

Laboratorio Progettazione Web

Le funzioni in PHP

Andrea Marchetti IIT-CNR

2014/2015

Funzioni

- Il PHP ha molte funzioni già definite (built-in) che possiamo usare

`print()`, `echo()`, `random()`, ...

- Il nome di una funzione è case insensitive

`print()`, `Print()`, `PrInT()` // sono la stessa funzione

- Possiamo definire nuove funzioni

Funzioni

- Sintassi per **definire** una funzione

```
function name() {  
    codice da eseguire  
}
```

- Sintassi per **invocare** una funzione

```
name();
```

Esempio

```
<?PHP
// definizione funzione
function writeMsg(){
    echo ("Hello World!");
}
// invocazione funzione
writeMsg();
?>
```

Esempio

```
<?PHP
function writeMsg() {
    echo ("Hello World!!!");
}

writeMsg(); // invocazione funzione
writeMsg(); // invocazione funzione
?>
```

Esempio

```
<?PHP
/* In PHP Una funzione può essere
invocata anche prima di essere definita
*/
writeMsg(); // invocazione funzione

// definizione funzione
function writeMsg(){
    echo ("Hello World!!!");
}
?>
```

Motivazioni

- Una funzione raccoglie una **sequenza di istruzioni** che svolgono una "funzione" particolare
- Nella programmazione possiamo incontrare più volte la stessa **sequenza di istruzioni**
- Conviene estrarre questa sequenza ed inserirla in una funzione che **invocheremo** invece di riscrivere la stessa sequenza

Effetti

- Programmazione più rapida
 - riuso del codice
- Codice più chiaro
- Manutenzione migliore

Parametri di funzioni

- Informazione può essere passata ad una funzione con gli parametri
- Un parametro è una variabile
- I parametri sono specificati all'interno delle parentesi in qualsiasi numero, separati da virgola

```
function name ($p1, ..., $pn) {  
    codice da eseguire  
}
```

Passaggio dei parametri

- In fase di invocazione della funzione fornisco i parametri reali o argomenti

```
name ( arg1 , ... , argn )
```

- Passaggio per **value** o per reference

Parametri di funzioni

```
<?php
function dichiarazione($nome,$citta,$data,$residenza){
    print("Il sottoscritto $nome nato a $citta il $data e
    residente in $residenza");
}

dichiarazione("Paolo Rossi","Vicenza","25/10/1992","via
Giuseppe Verdi 43, Pisa");

dichiarazione("Vito Bianchi", "Caltanissetta",
"5/03/1996","via XX Settembre 12, Pisa");

?>
```

Valore di default di un parametro

- Possiamo definire dei valori di default per parametri
- Se nella sua definizione imposto dei valori di default, al momento dell'invocazione posso omettere di passare l'argomento

Valore di default di un parametro

```
<?php
function
dichiarazione($nome,$citta,$data,$residenza="Pisa"){
    print("Il sottoscritto $nome nato a $citta il $data e
    residente in $residenza");
}

dichiarazione("Paolo Rossi","Vicenza","25/10/1992");

dichiarazione("Vito Bianchi","Caltanissetta","5/03/1996");

?>
```

Variabili Globali

```
<?php
$residenza = "Pisa";

function dichiarazione($nome,$citta,$data){
    global $residenza;
    print("Il sottoscritto $nome nato a $citta il $data e
    residente in $residenza");
}

dichiarazione("Paolo Rossi","Vicenza","25/10/1992");

dichiarazione("Vito Bianchi","Caltanissetta","5/03/1996");

?>
```

Restituzione di valore

```
<?php
function somma($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . somma(5, 10) . "<br>";
echo "7 + 13 = " . somma(7, 13) . "<br>";
echo "2 + 4 = " . somma(2, 4);

?>
```

Visibilità/utilizzabilità di una variabile

- Parametri (variabili locali)
 - sono visibili solo all'interno della funzione e in generale dove sono definite
- Variabili globali
 - sono visibili in tutto il codice
- Variabili statiche
 - come le variabili locali ma mantengono sempre lo stesso valore

Librerie di funzioni

- Scrivere librerie di funzioni di utilità da riutilizzare in più di un programma
- Utilizzare la funzione built-in `include()` o `require()` o `require_once()`

Librerie di funzioni

- Ad esempio uno potrebbe crearsi una libreria
 - per la connessione ad un DB
 - per gestire il codice HTML
 - in generale per raggruppare funzioni dello stesso tipo e riutilizzabili in script differenti

```
<?PHP
include("dbLibrary.php");

$db      = openDB("localhost","root",NULL,"lpw");
$records = select($db,"SELECT * FROM capitali");

foreach($records as $record)
    echo "<li>".$record['capitale']." è la capitale di
    ".$record['nazione']."</li>";

closeDB($db);
?>
```

```

<?php
// DbLibrary.php

/*****
 * Open a Connection to MySQL *
 *****/
function openDB($servername="localhost", $username="root", $password=NULL, $database="lpw"){
    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $database);

    // Check connection
    if (!$conn) die("Connection failed: " . mysqli_connect_error());
    return $conn;
}

/*****
 * Lettura dei records *
 *****/
function select($conn,$sql){
    $resultSet = mysqli_query($conn, $sql);

    while ($record = mysqli_fetch_assoc($resultSet))
        $records[]=$record;

    // liberazione della memoria impegnata dal result set
    mysqli_free_result($resultSet);

    return $records;
}

/*****
 * Close the Connection to MySQL *
 *****/
function closeDB ($conn){
    mysqli_close($conn);
}

?>

```