

## Approfondimento sul metodo *Get* e il passaggio di dati tramite indirizzo URL

Quando viene richiesta una pagina Web con passaggio di parametri, lo script deve costruire una o più variabili che fanno riferimento al parametro o ai parametri ricevuti.

Il nome e il valore della variabile corrisponderanno al nome e al valore del parametro.

Si procede come nel caso del passaggio di parametri da un modulo secondo la modalità **GET**.

Un esempio di passaggio di parametri è il seguente:

```
http://localhost/ordini.php?codice=5&qta=3
```

Lo script *ordini.php* acquisisce i valori di *codice* e *qta* per mezzo dell'array predefinito **\$\_GET**.

È buona norma salvare i valori contenuti nell'array associativo *\$\_GET* in variabili Php di nome identico a quello dei parametri passati:

```
$codice = $_GET["codice"]      // ---- $codice contiene 5
$qta = $_GET["qta"]           // ---- $qta contiene 3
```

Le variabili *\$codice*, *\$qta*, che contengono i valori dei corrispondenti parametri, possono essere utilizzate all'interno dello script *ordini.php*.

Lo script *ordini.php* attende due parametri. Se, per una qualsiasi ragione, lo script viene richiamato precisando nel suo indirizzo uno solo dei due parametri richiesti come, per esempio, in:

```
http://localhost/ordini.php?codice=5
```

l'interprete Php genera un messaggio di errore del tipo:

```
Notice: Undefined index: qta in c:\... \ordini.php on line 17
```

Il messaggio fa riferimento alla riga dello script (line 17) che contiene l'istruzione:

```
$qta=$_GET["qta"];
```

Per gestire situazioni di questo tipo si fa uso della funzione **isset**, (presentata nel Paragrafo 8), che controlla l'esistenza o meno di una variabile.

Per esempio, per gestire la possibile assenza della variabile *qta*, si può scrivere la seguente istruzione *if*:

```
if ( isset($_GET["qta"]) )
    $qta = $_GET["qta"];
else
    $qta = 1;
```

Nell'esempio la funzione *isset* viene utilizzata per controllare l'esistenza di una variabile dell'array associativo *\$\_GET*, di conseguenza, che il parametro di nome *qta* sia presente nell'indirizzo dello script. Se *isset* restituisce *True* il valore di *qta* è assegnato alla variabile *\$qta*, diversamente a tale variabile viene attribuito un valore di *default*.

Si osservi che il sistema avrebbe visualizzato il medesimo messaggio di errore anche nel caso che lo script *ordini.php* fosse stato richiamato tramite l'URL:

```
http://localhost/ordini.php?codice=5&quantita=3
```

dove, per errore, il secondo parametro è stato indicato con il nome *quantita* al posto di *qta*.

Differente è il caso di parametri ai quali non è stato assegnato alcun valore, trasmessi allo script da un modulo HTML.

Per esempio, se in un form, contenente due caselle di testo *nome* e *cognome* e un pulsante *Invia*, si fa clic sul pulsante senza avere immesso alcun valore nelle caselle di testo, il browser costruisce l'URL:

```
http://127.0.0.1/RispostaRichiesta.php?nome=&cognome=
```

per effetto del quale i parametri *nome* e *cognome* risultano definiti e hanno per valore la stringa nulla. Queste affermazioni possono essere verificate inserendo il comando **var\_dump(\$\_GET)** nello script di risposta. La funzione *var\_dump(\$\_GET)* descrive tipo e valore delle componenti dell'array *\$\_GET*, visualizzando:

```
array(2) { ["nome"]=> string(0) "" ["cognome"]=> string(0) "" }
```

L'output di *var\_dump* mostra che l'array è composto da due variabili:

*\$\_GET["nome"]*, *\$\_GET["cognome"]*, di tipo stringa con zero caratteri, ovvero la stringa nulla "".