce due cose
  - Dati
    - costanti, variabili, tipi di dati, **array**
  - Elaborazione
    - Operatori, condizioni, cicli, **funzioni**

# Concetti generali

**Algoritmo** procedimento per la soluzione di una **classe di problemi** attraverso un numero finito di passi.

**Programma** codifica di un algoritmo in un certo **linguaggio di programmazione** come sequenza ordinata di **istruzioni** che, a partire da dei **dati in ingresso**, restituisce dei **risultati in uscita** in seguito alla loro elaborazione.

Traduttore

**Eseguibile** programma scritto in **linguaggio macchina** ovvero nel formato adatto ad essere caricato dal **sistema operativo**, e all'architettura hardware del **processore** che lo esegue

# Traduttori

- I **traduttori** sono programmi particolari che provvedono a convertire il codice di programmi scritti in un dato linguaggio di programmazione (***sorgenti***) nella corrispondente rappresentazione in linguaggio macchina (programmi ***oggetto*** o ***eseguibili***)

# Traduzione

codice sorgente (scritto  
in qualche linguaggio  
di alto livello)

$x = y + 2$

traduttore

codice oggetto ("eseguibile")

```
00010100 11001011
00010110 00000010
00010111 11001111
```

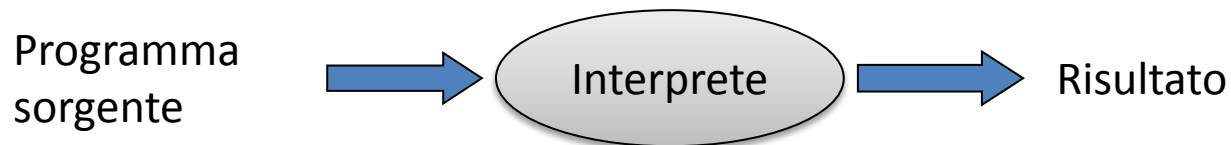
indirizzo di y

indirizzo di x

# Due tipi di traduttori

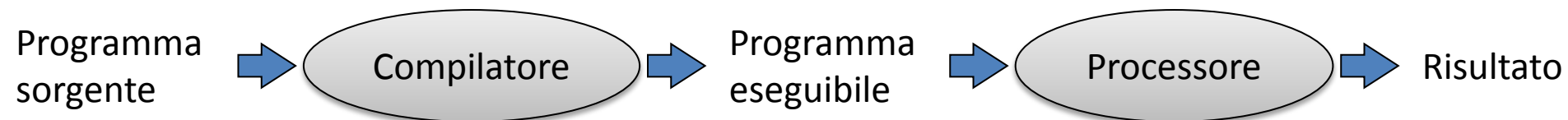
- **Interpreti**

- traducono ed eseguono direttamente ciascuna istruzione del programma sorgente, istruzione per istruzione



- **Compilatori**

- accettano in input l'intero programma e producono in output la rappresentazione dell'intero programma in linguaggio macchina



# Interprete vs compilatore

- compilazione
  - migliori prestazioni (la traduzione viene effettuata una volta sola)
  - processi di ottimizzazione (compilazione sull'intero programma)
- interprete
  - in caso di modifiche, consente di eseguire il programma non appena il codice sorgente sia stata aggiornato
- Il PHP è un linguaggio interpretato



# Elementi di PHP

- Costanti, variabili, tipi di dati
- Operatori, condizioni, cicli
- Funzioni
- Array
- Interazione con i browser
- Interazione con i DB

# Scheda PHP



	
Sviluppatore	The PHP Group
Ultima versione	5.3.10 (2 febbraio 2012)
S.O.	Multiplatforma
Genere	Linguaggio di scripting
Licenza	Free, open source
Sito web	<a href="http://www.php.net">http://www.php.net</a>

# PHP

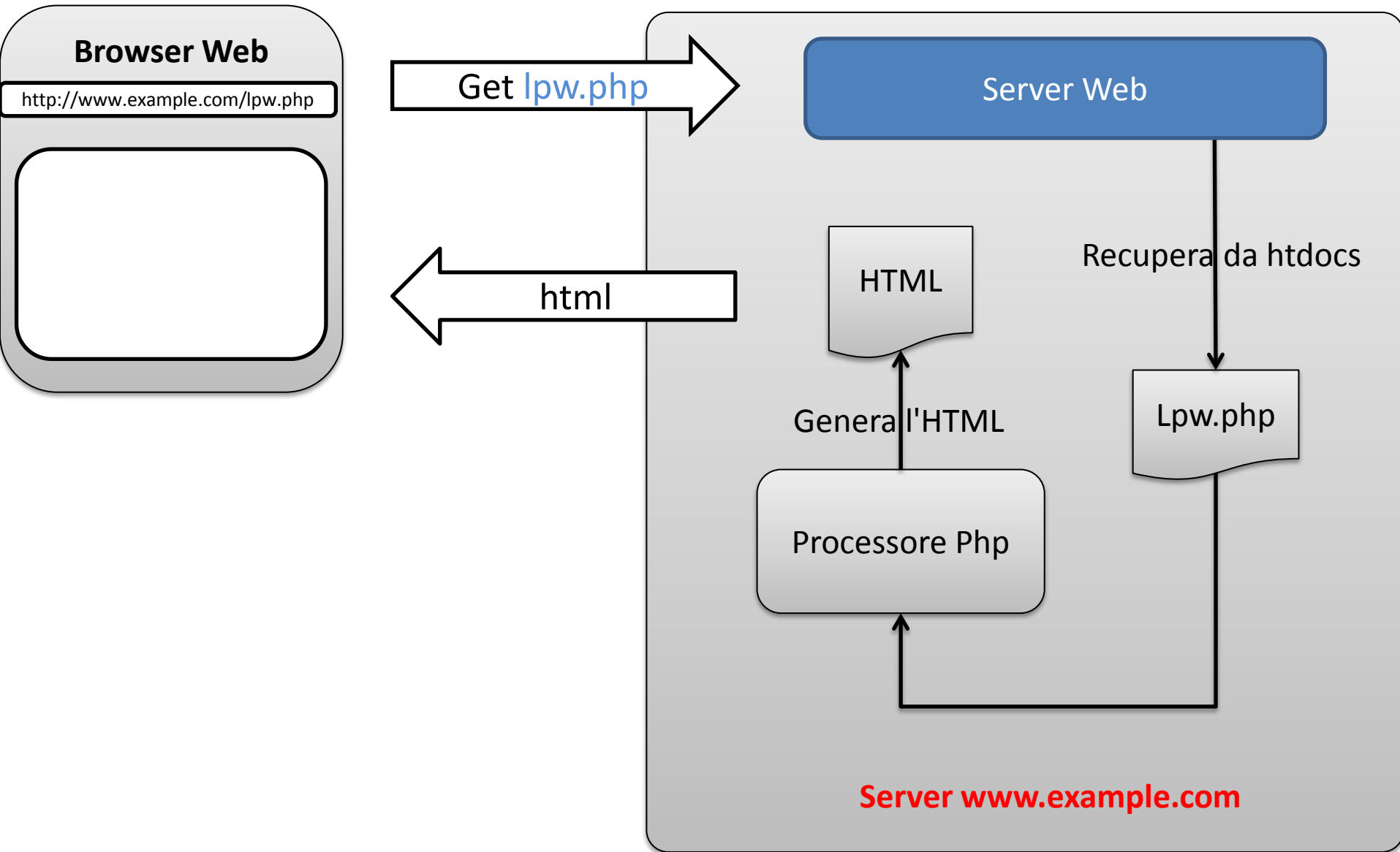
- Supportato da numerose comunità online
- Integrabile con numerose librerie esterne
- Interpretato
- Introdotto da Lerdorf nel '94
- PHP è un **acronimo ricorsivo** che sta per
  - PHP: **H**ypertext **P**reprocessor



# PHP

- Concepito per la **programmazione Web** ovvero la realizzazione di **pagine web dinamiche**
- Il linguaggio PHP è un **linguaggio di script** lato server, viene cioè interpretato da una componente aggiuntiva del server web, detto **processore PHP** o **interprete PHP**
- L'**elaborazione** di codice PHP sul server produce **codice HTML** da inviare al browser dell'utente che ne fa richiesta
- Permette di interagire con vari **database**, tipicamente MySQL ma anche Oracle, Postgres e molti altri.

# Funzionamento PHP



# Caratteristiche di uno script PHP

- Requisito: Apache con il processore PHP
- Termina con l'estensione .php
  - Quando il server gestisce un documento richiesto con estensione .php lo passa direttamente al Processore PHP
  - Si può configurare il server web affinché invochi sempre il processore php anche per documenti con estensione .htm o .html . Utile se si vuole nascondere che stiamo usando php.
- Il processor php restituisce alla fine dell'elaborazione un file HTML
- Se il documento .php contiene solo codice HTML il risultato sarà il documento stesso

# Programmi PHP per Applicazioni Web

```
<?php
echo "<html>";
echo "Oggi è il " . date("d/m/y");
echo "<br>";
echo "e sono le ore " . date("h:i:s");
echo "</html>";
?>
```

```
<html>
Oggi è il <?php echo date("d/m/y"); ?><br>
Sono le ore <?php echo date("h:i:s"); ?>
</html>
```

Processore PHP

Il Processore PHP interpreta solo il codice contenuto tra `<?php` e `?>` il resto lo fa passare

```
<html>
Oggi è il 22/03/2012<br>
e sono le ore 15:00
</html>
```

L'esecuzione del programma PHP produce una pagina HTML

# Delimitatori PHP

- Il codice PHP si intervalla al codice HTML.
- I delimitatori permettono al server web di distinguerlo dall'HTML
- I delimitatori di PHP sono i caratteri:
  - `<?php` per indicare l'inizio (l'uso `<?` È deprecato)
  - `?>` per indicare la fine



# Test Esecuzione PHP

```
<?php
echo "<html>";
echo "Oggi è il " . date("d/m/y");
echo "</br>";
echo "e sono le ore " . date("h:i:s");
echo "</html>";
?>
```

- Si colloca il file php (ex prova.php) sotto la radice htdocs e si invoca da browser
  - http://localhost/prova.php
- Si invoca direttamente il processore php passandogli il file
  - php -f prova.php
  - In questo secondo caso non occorre che il risultato dell'esecuzione sia in HTML in quanto non viene passato al browser
  - Assicurarsi di poter vedere il processore php dalla directory in cui si trova il programma( PATH=C:\....\PHP\ )

# File di configurazione di PHP: php.ini

- I parametri di funzionamento di PHP sono definiti in un apposito file, denominato `php.ini` che il **server web legge ad ogni riavvio**
- In questo file sono definiti alcuni parametri con i valori di default
- Non è necessario modificare il file per il corretto funzionamento di PHP, i parametri predefiniti generalmente sono sufficienti
- I parametri riguardano molti degli aspetti di PHP, ad esempio path dei file, uso della sessioni e dei cookie, la definizione della timezone, i messaggi di errore

# Primo esempio

L'istruzione di stampa di PHP è `echo` (o `print`)

```
<?php echo "Ciao"; ?>
```

oppure

```
<?php echo("Ciao"); ?>
```

oppure

```
<?php print "Ciao"; ?>
```

# Esempio

```
<HTML>
```

```
  <HEAD><TITLE>Esempio PHP</TITLE></HEAD>
```

```
<BODY>
```

Questo testo è in HTML

```
<P>
```

```
<?php echo "questo testo è in PHP"; ?>
```

```
<P>
```

questo testo è ancora in HTML

```
<?php echo "questo testo è ancora in PHP"; ?>
```

```
</BODY>
```

```
</HTML>
```

# Sintassi di Base

- I comandi in PHP terminano con il carattere **;**
- Dimenticarlo costituisce il 90% dei primi errori
  - `echo "Salve";`
- Le variabili in PHP sono precedute dal carattere **\$**
  - `$Eta=18;`

# Commenti

- I commenti sono utili per
  - spiegare il codice scritto (USARLI SPESSO!!!)
  - per disabilitare temporaneamente porzioni di programma.
- Esistono tre tipi di commenti:
  1. `/*` Questo è un commento a riga multipla, fate attenzione a non annidarli `*/`
  2. `//` commento a riga singola
  3. `#` commento a riga singola

# Variabili

- Una **variabile** identifica una *porzione di memoria* destinata a contenere dei dati, che possono essere modificati nel corso dell'esecuzione di un programma.
- Una **variabile** è acceduta tramite un **nome** definito in fase di **dichiarazione della variabile**
- Una variabile prende un determinato valore mediante un'operazione di **assegnamento**

– età = 18;

età



18

- Una **costante** è invece una porzione di memoria che mantiene lo stesso valore per tutta la durata dell'esecuzione del programma.
- Nei linguaggi tipizzati, ciascuna variabile è caratterizzata da un **tipo di dato**, che restringe i valori accettabili

# Variabili in PHP

- La gestione delle variabili in PHP è molto semplificata
- La dichiarazione di una variabile avviene al suo primo utilizzo
  - `$indirizzo = "Via Guglielmo Marconi, 1" ; /* posso usare una variabile anche senza dichiararla*/`
- PHP non è un linguaggio tipizzato
  - `$età = 18; // non c'è modo di dichiarare che è una variabile intera`
  - `$età = "diciotto"; /* posso assegnare alla stessa variabile tipi di dati differenti nello stesso programma */`



# Vincoli sui nomi di variabili

- Le variabili in PHP si denotano con una sequenza di caratteri preceduti dal simbolo **\$**
- I nomi sono **case sensitive**
  - **Età** diverso da **età**
- Devono iniziare con una lettera o il carattere underscore  
–
- Possono contenere solo i caratteri: **a-z, A-Z, 0-9, \_**

\$miavar  
\$\_ENV  
\$var45

SI

~~miavar  
\$2pippo~~

NO

# Esempio

```
<HTML><HEAD><TITLE>Esempio PHP</TITLE></HEAD>
```

```
<BODY>
```

Questo testo è in HTML

```
<?php
```

```
    // memorizzo e visualizzo il mio nome
```

```
    $nome = "Andrea";
```

```
    echo $nome;
```

```
?>
```

```
</BODY></HTML>
```

# Tipi delle variabili

Un tipo è la descrizione del formato dei valori che una variabile può assumere

In PHP non devo dichiarare il tipo di una variabile, sarà assegnato al primo uso della variabile

Una variabile PHP può cambiare tipo

- Boolean (TRUE, FALSE)
  - `$pagato = FALSE;`
- Integer
  - `$count=1;`
- Float, double
  - `$miavar=1.456;`
- String
  - `$messaggio="benvenuto";`

# Variabili numeriche

- `$a = 4;`
- `$b = 2;`
- `echo $a * $b; // produce 8`
- `$a = 2.3;`
- `echo $a * $b; // produce 4.6`

# Operatori aritmetici

Operatore	Descrizione	Esempio $\$a=4; \$b=2;$
+	Addizione	$\$a + \$b; // 6$
-	Sottrazione	$\$a - \$b; // 2$
*	Moltiplicazione	$\$a * \$b; // 8$
/	Divisione	$\$a / \$b; // 2$
%	Modulo	$\$a \% \$b; // 0$
++	Incremento di 1	$\$a++; // 5$
--	Decremento di 1	$\$a--; // 3$

# Stringhe

- Le stringhe sono sequenza di caratteri alfanumerici.
- Possono essere definite con i caratteri ' oppure ''
- L'unione di stringhe si effettua con il carattere punto .
  - `$nome = "Mario";`
  - `$cognome="Rossi";`
  - `$nomeintero=$nome . $cognome; // = "MarioRossi"`
  - `$nomeintero=$nome . " " . $cognome; // = "Mario Rossi"`

# Stringhe

- Differenza tra virgoletta singola e doppia

```
$età=18;
```

```
echo 'Anna ha $età anni'; // letterale
```

```
echo "Anna ha $età anni"; // interpretato
```

```
/* la prima istruzione produce
```

```
Anna ha $età anni
```

```
la seconda istruzione produce
```

```
Anna ha 18 anni : provare */
```

# Stringhe

- Le stringhe che contengono un numero nella parte iniziale possono essere convertite in numero

Esempio:

```
$stringa="45 anni";
```

```
$numero=23;
```

```
$somma = $numero + $stringa; // produce 68
```



# Operatori su stringhe

- `strlen(stringa)` restituisce il numero di caratteri della stringa
- `trim/ltrim/rtrim(stringa)`. Trim elimina spazi all'inizio e alla fine della stringa, ltrim a sinistra rtrim a destra
- `substr(stringa, intero1 [,intero2])`. Restituisce la sottostringa che inizia alla posizione intero1 eventualmente fino a intero1+intero2
- `str_replace(str1,str2,str3)` restituisce una nuova stringa dove sostituisce tutte le occorrenze di str1 con str2 in str3.
- `strtolower/strtoupper(stringa)` converte tutti i caratteri in minuscolo/maiuscolo
- Il confronto tra stringhe si effettua con gli usuali operatori di confronto `==`, `<`, `>`

# Esempio manipolazione di stringhe

```
<?php  
$disney="pippo pluto e paperino";  
$n=strlen($disney);  
echo $n;  
?>
```

Restituirà la lunghezza in caratteri della stringa

# Variabili predefinite

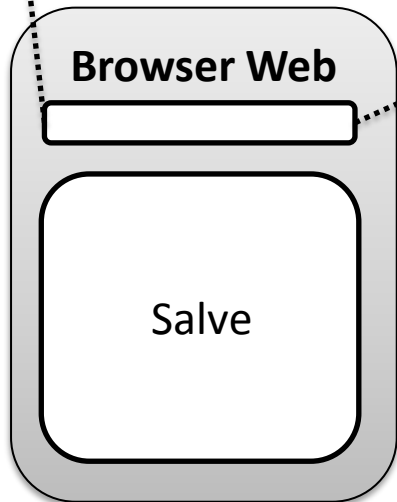
- Fornite direttamente dall'ambiente PHP sono disponibili da programma
- Contengono molte informazioni sull'ambiente di esecuzione del programma o script
- Sono organizzati come array associativi (vedremo meglio in seguito)
- Evitare di ridefinirle.

Variabile predefinite	Descrizione
<code>\$GLOBALS</code>	Tutte le variabili che sono correntemente definite nel programma
<code>\$_SERVER</code>	Create dal server web, danno informazioni sull'ambiente di esecuzione del programma
<code>\$_GET</code>	Variabili passate al programma tramite il metodo HTTP GET
<code>\$_POST</code>	Variabili passate al programma tramite il metodo HTTP POST
<code>\$_FILES</code>	Oggetti passati al programma tramite il metodo HTTP POST
<code>\$_COOKIE</code>	Variabili passate al programma tramite HTTP cookies
<code>\$_REQUEST</code>	<code>\$_GET</code> , <code>\$_POST</code> , <code>\$_COOKIE</code>
<code>\$_ENV</code>	Variabili passate al programma tramite metodi di ambiente

# Passaggio di parametri con la richiesta

## URL

http://localhost/lpw.php?reg=Toscana&pro=Pisa



1. lpw.php
2. reg=Toscana
3. pro=Pisa

lpw.php

```
<?php
$regione   = $_GET['reg'];
$provincia = $_GET['pro'];
echo "la regione è $regione</br>";
echo "la provincia è $provincia";
?>
```

# Esercizio

- Scrivere una **URL con una Query String** dove l'utente possa **passare** una parola e un colore
  - <http://localhost/ex2.php?word=Mondo&color=red>
- Scrivere un **programma php** che **legge** i dati della URL e in base al colore restituisce una pagina HTML contenente la parola nel colore indicato

# Esercizio

- <http://localhost/lpw.php?word=Welcome&color=red>
- La pagina HTML restituita dovrà essere visualizzata dal browser nel seguente modo



Welcome

# Una possibile soluzione

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  Transitional//EN">

<html>
  <head>
    <title>Colored Word</title>
  </head>
  <body>
    <?php
      $color=$_GET['color'];
      $word  =$_GET['word'];
      echo "<p style='color:$color'>$word</p>";
    ?>
  </body>
</html>
```

# Esercizio

- `http://localhost/lpw.php?word=Welcome`
- La pagina HTML restituita dovrà contenere la parola e la sua lunghezza
  - Usare la funzione `strlen()`

Welcome è lunga 7