Tecnologie Web - MFN0634
PHP

Docente

Giancarlo **Ruffo**[ruffo@di.unito.it - twitter: @giaruffo]

- Sviluppo server side
- Sintassi PHP di base
- Includere codice PHP nelle pagine
- Sintassi PHP avanzata
- Caso di studio

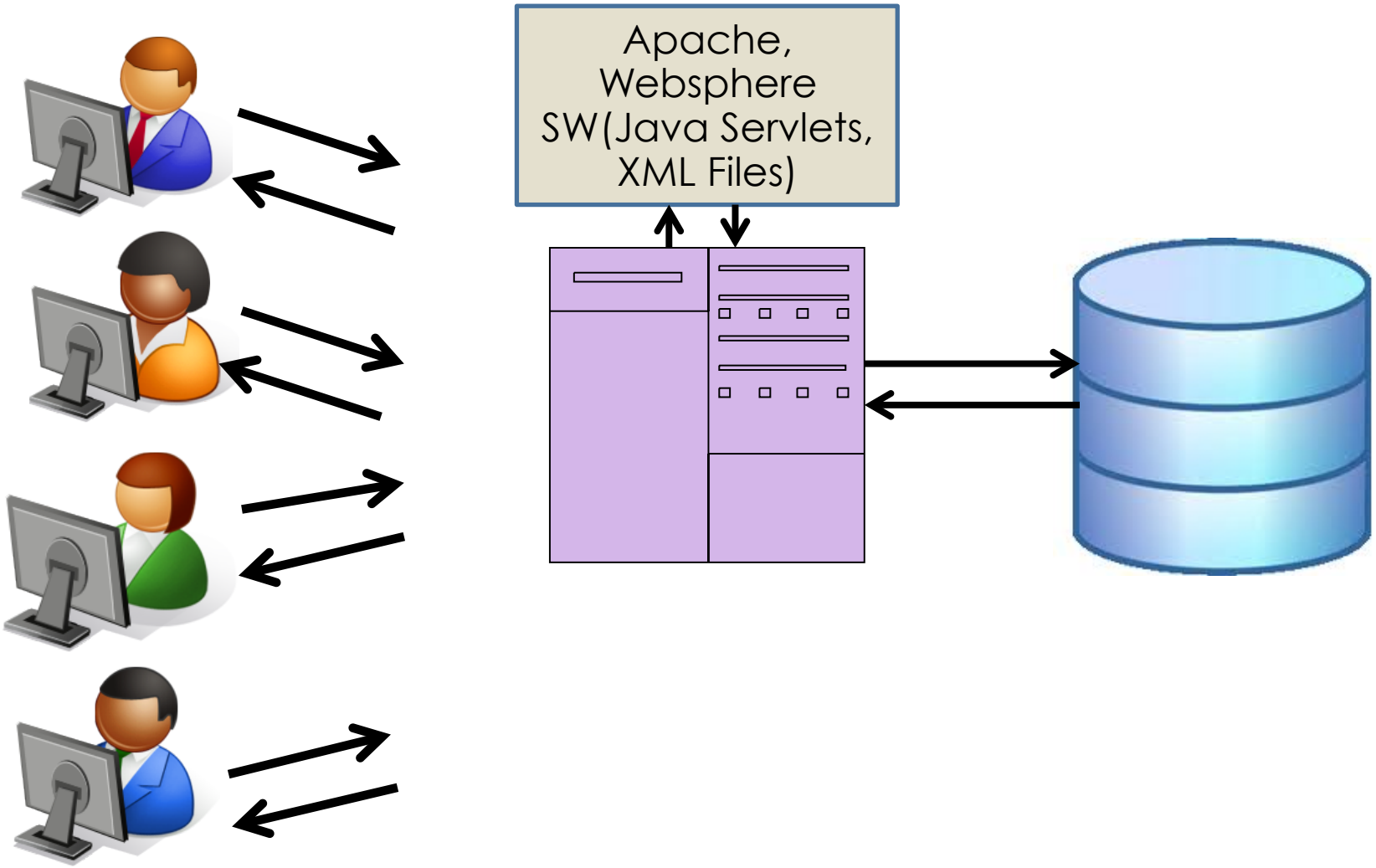
- **Sviluppo server side**
- Sintassi PHP di base
- Includere codice PHP nelle pagine
- Sintassi PHP avanzata
- Caso di studio

URL e server web

http://server/path/file

- Normalmente quando inserisci una URL nel tuo browser:
  - Il S.O. del tuo computer cerca l'indirizzo IP del server usando il DNS
  - Il tuo browser si connette a quell'indirizzo IP e richiede quel dato file
  - Il processo software del server web remoto (e.g. Apache) individua quel file dal suo file system locale
  - Il server invia ti spedisce il contenuto di quel file (come *stream* html)

URL e server web (cont.)



URL e server web (cont.)

http://**www.facebook.com/home.php**

- Qualche URL in realtà specifica dei programmi che il server dovrebbe *eseguire*. Ciò che viene restituito in questi casi è l'output di detti programmi:
  - La URL dell'esempio chiede al server **facebook.com** di eseguire il programma **home.php** e di mandare indietro il suo output

Programmazione Web Server-Side

- I programmi Server-side si scrivono usando uno dei tanti possibili linguaggi/ambienti di programmazione per il web
  - esempi: PHP, Java/JSP, Ruby on Rails, ASP.NET, Python, Perl

Programmazione Web Server-Side (cont.)

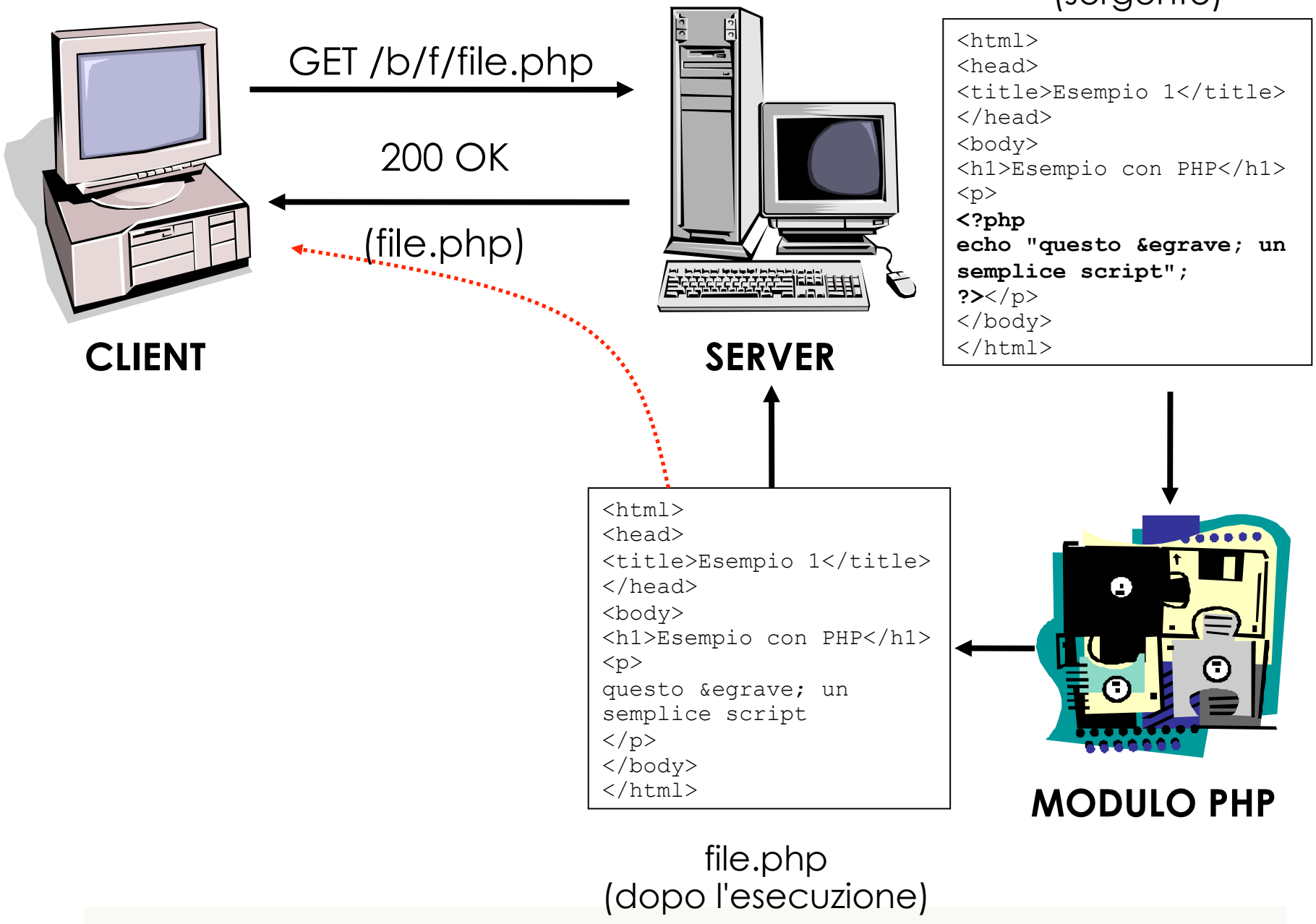
- Chiamato anche *scripting server side*, è in grado di:
  - Modificare dinamicamente, cambiare o aggiungere qualsiasi contenuto ad una pagina Web
  - Rispondere alle richieste dell'utente o ai dati inviati dai moduli (form) HTML
  - Accedere ai dati (spesso gestiti tramite db backend) e restituire il risultato al browser
  - Personalizzare una pagina Web per renderela più fruibile per gli utenti individuali
  - Fornire maggiore protezione dato che il tuo codice lato server non può essere visto da un browser

Programmazione Web Server-Side (cont.)

- Un server Web:
  - Contiene il software che ti consente di eseguire i programmi server side
  - Invia al mittente l'output generato in forma di risposta alle rechieste HTTP
- Ogni linguaggio/ambiente ha i sui pro ed i suoi contro
  - Noi useremo PHP (non necessariamente la scelta migliore in ogni situazione)

Cosa è PHP?

- PHP: "PHP Hypertext Preprocessor"
- Linguaggio di scripting Server-side
- Usato per rendere dinamiche le pagine web:
  - Fornisce contenuti diversi dipendenti dal contesto
  - Si interfaccia con altri servizi: database, e-mail, etc.
  - Autentica gli utenti
  - Processa l'informazione proveniente dai moduli presenti sulla parte client-side dell'applicazione
- Il codice PHP può essere inserito direttamente all'interno del codice (X)HTML


file.php
(sorgente)
<html>
<head>
<title>Esempio 1</title>
</head>
<body>
<h1>Esempio con PHP</h1>

<?php
echo "questo è un
semplice script";
?>
</body>
</html>
GET /b/f/file.php
200 OK
(file.php)
CLIENT
SERVER
<html>
<head>
<title>Esempio 1</title>
</head>
<body>
<h1>Esempio con PHP</h1>

questo è un
semplice script

</body>
</html>
MODULO PHP
file.php
(dopo l'esecuzione)

Perché PHP?

- Gratuito e open source
- Compatibile
  - A gennaio 2013 quasi 100.000.000 di siti attivi usavano PHP.
- Semplice

Hello World!

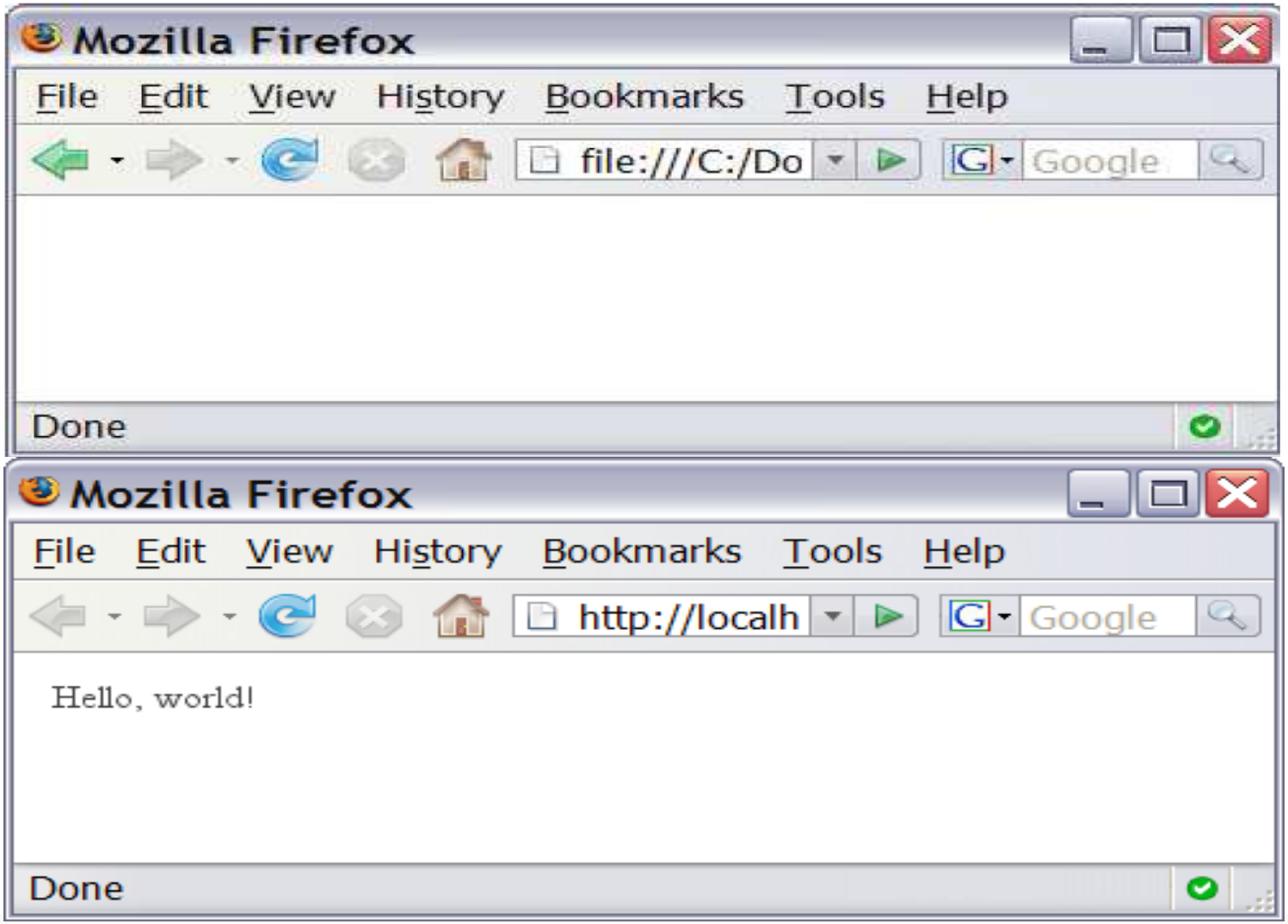
```php
<?php
print "Hello, world!";
?>
```

PHP

```
Hello world!
```

output

Visualizzare l'output PHP



PHP info

- Un altro esempio è il comando che ci dà informazioni sulla configurazione del server. Se dentro al file *.php* si inserisce la seguente linea:

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

si ottengono tutte le informazioni sulla configurazione del web server e del php su cui si sta lavorando.

HTTP Server Agent

Lo script:

```
<html>
<body>
   Il server agent che stai usando &egrave;:
   <?php echo $_SERVER["HTTP_USER_AGENT"]; ?>
</body>
</html>
```

HTTP Server Agent

.... produrrà (su una macchina Windows) il seguente output:

<html>
<body>
Il server agent che stai usando è:
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
</body>
</html>

- Sviluppo server side
- **Sintassi PHP di base**
- Includere codice PHP nelle pagine
- Sintassi PHP avanzata
- Caso di studio

Template della sintassi PHP

```
HTML content
<?php
PHP code
?>
HTML content
<?php
PHP code
?>
HTML content ...                    PHP
```

- Il codice php all'interno dei tag `<?php` and `?>` è eseguito dal server
- Tutto il resto del contenuto è interpretato direttamente come HTML puro
- Possiamo passare continuamente dalla modalità PHP a quella HTML e viceversa

Console output: `print`

```
print "text";
```
PHP

```
print "Hello, World!\n";
print "Escape \"chars\" are the SAME as in Java!\n";
print "You can have
line breaks in a string.";
print 'A string can use "single-quotes". It\'s cool!';
```
PHP

```
Hello world!  Escape "chars" are the SAME as in Java! You can have line
breaks in a string. A string can use "single-quotes". It's cool!
```
output

- Alcuni preferiscono il comando equivalente **`echo`** al posto di **`print`**

Variabili

```
$name = expression;
```
PHP

```
$user_name = "mundruid78";
$age = 16;
$drinking_age = $age + 5;
$this_class_rocks = TRUE;
```
PHP

- I nomi sono *case-sensitive*
- I nomi iniziano sempre con un $, sia in fase di dichiarazione che nell'accesso
- Sono sempre dichiarati nel momento dell'assegnazione (senza dichiarre il tipo)
- E' un linguaggio debolmente tipato (come JavaScript o Python)

Variabili

- Tipi di base: *int, float, boolean, string, array, object, NULL*
  - Verifica il tipo di una variabile con le funzioni `is_type`, e.g. `is_string`
  - La funzione `gettype` restituisce il tipo di una variabile in termini di una stringa
- PHP converte da un tipo all'altro *automaticamente in molti casi*:
  - string → int conversione automatica con +
  - int → float conversione automatica con /
- Cast di tipo esplicito con **(type):**
  - `$age = (int) "21";`

Operatori aritmetici

- + - * / % . ++ --
- = += -= *= /= %= .=
- Molti operatori convertono il tipo implicitamente: 5 + "7" is 12

Commenti

```
# single-line comment
// single-line comment
/*
multi-line comment
*/
```
PHP

- Come in Java, ma anche con #
  - Molto codice PHP adotta commenti con # invece che con //

Tipo `String`

```php
$favorite_food = "Ethiopian";
print $favorite_food[2];
$favorite_food = $favorite_food . " cuisine";
print $favorite_food;
```

PHP

- Indicizzazione a base zero e notazione a parentesi []
- Non esiste un tipo `char`; ogni lettera è comunque una stringa
- L'operatore di concatenazione tra stringhe è . (punto), non +
  - `5 + "2 turtle doves" === 7`
  - `5 . "2 turtle doves" === "52 turtle doves"`
- Si può specificare tra "" oppure tra ''

Funzioni `String`

```php
# index  0123456789012345
$name = "Stefanie Hatcher";
$length = strlen($name);
$cmp = strcmp($name, "Brian Le");
$index = strpos($name, "e");
$first = substr($name, 9, 5);
$name = strtoupper($name);
```

PHP

Funzioni `String` (cont.)

Nome	**Equivalente in Java**
strlen	length
strpos	indexOf
substr	substring
strtolower, strtoupper	toLowerCase, toUpperCase
trim	trim
explode, implode	split, join
strcmp	compareTo

Interpretazione di stringhe

```php
$age = 16;
print "You are " . $age . " years old.\n";
print "You are $age years old.\n"; # You are 16 years old.
```
PHP

- Stringhe dentro " " vengono *interpretate*
  - Variabili che sono richiamate all'interno saranno sostituite dal loro valore
- Stringhe dentro ' ' non vengono interpretate:

```php
print 'You are $age years old.\n'; # You are $age years
old. \n
```
PHP

Interpretazione di stringhe (cont.)

```
print "Today is your $ageth birthday.\n"; # $ageth not
found
print "Today is your {$age}th birthday.\n";
```

PHP

- Se è necessario evitare ambiguità, usare le parentesi {} per delimitare il nome della variabile

Interpretazione di stringhe (cont.)

```php
$name = "Xenia";
$name = NULL;
if (isset($name)) {
print "This line isn't going to be reached.\n";
}
```
PHP

- Una variabile è NULL se
  - Non le è stato assegnato nessun valore (variabili non definite)
  - Le viene assegnato il valore NULL
  - Il suo valore è stato rimosso usando la funzione unset
- Puoi verificare che una variabile sia NULL usando la funzione **`isset`**
- NULL in fase di stampa è equivalente ad una stringa vuota (no output)

Ciclo `for` (uguale a Java)

```
for (initialization; condition; update) {
   statements;
}
```
PHP

```
for ($i = 0; $i < 10; $i++) {
   print "$i squared is " . $i * $i . ".\n";
}
```
PHP

Tipo (booleano) `bool`

```php
$feels_like_summer = FALSE;
$php_is_great = TRUE;
$student_count = 7;
$nonzero = (bool) $student_count; # TRUE
```
PHP

- I valori seguenti sono considerati FALSE (tutti gli altri sono TRUE):
  - 0 e 0.0 (but NON 0.00 o 0.000)
  - "", "0", and NULL (include variabili non definite)
  - Array con 0 elementi
- FALSE in fase di stampa è equivalente ad una stringa vuota (no output); TRUE viene stampata come 1

Condizioni `if/else`

```php
if (condition) {
   statements;
} elseif (condition) {
   statements;
} else {
   statements;
}
```

PHP

Cicli `while` (come in Java)

```php
while (condition) {
   statements;
}
```
PHP

```php
do {
   statements;
} while (condition);
```
PHP

Operazioni matematiche

```php
$a = 3;
$b = 4;
$c = sqrt(pow($a, 2) + pow($b, 2));
```

PHP

Funzioni matematiche

abs	ceil	cos	floor	log	log10	max
min	pow	rand	round	sin	sqrt	tan

Costanti matematiche

M_PI	M_E	M_LN2

Tipi `Int` e `Float`

```php
$a = 7 / 2; # float: 3.5
$b = (int) $a; # int: 3
$c = round($a); # float: 4.0
$d = "123"; # string: "123"
$e = (int) $d; # int: 123
```

PHP

- int per gli interi e float per i reali
- Divisione tra due valori int può restituire un float

Esercizio PHP n. 1 (non consegnare)

- Per il tuo primo esercizio, stampa sul video del tuo browser questa stringa:

"Twinkle, Twinkle little star."

- Dopo, crea due variabili, una per la parola "Twinkle" e l'altra per la parola "star". Stampa la frase di prima sul browser, questa volta sostituendo le variabili alle parole corrispondenti. Poi, cambia il valore di ogni variabile come vuoi tu, e stampa la frase risultante una terza volta. Stampa le tre frasi una dopo l'altra, andando a capo ogni volta.

Esercizio PHP n. 2 (non consegnare)

- PHP fornisce tutti gli operatori standard per le operazioni aritmetiche. Per questo esercizio, dovrai usarle insieme alle variabili per stampare sul browser anche le equazioni corrispondenti. Con questo script che segue, crea le seguenti variabili:

  ```
  $x=10;
  $y=7;
  ```

- Scrivi il codice per stampare le seguenti equazioni:

```
10 + 7 = 17
10 - 7 = 3
10 * 7 = 70
10 / 7 = 1.4285714285714
10 % 7 = 3
```

- Usa I numeri riportati sopra usando soltanto il valore assegnato alle variabili, ma non direttamente con comandi **`echo`**.

Esercizio PHP n. 3 (non consegnare)

- Gli *operatori di assegnamento composti* consentono l'esecuzione di un'operazione su una variabile nel momento stesso in cui alla variabile viene assegnato un nuovo valore. Per questo esercizio, scrivi uno script che riproduca l'output indicato sotto. Lavora soltanto con una variabile evitando di usare operatori aritmetici semplici per riprodurre i valori mostrati nelle frasi riportate sotto.

- Suggerimento: Nello script ogni istruzione termina con la stringa "Il valore adesso è $variable."

Il valore adesso è 8.
Somma 2. Il valore adesso è 10.
Sottrai 4. Il valore adesso è 6.
Moltiplica per 5. Il valore adesso è 30.
Dividi per 3. Il valore adesso è 10.
Incrementa il valore di un'unità. Il valore adesso è 11.
Decrementa il valore di un'unità. Il valore adesso è 10.

Esercizio PHP n. 4 (non consegnare)

- Per questo eserizio, scrivi uno script che usi questa variabile:
  $around="around";

- Gli apici singoli e quelli doppi non funzionano allo stesso modo in PHP. Se usi gli apici singoli (' ') e l'operatore di concatenazione, stampa con il comando echo sullo schermo del browser la seguente frase, usando la variabile che hai creato:

  What goes around, comes around.

Esercizio PHP n. 5 (non consegnare)

- In questo esercizio, userai un'istruzione condizionale per determinare cosa stampare sullo schermo del browser. Scrivi uno script in grado di recuperare il mese corrente e stampare una frase che dipende se è agosto oppure no:

E' agosto, e c'e' davvero caldo.
Non e' agosto, per lo meno non e' il periodo piu' caldo dell'anno.

- Suggerimento: la funzione per recuperare il nome completo del mese corrente è '*date('F', time())*'

Esercizio PHP n. 6 (non consegnare)

- I cicli sono molto utili per creare liste e tabelle. In questo esercizio, scriverai un ciclo per creare una lista di equazioni.

- Usando un ciclo for, scrivi uno script che manda al browser una lista di quadrati per i numeri da 1 a 12.
  Usa il formato "1 * 1 = 1" e assicurati di includere il codice che che stampi ogni formula in una linea differente.

Esercizio PHP n. 7 (non consegnare)

- La tebelle HTML hanno molto codice ripetitivo – un contesto perfetto per usare dei cicli. Puoi sfruttarlo al massimo se annidi più cicli for.

- In questo esercizio, usa due cicli for, uno annidato dentro l'altro. Crea la seguente tabella pitagorica:

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

- Sviluppo server side
- **Sintassi PHP di base (cont.)**
- Includere codice PHP nelle pagine
- Sintassi PHP avanzata
- Caso di studio

Errori e debugging

- Certamente non ti accadrà mai, ma nel caso dovessi fare degli errori…
- Un server web normalmente disattiva la stampa a video degli errori
- Bisogna configurare opportunamente il server in modo da usare un tool appropriato di gestione dell'errore (es. Xdebug)
- Gli IDE più diffusi (e.g., NetBeans, Eclipse) sono in grado di offrire funzionalità di debugging avanzate

error_reporting

- Il codice PHP può generare diverse notifiche: warning ed errori di vario tipo
  - Possono essere errori sintattici o di runtime
- Per attivare l'inserimento nel flusso HTML dei messaggi di errore, usare:

```
error_reporting(E_ALL)
```
oppure
```
error_reporting(E_ALL | E_STRICT)
```

- Per disattivare il reporting:

```
error_reporting(E_NONE)
```

- Sviluppo server side
- Sintassi PHP di base
- **Includere codice PHP nelle pagine**
- Sintassi PHP avanzata
- Caso di studio

Scrivere tag HTML PHP = stile errato!

```
<?php
print "<!DOCTYPE html>\n";
print "<html>\n";
print " <head>\n";
print " <title>Geneva's web page</title>\n";
...
for ($i = 1; $i <= 10; $i++) {
print  "<p> I can count to $i! </p>\n";
}
?>
```

HTML

- Lo stile PHP migliore è quello che minimizza il numero di istruzioni print/echo
- Se non usiamo print, come inseriamo il contenuto dinamico dentro la pagina?

Espressioni blocco PHP

```
<?= expression ?>
```
PHP

```
<h2> The answer is <?= 6 * 7 ?> </h2>
```
PHP

The answer is 42

output

- Espressione blocco PHP: un piccolo pezzo di codice php che valuta ed include automaticamente il risultato di un'espressione all'interno del codice HTML
  - `<?= expression ?>` è equivalente a: `<?php print expression ?>`

Esempio

```
<!DOCTYPE html>
<html>
<head><title>CSE 190 M: Embedded PHP</title></head>
<body>
<?php
for ($i = 99; $i >= 1; $i--) {
?>
<p> <?= $i ?> bottles of beer on the wall, <br />
<?= $i ?> bottles of beer. <br />
Take one down, pass it around, <br />
<?= $i - 1 ?> bottles of beer on the wall. </p>
<?php
}
?>
</body>
</html>
```

PHP

Errori comuni: parentesi non chiuse, simbolo = mancante

```
...
<body>
<p>Watch how high I can count:
<?php
for ($i = 1; $i <= 10; $i++) {
?>
   <? $i ?>
</p>
</body>
</html>
```

PHP

- Se dimentichi di chiudere le parentesi, genererai un errore: 'unexpected `$end`'
- Se dimentichi = in `<?=`, l'espressione non produrrà alcun output

Espressioni blocco complesse

```
...
<body>
<?php
for ($i = 1; $i <= 3; $i++) {
    ?>
    <h<?= $i ?>>This is a level <?= $i ?> heading.</
h<?= $i ?>>
    <?php
}
?>
</body>
```

PHP

This is a level 1 heading.

This is a level 2 heading.

This is a level 3 heading.

output

- Sviluppo server side
- Sintassi PHP di base
- Includere codice PHP nelle pagine
- **Sintassi PHP avanzata**
- Caso di studio

Funzioni

```php
function name(parameterName, ..., parameterName) {
   statements;
}
```
PHP

```php
function quadratic($a, $b, $c) {
   return -$b + sqrt($b * $b - 4 * $a * $c) / (2 *
$a);
}
```
PHP

- Non si dichiarano i tipi dei parametri e del valore di ritorno
- Una funziona senza l'istruzione di `return` restituirà implicitamente NULL

Chiamata di funzioni

```
name(expression, ..., expression);
```

```
$w = 163;  # pounds
$h = 70;   # inches
$my_bmi = bmi($w, $h);
```

- Se passate il numero di parametri sbagliati, generate un errore

Valori di default dei parametri

```php
function print_separated($str, $separator = ", ") {
   if (strlen($str) > 0) {
      print $str[0];
      for ($i = 1; $i < strlen($str); $i++) {
         print $separator . $str[$i];
      }
   }
}
```
PHP

```php
print_separated("hello"); # h, e, l, l, o
print_separated("hello", "-"); # h-e-l-l-o
```
PHP

- Se non passi un valore, quello di default sarà utilizzato

Scope delle variabili: locali e globali

- Variabili dichiarati dentro una funzione sono **locali**; le altre sono **globali**.
- Se una funzione vuole usare una variabile globale, deve usare l'istruzione `global`
  - Non abusarne; usa piuttosto i parametri passati alla funzione stessa.

```
$school = "UW";                          # global
...

function downgrade() {
  global $school;
  $suffix = "(Wisconsin)";               # local

  $school = "$school $suffix";
  print "$school\n";
}
```

Array

```php
$name = array();          # create
$name = array(value0, value1, ..., valueN);
$name[index]              # get element value
$name[index] = value;     # set element value
$name[] = value;          # append
```

PHP

```php
$a = array();  # empty array (length 0)
$a[0] = 23;    # stores 23 at index 0 (length 1)
$a2 = array("some", "strings", "in", "an", "array");
$a2[] = "Ooh!";
               # add string to end (at index 5)
```

PHP

- Append: usa la notazione a parentesi [] senza un indice all'interno
- Il tipo dell'elemento non è specificato: un array può contenere elementi di tipi diversi

Funzioni per Array

function name(s)	description
count	number of elements in the array
print_r	print array's contents
array_pop, array_push, array_shift, array_unshift	using array as a stack/queue
in_array, array_search, array_reverse, sort, rsort, shuffle	searching and reordering
array_fill, array_merge, array_intersect, array_diff, array_slice, range	creating, filling, filtering
array_sum, array_product, array_unique, array_filter, array_reduce	processing elements

Esempio

```php
$tas = array("MD", "BH", "KK", "HM", "JP");
for ($i = 0; $i < count($tas); $i++) {
   $tas[$i] = strtolower($tas[$i]);
}
$morgan = array_shift($tas);
array_pop($tas);
array_push($tas, "ms");
array_reverse($tas);
sort($tas);
$best = array_slice($tas, 1, 2);
```

PHP

- Un array in PHP sostituisce molti altri tipi di collezioni in Java
  - list, stack, queue, set, map, ...

Ciclo `foreach`

```
foreach ($array as $variableName) {
   ...
}
```
PHP

```
$fellowship = array("Frodo", "Sam", "Gandalf",
"Strider", "Gimli", "Legolas", "Boromir");
print "The fellowship of the ring members are: \n";
for ($i = 0; $i < count($fellowship); $i++) {
   print "{$fellowship[$i]}\n";
}
print "The fellowship of the ring members are: \n";

foreach ($fellowship as $fellow) {
   print "$fellow\n";
}
```
PHP

Array multidimensionali

```php
<?php $AmazonProducts = array( array("BOOK",
"Books", 50),
                        array("DVDs", "Movies", 15),
                        array("CDs", "Music", 20)
                    );
for ($row = 0; $row < 3; $row++) {
   for ($column = 0; $column < 3; $column++) { ?>
       <p> | <?= $AmazonProducts[$row][$column] ?>
   <?php } ?>
      </p>
<?php } ?>
```

PHP

Array multidimensionali (cont.)

```php
<?php $AmazonProducts = array( array("Code" =>"BOOK",
"Description" => "Books", "Price" => 50),
                    array("Code" => "DVDs", "Description"
=> "Movies", "Price" => 15),
                    array("Code" => "CDs", "Description"
=> "Music", "Price" => 20)
                  );
for ($row = 0; $row < 3; $row++) { ?>
   <p> | <?= $AmazonProducts[$row]["Code"] ?> | <?=
$AmazonProducts[$row]["Description"] ?> | <?=
$AmazonProducts[$row]["Price"] ?>
      </p>
<?php } ?>
```

PHP

Funzioni di confronto per `String`

Name	**Function**
strcmp	compareTo
strstr, strchr	find string/char within a string
strpos	find numerical position of string
str_replace, substr_replace	replace string

Esempi su confronto di tipi `String`

```php
$offensive = array( offensive word1, offensive
word2);
$feedback = str_replace($offcolor, "%!@*",
$feedback);
```

PHP

```php
$test = "Hello World! \n";
print strpos($test, "o");
print strpos($test, "o", 5);
```

PHP

```php
$toaddress = "feedback@example.com";
if(strstr($feedback, "shop")
   $toaddress = "shop@example.com";
else if(strstr($feedback, "delivery")
   $toaddress = "fulfillment@example.com";
```

PHP

Espressioni regolari

```
[a-z]at      #cat, rat, bat…
[aeiou]
[a-zA-Z]
[^a-z]       #not a-z
[[:alnum:]]+   #at least one alphanumeric char
(very) *large  #large, very very very large…
(very){1, 3}      #counting "very" up to 3
^bob         #bob at the beginning
com$         #com at the end          PHPRegExp
```

- Espressioni regolari: una regolarità (un pattern) all'interno di un testo
- PHP include:
  - POSIX
  - **Espressioni regolari del Perl**

Esercizio 1 su array PHP (non consegnare)

- *Gli array ti consentono di assegnare più valori ad una variabile. Prova a scrivere una variabile array contente differenti condizioni atmosferiche: pioggia, sole, nuvole, grandine, nebbia, neve, vento. Usando la variabile array per tutte queste condizioni atmosferiche, stampa il seguente messaggio sul browser:*

Abbiamo avuto tutti i tipi di clima questo mese. All'inizio del mese abbiamo avuto neve e vento. Poi è arrivato il sole con un po' di nuvole ed infine un po' di pioggia. Almeno non abbiamo avuto grandine e nebbia.

- *Non dimenticare di includere un titolo per la tua pagina, sia nell'header che nella pagina stessa.*

Esercizio 2 su array PHP (non consegnare)

- *Crea una lista con le 10 città più grandi del mondo.usando i seguenti valori: Tokyo, Mexico City, New York City, Mumbai, Seoul, Shanghai, Lagos, Buenos Aires, Cairo, London.*
- *Stampa questi valori, separati da virgola, sul browser. Usate un ciclo per accedere ai singoli valori dell'array. Ordina quindi l'array, stampando di nuovo i singoli valori sul browser uando una lista html non ordinata (<ul>), usando ancora una volta un ciclo.*
- *Aggiungi le seguenti città Add the following cities to the array: Los Angeles, Calcutta, Osaka, Beijing. Ordina nuovamente l'array e stampalo ancora una volta usando una lista non ordinata.*

- Sviluppo server side
- Sintassi PHP di base
- Includere codice PHP nelle pagine
- **Sintassi PHP avanzata (cont.)**
- Caso di studio

Inclusione file in PHP

- Inserisci il contenuto di un un file PHP in un altro file PHP, prima che il server lo esegua, con queste funzioni:
  - `include(f)` copia a runtime il contenuto del file f all'interno del file che chiama la funzione. Se il file non è trovato, genera un warning, ma l'esecuzione non sarà interrotta
  - `require(f)` come include, ma se il file non è trovato, genera un errore, che blocca l'esecuzione
- Preferite però le funzioni seguenti:
  - `include_once(f)` come include, ma evita copia ed incolla a cascata se lo stesso file è incluso più volte
  - `Require_once(f)` come require, ma evita copia ed incolla a cascata se lo stesso file è incluso più volte

Esempio `include()`

```php
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/contact.php">Contact Us</a>
```

PHP

```php
<html>
<body>

<div class="leftmenu">
<?php include("menu.php"); ?>
</div>

<h1>Welcome to my home page.</h1>
<p>I have a great menu here.</p>

</body>
</html>
```

PHP

Funzioni I/O con file in PHP

funzioni	categoria
file, file_get_contents, file_put_contents	leggere/scrivere file interi
basename, file_exists, filesize, fileperms, filemtime, is_dir, is_readable, is_writable, disk_free_space	Recupera informazioni sul file
copy, rename, unlink, chmod, chgrp, chown, mkdir, rmdir	Manipolazione file e directory
glob, scandir	Leggere directory

Leggere da/scrivere su file

contents of foo.txt	file("foo.txt")	file_get_contents ("foo.txt")
Hello how are you? I'm fine	array("Hello\n", #0 "how are\n", #1 "you?\n", #2 "\n", #3 "I'm fine\n" #4)	"Hello\n how are\n you?\n \n I'm fine\n"

- `file` restituisce le righe di un file dentro un array
- `file_get_contents` restituisce l'intero contenuto di un file in una stringa

Leggere/scrivere un file intero

```php
# reverse a file
$text = file_get_contents("poem.txt");
$text = strrev($text);
file_put_contents("poem.txt", $text);
```
PHP

- `file_get_contents` restituisce l'intero contenuto di un file in una stringa
- `file_put_contents` sostituisce l'intero contenuto precedente di un file con quello contenuto in una stringa

Opzione append

```php
# add a line to a file
$new_text = "P.S. ILY, GTG TTYL!~";
file_put_contents("poem.txt", $new_text,
FILE_APPEND);
```

PHP

Contenuto precedente

Roses are red,
Violets are blue.
All my base,
Are belong to you.

Nuovo contenuto

Roses are red,
Violets are blue.
All my base,
Are belong to you.
P.S. ILY, GTG TTYL!~

La funzione `file`

```php
# display lines of file as a bulleted list
$lines = file("todolist.txt");
<?php
foreach ($lines as $line) {
   ?>
   <li> <?= $line ?> </li>
<?php
}
?>
```
PHP

- Restituisce le righe di un file sotto forma di un array di stringhe
  - Ultimo carattere di ogni stringa: \n
  - Per togliere il \n, usa questo parametro opzionale:

```php
$lines = file("todolist.txt",FILE_IGNORE_NEW_LINES);
```
PHP

Estrarre i valori dall'array: `list`

```
list($var1, ..., $varN) = array;
```
PHP

```
$values = array("mundruid", "18", "f", "96");
...
list($username, $age, $gender, $iq) = $values;
```
PHP

- La funzione list accetta una lista di nomi di variabili separate da virgola come parametri
- Usa questa funzione per "spacchettare" il contenuto dell'array in diverse variabili

File di lunghezza fissa, `file` e `list`

```
Xenia Mountrouidou
(919)685-2181
570-86-7326
```

contents of file personal.txt

```
list($name, $phone, $ssn) = file("personal.txt");
```

PHP

- Legge il file in reads con un array e ne memorizza subito il contenuto di ogni linea in una variabile separata
- Devi conoscere ovviamente la lunghezza esatta del file in anticipo (oltre che il formato)

Dividere/unire stringhe

```php
$array = explode(delimiter, string);
$string = implode(delimiter, array);
```
PHP

```php
$class = "CS 380 01";
$class1 = explode(" ", $s); # ("CS", "380", "01")
$class2 = implode("...", $a); # "CSE...380...01"
```
PHP

- `explode` e `implode` convertono stringhe in array

Esempio `explode`

```
Harry Potter, J.K. Rowling
The Lord of the Rings, J.R.R. Tolkien
Dune, Frank Herbert
```
contents of input file books.txt

```php
<?php foreach (file("books.txt") as $book) {
   list($title, $author) = explode(", ", $book);
   ?>
   <p> Book title: <?= $title ?>, Author: <?=
$author ?> </p>
<?php
}
?>
```
PHP

Leggere il contenuto delle directory

funzione	**descrizione**
scandir	Restituisce un array contenente tutti i nomi dei file della cartella (restituisce solo i nomi, come `"myfile.txt"`)
glob	Restituisce un array contenente tutti i nomi dei file corrispondenti ad un pattern (restituisce i cammini, come `"foo/bar/myfile.txt"`)

Esempio di `scandir`

```php
<ul>
<?php
$folder = "taxes/old";
foreach (scandir($folder) as $filename) {
   ?>
   <li> <?= $filename ?> </li>
<?php
}
?>
</ul>
```

PHP

- .
- ..
- 2009_w2.pdf
- 2007_1099.doc

output

Esempio di `glob`

```php
# reverse all poems in the poetry directory
$poems = glob("poetry/poem*.dat");
foreach ($poems as $poemfile) {
   $text = file_get_contents($poemfile);
   file_put_contents($poemfile, strrev($text));
   print "I just reversed " . basename($poemfile);
}
```

PHP

- `glob` può usare le "wildcard" con il carattere *
- La funzione `basename` recupera ogni directory dal cammino di un file

Perché si usano oggetti e classi?

- PHP è fondamentalmente un linguaggio procedurale
- Programmi con poche linee di codice possono essere scritti facilmente senza usare oggetti e classi
- Per programmi più lunghi, comunque, diventa complicato gestire molte funzioni disorganizzate
- Raggruppare *dati e comportamenti* con degli oggetti aiuta a gestire le dimensioni e la complessità del programma

Costruire e usare gli oggetti

```php
# construct an object
$name = new ClassName(parameters);
# access an object's field (if the field is public)
$name->fieldName
# call an object's method
$name->methodName(parameters);
```

PHP

```php
$zip = new ZipArchive();
$zip->open("moviefiles.zip");
$zip->extractTo("images/");
$zip->close();
```

PHP

- Il codice d'esempio decomprime un file
- Verifica l'esistenza di una classe con class_exists

Esempio: Recupera un file dal web

```
# create an HTTP request to fetch student.php
$req = new HttpRequest("student.php",
HttpRequest::METH_GET);
$params = array("first_name" => $fname, "last_name"
=> $lname);
$req->addPostFields($params);
# send request and examine result
$req->send();
$http_result_code = $req->getResponseCode(); # 200
means OK
print "$http_result_code\n";
print $req->getResponseBody();
```

PHP

- L'oggetto PHP `HttpRequest` può reperire un file dal web

La sintassi per dichiarare una classe

```php
class ClassName {
   # fields - data inside each object
   public $name; # public field
   private $name; # private field
   # constructor - initializes each object's state
   public function __construct(parameters) {
      statement(s);
      }
   # method - behavior of each object
   public function name(parameters) {
      statements;
   }
}
```

PHP

- All'interno di un costruttore o di un metodo, riferisciti all'oggetto con `$this`

Esempio

```php
<?php
class Point {
    public $x;
    public $y;
    # equivalent of a Java constructor
    public function __construct($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }
    public function distance($p) {
        $dx = $this->x - $p->x;
        $dy = $this->y - $p->y;
        return sqrt($dx * $dx + $dy * $dy);
    }
    # equivalent of Java's toString method
    public function __toString() {
        return "(" . $this->x . ", " . $this->y . ")";
    }
} ?>
```

PHP

Esempio di uso di una classe

```php
<?php
# this code could go into a file named use_point.php
include("Point.php");
$p1 = new Point(0, 0);
$p2 = new Point(4, 3);
print "Distance between $p1 and $p2 is " . $p1->distance($p2) .
"\n\n";
var_dump($p2); # var_dump prints detailed state of an object
?>
```

PHP

```
Distance between (0, 0) and (4, 3) is 5
object(Point)[2]
public 'x' => int 4
public 'y' => int 3
```

output

Ereditierietà di base

```php
class ClassName extends ClassName {
...
}
```

PHP

```php
class Point3D extends Point {
   public $z;
   public function __construct($x, $y, $z) {
      parent::__construct($x, $y);
      $this->z = $z;
   }
   ...
}
```

PHP

- La data classe erediterà tutti i dati e tutti i comportamenti dalla classe ClassName

Metodi statici, campi e costanti

```php
static $name = value; # declaring a static field
const $name = value; # declaring a static constant
```

```php
# declaring a static method
public static function name(parameters) {
   statements;
}
```

```php
ClassName::methodName(parameters); # calling a
static method (outside class)
self::methodName(parameters); # calling a static
method (within class)
```

- Metodi e campi statici sono condivisi all'interno della classe invece che essere replicati per ogni oggetto

Classi astratte ed interfacce

```php
interface InterfaceName {
public function name(parameters);
public function name(parameters);
...
}
class ClassName implements InterfaceName { ...
```

PHP

```php
abstract class ClassName {
   abstract public function name(parameters);
   ...
}
```

PHP

Classi astratte ed interfacce

- Le interfacce sono dei "supertipi" che dichiarano i metodi (e come richiarmarli) senza implementarli
  - Non possono essere istanziate; non possono contenere istruzioni o campi
  - Abilitano il polimorfismo tra sottotipi senza condividerne il codice
- La classi astratte sono come le interfacce, ma potete specificare campi, costruttori e metodi
  - Anche in questo caso non potete istanziarle; abilitano il polimorfismo tra sottotipi consentendo la condivisione del codice

- Sviluppo server side
- Sintassi PHP di base
- Includere codice PHP nelle pagine
- Sintassi PHP avanzata
- **Caso di studio**

Caso di studio: Word of the Day

GRE Vocab Word of the Day

Welcome to Jessica's GRE vocab word of the day page! Each time you visit, a random word and its definition will be shown.

See my word of the day!

This page has been accessed 163 times.

`index.php`

Caso di studio: Word of the Day (cont.)

GRE Vocab Word of the Day

Your word of the day is:

pyrrhic - **adjective**.
costly to the point of negating or outweighing expected benefits

`word.php`

Caso di studio: Word of the Day (cont.)

- Passi dello sviluppo:
  - Scrivere il file index.php senza l'hit counter. All'inizio questa pagina avrà solo codice html (no php), per abbozzare il templete più velocemente
  - Applicare a questo file gli stili CSS di base
  - Aggiungere la funzionalità hit counter usando php
  - Scrivere lo scheletro html della pagina word-of-the-day
  - Aggiungere il codice php alla pagina per scegliere la parola del giorno in modo random da un file di input, migliorare l'efficienza della pagina, etc.

index.php (prima versione)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Word of the Day</title>
    <link href="word.css" type="text/css" rel="stylesheet" />
  </head>

  <body>
    <h1>GRE Vocab Word of the Day</h1>
    <p>Welcome to Jessica's GRE vocab word of the day page!  Each
time
      you visit, a random word and its definition will be
shown.</p>
    <p><a href="word.php">See my word of the day!</a></p>
    <hr />
    <p>This page has been accessed ??? times.</p>
  </body>
</html>
```

PHP

word.css (prima versione)

```css
blockquote {
  font-style: italic;
}

body {
  background-color: white;
  padding: 1em 1em;
  font-family: Garamond, serif;
  font-size: 14pt;
}
```

CSS

Hit counter

index.php

```php
<?php

    $hits = file_get_contents("hits.txt");
    $hits++;
    file_put_contents("hits.txt", $hits);
}
?>
```

PHP

Hit counter (migliorata)

index.php

```php
<?php
$HIT_COUNTER_FILENAME = "hits.txt";

# Reads hit count from disk, increments/writes it, and returns
it.
function hit_counter() {
  global $HIT_COUNTER_FILENAME;
  if (file_exists($HIT_COUNTER_FILENAME)) {
    $hits = file_get_contents($HIT_COUNTER_FILENAME);
  } else {
    $hits = 0;
  }
  $hits++;
  file_put_contents($HIT_COUNTER_FILENAME, $hits);
  return $hits;
}
?>
```

PHP

Word (prima versione)

word.txt

```
prothalamion    noun        a song in celebration of a marriage
atrabilious     adjective   given to or marked by melancholy; GLOOMY
pyrrhic         adjective   costly to the point of negating benefits
```

```php
<!DOCTYPE html>
…
  <body>
    …
    <div>
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" /> <br />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" />
        <img src="vocab.jpg" alt="vocab guy" /> <br />

    </div>
    <p>Your word of the day is:</p>
    ???
  </body>
</html>
```

PHP

word.php

word (migliorata)

Ripetizione della stessa istruzione html: usare i loop php

```
<!DOCTYPE html>
…
  <body>
    …
    <div>
        <?php
      for ($row = 1; $row <= 2; $row++) {
        for ($col = 1; $col <= 6; $col++) {
          ?>
          <img src="vocab.jpg" alt="vocab guy" />
          <?php
        }
        ?>
        <br />
        <?php
      }
      ?>
    </div>

    <p>Your word of the day is:</p>
    ???
  </body>
</html>
```

PHP

word.php

Inclusione file: parti ripetute su più pagine

top.html

```HTML
<!DOCTYPE html>
<html>
  <head>
    <title>Word of the Day</title>
    <link href="word.css" type="text/css" rel="stylesheet" />
  </head>

  <body>
    <h1>GRE Vocab Word of the Day</h1>
```

word.php – reperire da file parole a caso

```php
<?php
include("top.html");
$WORDS_FILENAME = "words.txt";

# reads a random word line from disk and displays its text
function read_random_word() {
  global $WORDS_FILENAME;
  $lines = file($WORDS_FILENAME);
  $random_index = rand(0, count($lines) - 1);
  $random_line = $lines[$random_index];
  $tokens = explode("\t", $random_line);
  list($word, $part, $definition) = $tokens;
  ?>

  <blockquote>
    <p>
      <?= $word ?> -
      <span class="partofspeech"><?= $part ?></span>. <br />
      <?= $definition ?>
    </p>
  </blockquote>

  <?php
}
?>
```

PHP

Modifiche

```
<p>Your word of the day is:</p>
    <?php
    read_random_word();
    ?>
```

word.php

```
.partofspeech {
  font-family: monospace;
  font-weight: bold;
}
```

word.css

File definitivi

- Cartella `ch05-php-files`
  - `hits.txt`
  - `words.txt`
  - `top.html`
  - `index.php`
  - `word.php`
  - `word.css`
- Caricateli su server e provatelo!