PHP

Cos'è PHP?

PHP è un linguaggio di scripting open source lato server, che viene utilizzato per sviluppare applicazioni web dinamiche e interattive.

PHP è un acronimo ricorsivo che sta per "PHP: Hypertext Preprocessor".

Ripasso di HTML

Per capire cos'è PHP, dobbiamo parlare prima di HTML.
HTML sta per HyperText Markup Language. È un linguaggio di markup che viene utilizzato per creare pagine web. I tag fondamentali di HTML sono:

`<html>` : Definisce l'inizio e la fine di un documento HTML.
`<head>` : Contiene informazioni di intestazione come il titolo della pagina.
`<title>` : Specifica il titolo della pagina visualizzato nella barra del browser.
`<body>` : Contiene il contenuto principale della pagina visibile agli utenti.
`<h1>` , `<h2>` , `<h3>` , `<h4>` , `<h5>` , `<h6>` : Definiscono i titoli e i sottotitoli della pagina.
`<p>` : Definisce un paragrafo.
`<img>` : Definisce un'immagine.
`<a>` : Definisce un collegamento ipertestuale.

Ripasso di HTML

```
<html>
  <head>
    <title>Titolo della pagina</title>
  </head>
  <body>
    <h1>Titolo</h1>
    <h2>Sottotitolo</h2>
    <p>Paragrafo</p>
    <img src="immagine.jpg" alt="descrizione immagine">
    <a href="https://www.google.com">Google</a>
  </body>
</html>
```

Form

Per richiedere dei dati da una pagina web possiamo utilizzare i form.
HTML mette a disposizione una serie di tag il cui scopo è quello di creare caselle di testo, checkbox e bottoni. Il loro scopo è quello di creare interazione e scambio dati tra gli utenti ed il sito, più tecnicamente è possibile raccogliere dati da un utente e inviarli ad un qualche sistema di elaborazione.

I form da soli permettono la sola realizzazione dell'interfaccia, ma non offrono alcun modo per elaborare i dati inseriti dall'utente. Per fare questo è necessario utilizzare un linguaggio di scripting lato server come PHP.

Form

Per creare un form si utilizza l'omonimo tag `<form>`. Questo tag ha un attributo `action` che indica la pagina a cui inviare i dati inseriti dall'utente. L'attributo `method` indica il metodo HTTP da utilizzare per inviare i dati. I metodi più comuni sono `GET` e `POST`.

```
<form action="processa_dati.php" method="post">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <input type="submit" value="Invia">
</form>
```

Vediamo ora un esempio di form più complesso.

```
<form method="post" action="esegui.php">

  <!-- CASELLE DI TESTO -->
  Nome<br>
  <input type="text" name="nome"><br>
  Cognome<br>
  <input type="text" name="cognome"><br>

  <!-- SELECTBOX -->
  Paese<br>
  <select name="paese">
  <option value="I">Italia</option>
  <option value="E">Estero</option>
  </select><br>


  <!-- TEXTAREA -->
  Commento<br>
  <textarea name="commento" rows="5" cols="30"></textarea>
  <br><br>

  <!-- SUBMIT -->
  <input type="submit" name="invia" value="Invia i dati">

</form>
```

Form

Da questo form possiamo ottenere i seguenti dati:

- nome
- cognome
- paese
- commento

A seconda del metodo utilizzato per inviare i dati, questi saranno disponibili in variabili diverse.
Con il metodo `GET` i dati saranno disponibili nella variabile `$_GET`, mentre con il metodo `POST` saranno disponibili nella variabile `$_POST`. Con `GET` i dati vengono inviati come parametri dell'URL, mentre con `POST` vengono inviati nel corpo della richiesta HTTP (non visibili nell'URL).

Form

Nell'esempio abbiamo ipotizzato l'esistenza di un file `esegui.php` che si occupa di elaborare i dati inviati dal form. Il contenuto della pagina web `esegui.php` cambierà ovviamente ogni volta che l'utente invia dei dati diversi. Per questo motivo, il contenuto della pagina non è statico, ma è generato dinamicamente.

Form

Guardiamo ora come potrebbe essere il contenuto della pagina `esegui.php`:

```php
<?php
  //salviamo i nomi in delle nuove variabili
  $nome = $_POST["nome"];
  $cognome = $_POST["cognome"];
  $paese = $_POST["paese"];
  $commento = $_POST["commento"];

  //stampiamo i dati
  echo "Benvenuto " . $nome . " " . $cognome . "<br>";
  echo "Paese: $paese<br>";
  echo "Commento: $commento<br>";
?>
```

Sintassi PHP

Variabili in PHP:

- Dichiarazione di variabili con il simbolo $.
- Esempio: $nome = "Mario";

Tipi di Dati Comuni:

- Stringhe: Sequenze di caratteri.
- Numeri: Interi o decimali.
- Booleani: true o false.
- Array: Raccolta ordinata di dati.

Sintassi PHP

Operatori in PHP:

- Aritmetici: `+`, `-`, `*`, `/`, `%`.
- Di confronto: `==`, `≠`, `<`, `>`, `⩽`, `⩾`.
- Logici: `&&`, `||`, `!`.
- Di assegnazione: `=`, `+=`, `-=`, `*=`, `/=`.

Strutture di controllo:

- if, else, elseif.

```
if ($voto ⩾ 60) {
    echo "Studente promosso!";
} else {
    echo "Studente bocciato!";
}
```

Connessione a MySQL

Innanzi tutto dobbiamo creare un database MySQL. Per farlo possiamo utilizzare phpMyAdmin, un'applicazione web che ci permette di gestire i nostri database MySQL. Noi useremo XAMPP, che include al suo interno phpMyAdmin e un server web Apache.

Una volta creato il nostro database, possiamo connetterci ad esso utilizzando PHP. Per farlo, dobbiamo utilizzare la funzione `mysqli_connect()`.

```php
$servername = "localhost";
$username = "utente";
$password = "password";
$dbname = "nome_database";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Connessione fallita: " . mysqli_connect_error());
}
echo "Connessione riuscita!";
```

- `localhost` è il nome del server MySQL.
- `utente` è il nome utente per accedere al database.
- `password` è la password per accedere al database.
- `nome_database` è il nome del database a cui connettersi.

Idealmente non dovremmo riscrivere ogni volta i dati di connessione al database. Per questo motivo, è buona norma creare un file `conn.php` che contenga questi dati. In questo modo, se dovessimo cambiare i dati di connessione, dovremmo modificarli solo in un unico file.

```php
// conn.php
<?php
  $servername = "localhost";
  $username = "utente";
  //...
?>
```

```php
// index.php
<?php
  include "conn.php";
  $conn = mysqli_connect($servername, $username, $password, $dbname);
  //...
?>
```

Query SQL

Una query SQL è una richiesta di informazioni al database. Le query SQL vengono eseguite con la funzione `mysqli_query()`.

```
$query = "SELECT * FROM tabella";
$result = mysqli_query($conn, $query);

if ($result) {
    while ($row = mysqli_fetch_assoc($result)) {
        echo "Nome: " . $row["nome"] . "<br>";
    }
} else {
    echo "Errore nella query: " . mysqli_error($conn);
}
```

Si possono fare delle query più semplici, ad esempio trovare il nome utente associato alla mail di un login:

```
$query = "SELECT nome FROM utenti WHERE email = '$email'";
$result = mysqli_query($conn, $query);

if ($result) {
    $row = mysqli_fetch_assoc($result);
    $nome = $row["nome"];
} else {
    echo "Errore nella query: " . mysqli_error($conn);
}
```

Finita la connessione al database, è buona norma chiuderla:

```
mysqli_close($conn);
```

Sessioni

Una volta ottenuto il nome utente, possiamo salvarlo in una variabile di sessione:

```
session_start();
$_SESSION["nome"] = $nome;
```

Quando l'utente effettua il login, possiamo controllare se la variabile di sessione è settata:

```
session_start();
if (isset($_SESSION["nome"])) {
    echo "Benvenuto " . $_SESSION["nome"];
} else {
    echo "Effettua il login";
}
```

In caso di logout, possiamo eliminare la variabile di sessione:

```
session_destroy();
```