

Linguaggio PHP - Scheda N. 01 - Elementi di Base

<p style="text-align: center;">PHP <i>(Hypertext Preprocessor)</i></p> <p style="text-align: center;">Linguaggio di Scripting Intepretato</p> <p style="text-align: center;"><i>Principalmente usato per lo sviluppo di Applicazioni Web Lato Server</i></p>	<p>Script PHP</p> <p>Blocco di istruzioni posto in un determinato punto, all'interno di una pagina web.</p> <p>Alla richiesta della pagina, lo script viene eseguito dal server e, in genere, genera codice HTML che viene inserito al posto dello script stesso.</p> <p>Lo script viene rieseguito ad ogni richiesta, per cui può "personalizzare" la pagina ogni volta in modo diverso, creando delle pagine web dinamiche.</p> <p>Struttura dello Script PHP</p> <pre><?php « istruzione » ; « istruzione » ; ... ?></pre> <p>Estensione ".php"</p> <p>Il file contenente uno o più script PHP deve avere un nome con estensione ".php" (es. <i>index.php</i>), altrimenti lo script non viene eseguito.</p>	<p><i>Esempi:</i></p> <p>La richiesta (Request http) della pagina web "index.php" ...</p> <pre><html> <head></head> <body> <?php echo date('H:i'); ?> </body> </html></pre> <p>... provoca l'esecuzione, sul server, dello script PHP, per cui, sarà inviata al client (Response http) in questa forma:</p> <pre><html> <head></head> <body> 12:35 </body> </html></pre>
<p style="text-align: center;">parte "dinamica" della pagina web</p>		
<p>come generare il codice HTML da sostituire allo script PHP <i>(istruzione echo)</i></p>	<p>echo <stringa-html></p> <p>Scrive nel flusso di output dell'HTTP Response, la <stringa> indicata. Generalmente tale stringa contiene codice HTML che viene quindi "inserito" nella pagina, nel punto in cui si trova lo script.</p> <p>Per delimitare un valore di tipo string, si utilizza il simbolo " " (singolo apice).</p> <p>Per concatenare più stringhe in un'unica stringa utilizzare l'operatore "." (punto).</p> <p>N.B.: nel concatenamento, elementi di tipo non string, vengono automaticamente convertiti in string.</p>	<p><i>Esempi:</i></p> <pre>echo '<p> CIAO </p>'; echo 'Orario:' . date('H:i'); \$Importo = 30; echo 'Prezzo: ' . \$Importo . ' Euro
'; \$Nome = 'Mario'; echo '' . \$Nome . '';</pre>

<p>come usare le Variabili nel linguaggio PHP</p>	<p>I nomi delle variabili iniziano con il simbolo \$. Le variabili non devono essere dichiarate.</p> <p>Esse vengono create al momento del loro primo utilizzo nel codice. Il tipo di una variabile dipende solo dal dato che essa contiene.</p> <p>N.B.: tentare di leggere una variabile prima di assegnarvi un valore, provoca un errore in fase di esecuzione.</p> <p>Regole di Visibilità:</p> <p>Una variabile creata in un sottoprogramma non è visibile all'esterno del sottoprogramma stesso (variabile locale).</p> <p>Una variabile creata in uno script e fuori da sottoprogrammi, è visibile in tutti gli script della pagina (variabile globale).</p> <p>Una variabile globale non si può usare in un sottoprogramma a meno che non se ne dichiari esplicitamente l'utilizzo, all'inizio del sottoprogramma, con la dicitura:</p> <p style="text-align: center;">global <nome-variabile-globale></p> <p>Alcune speciali variabili dell'ambiente PHP, sono direttamente disponibili e sempre visibili (variabili super-globali)</p>	<p><i>Esempi:</i></p> <pre>\$Numero = 10 ; \$Nome = 'Mario Rossi' ;</pre> <hr/> <pre>..... \$Area = (\$Base * \$Altezza) / 2 ;</pre> <hr/> <pre><?php \$Numero = 0 ; // Variabile Globale... ?> <?php // ... è visibile in un altro script ... echo \$Numero ; ?> <?php function ... { // ... è visibile in un sottoprogr. // solo se specificata con "global" ... global \$Numero ; echo \$Numero ; } ?></pre> <hr/> <p>Esempi di variabili super-globali: \$_GET, \$_POST, ecc.</p>
<p>come generare Date e Oriario correnti</p>	<p>date (<stringa-formato>) ➔ string</p> <p>Lettere da usare nella <stringa-formato>:</p> <p>d (giorno), m (mese), Y (anno), h (ora), i (minuti), s (secondi), ecc. (altro)</p> <p>Caratteri diversi da quelli di formato vengono riprodotti senza variazioni.</p>	<p><i>Esempi:</i></p> <pre>date ('d / m / Y') genera '28 / 03 / 2020' date ('h : i . s') genera '18 : 48 . 20'</pre>
<p>come creare una Pagina Web che invia dati al server (form html)</p>	<p>E' necessario utilizzare il tag <form> all'interno del quale si dispongono dei tag <input> fra i quali deve essere presente uno speciale pulsante di tipo submit.</p> <pre><form method="GET" action="calcola.php"> Base: <input type="text" name="txtB">
 Altezza: <input type="text" name="txtH">
 <input type="submit" value="Area Triangolo"> </form></pre> <div data-bbox="1118 1547 1474 1693" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>Base: <input type="text" value="7"/></p> <p>Altezza: <input type="text" value="4"/></p> <p><input type="submit" value="Area Triangolo"/></p> </div> <p>L'utente, con questa pagina, può inserire i dati nelle caselle (tipo text) di nome txtA e txtB. Al click sul pulsante di tipo submit, il browser invia al server una Request HTTP:</p> <ul style="list-style-type: none"> - basata sul metodo specificato nell'attributo method (GET o POST) - che richiede la risorsa specificata nell'attributo action (calcola.php) - che riporta nei parametri i dati inseriti in tutti i tag <input> dotati di nome, secondo il formato application/x-www-form-urlencoded (<nome>=<valore>&<nome>=<valore>&...) <p>Nel nostro esempio, i parametri saranno trasmessi nella forma: txtB=7&txtH=4 Il metodo GET li invia nell'URL, mentre il metodo POST li invia nel corpo della request.</p>	

<p>come Recuperare il Valore dei Parametri da una script PHP eseguito sul Server</p>	<p>Le variabili super-globali <code>\$_GET</code> e <code>\$_POST</code> sono <i>Vettori Associativi</i> che consentono di accedere ai parametri ricevuti dal server nella Request HTTP.</p> <p><code>\$_GET ['<nome-parametro>']</code> ➔ <code><valore-parametro></code> (tipo string) <code>\$_POST ['<nome-parametro>']</code> ➔ <code><valore-parametro></code> (tipo string)</p> <p><code>\$_GET</code> si usa in caso di Request effettuata con il metodo GET e cerca i parametri nell'URL. In tal caso i parametri <i>sono visibili</i> nell'URL richiesto:</p> <p><code>http://itis.it/calcola.php?txtB=7&txtH=4</code></p> <p>Esempio:</p> <pre><code>\$Base = \$_GET ['txtB']</code> ... pone in <code>\$Base</code> il valore '7' recuperato dall'URL ... <code>\$Altezza = \$_GET ['txtH']</code> ... pone in <code>\$Altezza</code> il valore '4' recuperato dall'URL ...</pre> <p><code>\$_POST</code> si usa in caso di Request effettuata con il metodo POST e cerca i parametri nel corpo della Request stessa. In tal caso i parametri <i>non sono visibili</i> nell'URL.</p> <p>Esempio:</p> <pre><code>\$Base = \$_POST ['txtB']</code> ... recupera '7' dal corpo del della request ... <code>\$Altezza = \$_POST ['txtH']</code> ... recupera '4' dal corpo del della request ...</pre>	
<p>come riconoscere se la Request deriva da un click su un pulsante Submit</p>	<p>Una Request ricevuta dal server su una pagina PHP potrebbe essere:</p> <ol style="list-style-type: none"> una semplice "prima navigazione" sulla pagina conseguenza del click su un pulsante Submit <p>Nel primo caso non c'è invio di parametri, quindi per distinguere i due casi basta un semplice if:</p> <pre><code>if (count (\$_GET) == 0)</code> <i>// ... è una prima navigazione ...</i> else <i>// ... è stato cliccato un pulsante submit ...</i></pre> <p>Se nella pagina sono presenti più pulsanti Submit, nei parametri viene inviato solo la coppia nome/valore <i>relativa al pulsante cliccato</i>.</p> <p>Per riconoscere se un pulsante Submit è stato cliccato è quindi sufficiente verificare se fra i parametri esiste il pulsante stesso:</p> <pre><code>if (array_key_exists (\$_GET, 'plsCalcola'))</code> <i>// ... è stato premuto il pulsante plsCalcola ...</i></pre>	<p><i>Esempi:</i></p> <pre><code>if (count (\$_GET) == 0)</code> echo 'benvenuto!'; else { <code>\$Base = \$_GET['txtB'];</code> <code>\$Altezza = \$_GET['txtH'];</code> echo <code>\$Base * \$Altezza</code>; }</pre> <hr/> <pre><code>if (array_key_exists (\$_GET,</code> <code>'plsCalcola'))</code> { <code>\$Base = \$_GET['txtB'];</code> <code>\$Altezza = \$_GET['txtH'];</code> echo <code>\$Base * \$Altezza</code>; }</pre>