

# Laboratorio Progettazione Web

## **Le funzioni in PHP**

Angelica Lo Duca– IIT-CNR  
angelica.loduca@iit.cnr.it  
2012/2013

# Funzioni

- Una funzione è una sequenza di istruzioni che implementano una specifica funzionalità
  - Somma(2,3) – restituisce la somma tra due numeri
- Nascono dall'esigenza di non dover riscrivere il codice ogni volta che abbiamo la necessità di rieseguire quelle operazioni:
  - È sufficiente richiamare l'apposita funzione, fornendole i **parametri** di cui ha bisogno per la sua esecuzione

# Vantaggi

- Meno codice (meno errori, performance migliori)
- Codice più chiaro (manutenzione migliore)
- Riutilizzo del codice

# Definizione di una funzione

- La sintassi generale per definire una funzione è:

```
function nome_funzione ($arg_1,$arg_2,...,$arg_n)
```

```
{
```

```
    // istruzioni con $arg_1,$arg_2,...,$arg_n
```

```
    return valore;
```

```
}
```

- I nomi di funzioni sono **case-insensitive**
  - print, Print, PrInT // sono la stessa funzione
- Una funzione può contenere zero, uno o più **return**
  - L'istruzione **return** cessa l'esecuzione della funzione e restituisce al chiamante un eventuale valore
- Una funzione può ricevere in ingresso zero, uno o più **parametri \$arg**

# Invocazione di una funzione

- Una funzione può essere invocata all'interno di un qualsiasi blocco di codice
- La sintassi per invocare una funzione è  
$$\$valore = nome\_funzione (\$arg\_1, \$arg\_2, \dots, \$arg\_n);$$
- L'invocazione della funzione può avvenire anche prima di averla definita

# Esempio di funzione

```
<?php // somma.php?val1=2&val2=3
```

```
$val1 = $_GET['val1'];
```

```
$val2 = $_GET['val2'];
```

```
echo
```

```
"<h1>$val1+$val2=" . somma($val1, $val2) . "</h1>";
```

```
function somma($add1, $add2) {
```

```
    $addizione = $add1 + $add2;
```

```
    return $addizione;
```

```
}
```

```
?>
```

# Funzioni interne o Built-In

- Sono funzioni già definite dall'ambiente PHP e possono essere utilizzate direttamente dal programmatore.
- Alcune le abbiamo già incontrate:
  - `print("Hello World");`
  - `rand(1,100) // genera un numero casuale da 1 a 100`
  - `date("Y/m/d"); // fornisce la data odierna`

# Funzioni interne o Built-In

- E' ovviamente impossibile conoscere tutte le funzioni a memoria
  - Uso del manuale
- Il miglior manuale a disposizione è presente all'indirizzo **<http://www.php.net>**
- [http://www.php.net/nome\\_funzione](http://www.php.net/nome_funzione) contiene la pagina relativa alla funzione «nome\_funzione»



# Le principali funzioni PHP

- **bool empty ( mixed var )**
  - verifica se la variabile che le passiamo è vuota oppure no. Per 'vuota' si intende che la variabile può contenere una stringa vuota o un valore numerico pari a 0, ma può anche essere non definita o essere impostata al valore NULL. **Restituisce** un valore booleano (vero o falso).
- **bool isset ( mixed var [, mixed var [, ...]])**
  - verifica se la variabile è definita. Una variabile risulta non definita quando non è stata inizializzata o è stata impostata col valore NULL. **Restituisce** un valore booleano.

# Le principali funzioni PHP (cont.)

- **bool is\_null ( mixed var )**
  - verifica se la variabile equivale a NULL. **Restituisce** un valore booleano.
- **void unset ( mixed var [, mixed var [, mixed ...]])**
  - distrugge la variabile specificata. Dopo l'unset(), l'esecuzione di empty() o is\_null() sulla stessa variabile restituirà TRUE, mentre isset() restituirà FALSE. **Non restituisce** valori.

# Restituzione di più di un valore

- Il costrutto *return valore* permette di restituire il valore di una funzione
  - *valore* può essere
    - un valore numerico
    - una stringa
    - una variabile
    - un array
    - ...
- L'array può essere utilizzato quando si vuole restituire più di un valore

# Esempio

- Supponiamo che la funzione somma restituisca oltre alla somma anche il massimo tra i due valori

# Esempio (cont.)

```
<?php // sommaMax.php?val1=2&val2=3

$val1 = $_GET['val1'];
$val2 = $_GET['val2'];
$elaborazione = sommaMax($val1,$val2);
echo "<h1>Elaborazione dei numeri $val1 e $val2</h1>";

echo "<h2>Somma = $elaborazione[somma]</h2>";
echo "<h2>Max      = $elaborazione[max]</h2>";

function sommaMax($num1, $num2) {
    $result['somma'] = $num1 + $num2;
    $result['max']   = ($num1>$num2)?$num1:$num2;
    return $result;
}

?>
```



# Suggerimento

- Non costruire parti di pagine HTML dentro una funzione

```
function somma ($num1, $num2) {  
    $somma = $num1 + $num2;  
    echo "<h2>Somma = $somma </h2>"; ERRATO !!!  
    return $somma;  
}
```

# Visibilità di una variabile

- Una variabile non è visibile in tutte le parti del codice, ma la sua visibilità dipende da dove si trova.
  - **Variabili locali**
  - **Variabili globali**
  - **Variabili statiche**

# Variabili locali

- Sono le variabili definite internamente ad una funzione e sono visibili solo all'interno della funzione

```
function sommaMax($num1, $num2) {  
    $result['somma'] = $num1 + $num2;  
    $result['max']   = ($num1 > $num2) ? $num1 : $num2;  
    return $result;  
}
```

**echo \$result['somma']; // ERRATO !!!**

- \$num1, \$num2 e \$result sono variabili locali alla funzione sommaMax. Non sono visibili fuori dalla funzione.



# Variabili globali

- Sono accessibili da tutte le parti del codice

```
$result = array();
```

```
function sommaMax($num1, $num2) {  
    $result['somma'] = $num1 + $num2;  
    $result['max']    = ($num1 > $num2) ? $num1 : $num2;  
    return $result;  
}
```

```
echo $result['somma']; // CORRETTO !!!
```

- \$num1, \$num2 sono variabili locali alla funzione sommaMax, mentre \$result è una variabile globale, pertanto è accessibile fuori dalla funzione sommaMax.

# Variabili globali (cont.)

- Esiste un altro modo di usare le variabili globali: attraverso la parola chiave **global**.

```
<?php // somma.php?val1=2&val2=3
```

```
    $val1 = $_GET['val1'];
```

```
    $val2 = $_GET['val2'];
```

```
    echo "<h1>$val1+$val2=".somma()."</h1>";
```

```
function somma(){
```

```
    global $val1;
```

```
    global $val2;
```

```
    $addizione = $val1 + $val2;
```

```
    return $addizione;
```

```
}
```

```
?>
```

- \$val1, \$val2 sono variabili globali per cui possono essere utilizzate al di fuori della funzione somma.

# Variabili statiche

- Sono variabili locali ad una funzione, ma non perdono il loro valore quando la funzione cessa la sua esecuzione. Si definiscono usando la parola chiave **static**.

```
function conta()  
{  
    static $conta = 0;  
    $conta++;  
    return $conta;  
}  
echo conta(); // crea $conta, incrementa e ritorna 1  
echo conta(); // incrementa $conta e ritorna 2
```

- La variabile \$conta è statica: viene creata la prima volta che viene chiamata la funzione e viene inizializzata a 0. Quando si richiama la funzione le altre volte, lo statement static \$conta = 0 non viene più eseguito.

# Librerie

- Una libreria è una raccolta di funzioni che implementano delle funzionalità dello stesso tipo.
- Vengono definite dal programmatore per i suoi scopi
  - Connessione ad un database
  - Gestione codice HTML
  - ...

# Librerie (cont.)

- Una libreria viene scritta in un file a parte e poi viene inclusa nello script in cui serve attraverso una direttiva:
  - `include ('mia_libreria.php');`
- Se il file `mia_libreria.php` non si trova nella stessa cartella dello script in cui si include, occorre inserire tutto il percorso (relativo o assoluto) al file.

# Esempio

```
<?php // funzioni_matematiche.php
function somma($a, $b) {...}
function sottrazione($a, $b) {...}
function divisione($a, $b) {...}
?>
```

```
<?php // mio_script.php
include('funzioni_matematiche.php');
echo somma(2,3);
?>
```



# Suggerimento

- Sul Web esistono molte librerie definite da altri programmatori.
  - Si possono riutilizzare
  - Controllare però che effettivamente implementino quello che dicono di implementare !!!

# Esercizio

- Scrivere la libreria **html.php** che contiene le seguenti funzioni:

## **html\_stampa\_header(\$titolo\_pagina)**

prende in ingresso la pagina e costruisce il codice html per la stampa dell'header della pagina.

Restituisce una stringa contenente tutto il codice html. ATTENZIONE: non usare *echo* dentro la funzione!



# Esercizio (cont.)

## **html\_stampa\_footer()**

Restituisce una stringa contenente il footer della pagina (tag di chiusura).

## **html\_stampa\_menu(\$menu)**

Riceve in ingresso un array associativo contenente le voci del menu da stampare e la loro URL associata. Restituisce una stringa contenente il codice HTML da stampare.

# Esercizio (cont.)

- Usare la libreria `html.php` all'interno dello script **`mio_script.php`**
  - Invoca `html_print_header(...)`
  - Invoca `html_print_menu(...)`
  - Stampa una frase a piacere
  - Invoca `html_print_footer()`