

Il modulo di inserimento

Per creare il modulo di inserimento creeremo due file per semplicità:

insert.php sarà il file contenente il form dove inserire i dati. Premendo infine il pulsante "inserisci" si verrà reindirizzati alla pagina insert_data.php che verificherà la correttezza dei dati inseriti, li scriverà nel database e ci riporterà nella pagina precedente.

Iniziamo con il file insert.php:

Si tratta di un semplice form dove inserire i dati. In seguito aggiungeremo parte del sistema per gestire gli errori, ma per il momento iniziamo con il form.

```
<?php
```

```
include './include/page.php';
```

```
    HtmlHeader();
```

```
    Navigation();
```

```
    ?>
```

```
    <br>
```

```
    <br>
```

```
    <form action="insert_data.php" method="POST">
```

```
    Nome: <br>
```

```
    <input type="text" maxlength="20" name="nome" /> <br> <br>
```

```
    Cognome: <br>
```

```
    <input type="text" maxlength="20" name="cognome" /> <br> <br>
```

```
    Numero telefonico: <br>
```

```
    <input type="text" maxlength="15" name="numero" /> <br> <br>
```

```
    <input type="submit" value="inserisci" />
```

```
    </form>
```

```
<?php
```

```
    HtmlFooter();
```

```
    ?>
```

Come vedete, una volta inseriti i dati veniamo indirizzati alla pagina insert_data.php.

In questa pagina inizieremo con il fare dei controlli sui dati appena inseriti. Come prima cosa potremo verificare che nessun campo sia stato lasciato vuoto.

```

    if (empty($_POST['nome']) OR empty($_POST['cognome']) OR
empty($_POST['numero']))
{
    header("Location: insert.php?alert=1");
    die;
}

```

Come vedete, se il campo nome o cognome o numero risultano vuoti, facciamo in modo di venire reindirizzati alla pagina insert.php ed interrompiamo l'esecuzione del codice tramite die in modo da essere certi che non verrà più eseguito nulla. Con il metodo GET passiamo la variabile alert settata a 1.

Questo ci permetterà, nella pagina insert.php, di capire che genere di errore si è verificato (ricordate che vi ho detto che non era finita?).

Ora verificheremo che il nome e il cognome abbiano un formato corretto; diciamo che siano un minimo di tre lettere ed un massimo di 20 e che siano lettere e non simboli o numeri. Costruiamo allora un semplice pattern per un'espressione regolare.

```
$NamePattern = "[a-zA-Z]{3,20}$";
```

Creiamo ora un pattern per il numero di telefono, dove accetteremo un eventuale "+" per il prefisso internazionale ed un numero di cifre da 8 (meno di 8 non è un numero di telefono) a 15.

```
$NumberPattern = "^(\\+)?[0-9]{8,15}$";
```

Ed ora verifichiamo nome, cognome e numero assegnando ad ogni eventuale errore un valore diverso ad alert.

```

if (!ereg($NamePattern, $_POST['nome']))
{
    header("Location: insert.php?alert=2");
    die;
}

```

```

if (!ereg($NamePattern, $_POST['cognome']))
{
    header("Location: insert.php?alert=3");
    die;
}

if (!ereg($NumberPattern, $_POST['numero']))
{
    header("Location: insert.php?alert=4");
    die;
}

```

Notate il punto esclamativo prima di ereg, necessario in quanto è un errore se la condizione NON viene soddisfatta.

Ora se abbiamo superato questa serie di if, i dati sono corretti e possiamo quindi passare ad inserirli nel nostro database. Come prima cosa dobbiamo connetterci, semplicemente includendo il nostro db_connect.php.

```
include "./include/db_connect.php";
```

Dovremo quindi preparare la query. Una query non è altro che una stringa contenente dei comandi che andremo ad eseguire sul database. Per inserire dati, il comando che si utilizza è INSERT INTO ed ha questo formato:

INSERT INTO nome_tabella (campo,campo,campo,....) VALUES (valore, valore,valore,...)

Nel nostro caso per semplicità metteremo la query in una variabile.

```
$sql = "INSERT INTO rubrica_telefonica (nome,cognome,numero) VALUES ('$_POST[nome]','$_POST[cognome]','$_POST[numero]');
```

Notate la sintassi. Gli apici nei quali racchiudo i valori sono obbligatori in quanto si tratta di stringhe.

Ora per eseguire la query sul nostro database, utilizzeremo la funzione `mysql_query()`, nella quale passeremo la query e l'identificativo di connessione. In seguito torneremo a `insert.php` portandoci `alert` settato a 0 (che corrisponderà al messaggio di successo dell'operazione).

```
mysql_query($sql, $conn);  
header("Location: insert.php?alert=0");
```

Il listato completo:

```
<?php
```

```
$NamePattern = "^[a-zA-Z]{3,20}$";  
$NumberPattern = "^(\\+)?[0-9]{8,15}$";  
  
if (empty($_POST['nome']) OR empty($_POST['cognome']) OR  
empty($_POST['numero']))  
{  
    header("Location: insert.php?alert=1");  
    die;  
}  
  
if (!ereg($NamePattern, $_POST['nome']))  
{  
    header("Location: insert.php?alert=2");  
    die;  
}  
  
if (!ereg($NamePattern, $_POST['cognome']))  
{  
    header("Location: insert.php?alert=3");  
    die;  
}  
  
if (!ereg($NumberPattern, $_POST['numero']))  
{  
    header("Location: insert.php?alert=4");  
    die;  
}
```

```
}
```

```
include "./include/db_connect.php";
```

```
$sql = "INSERT INTO rubrica_telefonica (nome,cognome,numero) VALUES  
($_POST[nome],'$_POST[cognome'],'$_POST[numero])";
```

```
mysql_query($sql, $conn);
```

```
header("Location: insert.php?alert=0");
```

```
?>
```

Ora andiamo a rimaneggiare insert.php tenendo conto di quanto ci viene inviato da insert_data.php.

Potremmo inserire qualcosa del genere:

```
$message = "";
```

```
if (isset($_GET['alert']))
```

```
{
```

```
    if ($_GET['alert'] == 0)
```

```
    {
```

```
        $message = "Dati scritti correttamente nel database";
```

```
    }
```

```
    if ($_GET['alert'] == 1)
```

```
    {
```

```
        $message = "ERRORE: Nessun dato scritto nel database.<br>Tutti i  
campi devono essere completati!";
```

```
    }
```

```
    if ($_GET['alert'] == 2)
```

```
    {
```

```
        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo  
nome non nel formato corretto!";
```

```
    }
```

```
    if ($_GET['alert'] == 3)
```

```
    {
```

```

        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo
cognome non nel formato corretto!";
    }

    if ($_GET['alert'] == 4)
    {
        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo
numero telefonico non nel formato corretto!";
    }
}

echo "<br><br>";
echo $message;

```

Come vedete, se alert è settata, scattano i controlli per stabilire che messaggio debba apparire. Ecco il listato completo di insert.php

<?php

```
include './include/page.php';
```

```
HtmlHeader();
```

```
Navigation();
```

```
$message = "";
```

```
if (isset($_GET['alert']))
```

```
{
```

```
    if ($_GET['alert'] == 0)
```

```
    {
```

```
        $message = "Dati scritti correttamente nel database";
```

```
    }
```

```
    if ($_GET['alert'] == 1)
```

```
    {
```

```
        $message = "ERRORE: Nessun dato scritto nel database.<br>Tutti i
campi devono essere completati!";
```

```
    }
```

```
    if ($_GET['alert'] == 2)
```

```
    {
```

```

        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo
nome non nel formato corretto!";
    }

    if ($_GET['alert'] == 3)
    {
        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo
cognome non nel formato corretto!";
    }

    if ($_GET['alert'] == 4)
    {
        $message = "ERRORE: Nessun dato scritto nel database.<br>Campo
numero telefonico non nel formato corretto!";
    }
}

echo "<br><br>";
echo $message;

```

?>

```

<br>
<br>
<form action="insert_data.php" method="POST">
Nome:<br>
<input type="text" maxlength="20" name="nome" /><br><br>
Cognome:<br>
<input type="text" maxlength="20" name="cognome" /><br><br>
Numero telefonico:<br>
<input type="text" maxlength="15" name="numero" /><br><br>
<input type="submit" value="inserisci" />
</form>

```

<?php

```

HtmlFooter();

```

?>

Bene. Ora possiamo iniziare ad inserire un po' di dati.

Concludo con due note:

1. Potrete osservare i dati inseriti nel database da phpmyadmin semplicemente selezionando la tabella e cliccando su "mostra".
2. Osservate che, se viene commesso un errore di inserimento, i dati del form spariscono. Nella nostra situazione è un problema trascurabile in quanto si tratta solo di tre campi, ma immaginate un form molto complesso. Per ovviare a questo problema esistono diverse soluzioni. Se vogliamo utilizzare unicamente PHP, la strada più semplice è quella di inserire i dati in una sessione e farli ricomparire nei campi. altrimenti, tramite javascript, è possibile eseguire i controlli prima che i dati vengano inviati (onSubmit) ed eventualmente interrompere il flusso con un alert.