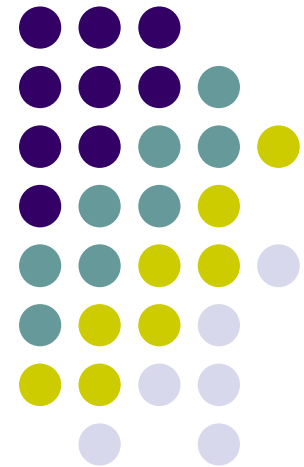


# UD4: Programmazione Server Side

---

## PARTE 2: Costrutti Principali del PHP

Fabio Cantaro





# PHP EMBEDDED

- Il PHP convive con l'HTML: una pagina PHP è spesso costituita da una fusione di HTML e istruzioni PHP (linguaggio Embedded)
- Le istruzioni PHP sono separate dal codice HTML attraverso due delimitatori:
  - tag di apertura: **PHP <?php**
  - tag di chiusura: **?>**
- Ogni istruzione PHP termina con un punto e virgola

```
<?php
    echo "Ciao Mondo!! <BR>";
?>
```



## Php (2)

- Altro metodo usano il **tag breve** `<? e ?>`

```
<?
    echo "Ciao Mondo!! <BR>";
?>
```

- Da usare con cautela: problemi con versioni precedenti alla 4 (dev'essere attivato nel file di configurazione **php.ini**)



## Php (3)

- Tutto il testo esterno ai simboli `<?php` e `?>` è considerato HTML.
  - => viene spedito al client senza modifiche.
- Il testo contenuto all'interno dei simboli `<?php` e `?>`
  - ...è codice PHP e viene gestito dal motore PHP.
- Le pagine vengono interpretate, elaborate, modificate
  - => soltanto successivamente vengono inviate al Browser



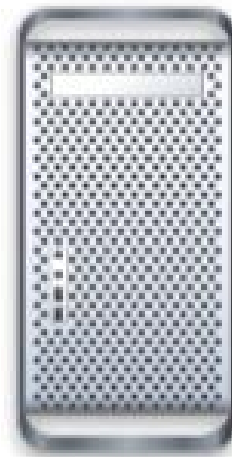
# Php (4)

- Il codice PHP viene eseguito sul server
  - Non viene inviato al client
- Non è possibile visualizzare il codice PHP sul client.
  - E' una prima forma di sicurezza dell'applicativo.
- Il codice PHP può effettuare output di testo.
  - L'output viene inviato al client
  - Al posto del codice PHP.
- L'output deve essere in formato HTML

# Php (5)



```
<I>ABC</I>  
  
<?php  
echo "<BR>"  
?>  
  
testo
```



Apache  
+  
PHP



```
<I>ABC</I>  
  
<BR>  
  
testo
```



# PHP.INI

- File di configurazione dell'interprete PHP
- Viene attivato all'attivazione di Apache
- Alcune righe di esempio di **php.ini**:

- `engine=on` ;abilita il motore PHP come modulo di Apache => FONDAMENTALE
- `short_open_tag` ; abilita l'uso del tag breve `<?...?>`
- `output_buffering=off` ;permette di spedire COOKIES e HEADER negli script PHP



# Esempio

```
<HTML><BODY>  
Oggi &egrave; il giorno  
<?php  
    echo gmdate("M d Y");  
?>  
</BODY><HTML>
```

Viene ricevuto dal browser come:

```
<HTML><BODY>  
Oggi &egrave; il giorno  
Mar 25 2002  
</BODY><HTML>
```





# commenti

- tre modi per inserire un commento:
  - farlo precedere dai caratteri `//`. Solo una riga,
  - caratteri `/*` all'inizio e `*/` alla fine. In questo caso il commento potrà estendersi su più righe;
  - farlo precedere dal carattere `#`. Solo una riga



# Commenti: esempio

```
<?php  
  
// questo è un commento  
echo "Ciao Mondo !";  
echo "Ciao Mondo !"; // altro commento  
  
/* questo è un commento  
un po' più lungo,  
che può stare su più righe */  
echo "Ciao Mondo !";  
  
# stile di commento più raro  
echo "Ciao Mondo !";  
  
?>
```



# echo

- Mostrare a video messaggi
- Il comando che useremo si chiama **echo** :

```
<?php echo("Ciao Mondo !"); ?>
```

- È un costrutto (non una funzione) => può essere usato sia con le virgolette “” che con le parentesi ():

```
<?php
// con le parentesi
echo ("Ciao");
// senza le parentesi...
echo "Ciao";
// ...si possono accodare più frasi
echo "Ciao" , " a tutti." , "Come va?";
?>
```



# OSS

```
<?php
    echo "Prima stringa, adesso andiamo
a capo. <br>";
    echo "Seconda riga .<br>";
    echo "Ricordi l'HTML?";
?>
```

- Abbiamo utilizzato il tag HTML <BR> per andare a capo

=> L'output di una pagina PHP deve quindi essere costituito da HTML



## Esempio 2

```
<?php  
echo "Normale<br>";  
echo "<strong>Grassetto</strong><br>";  
echo "<em>Corsivo</em>";  
?>
```

- Tutto ciò che viene mostrato dal browser viene quindi interpretato come pagina HTML e come tale rapp



# Variabili

- Per ora diciamo solamente che:
  - Una variabile in PHP è costituita da un nome arbitrario preceduto dal simbolo **\$**
  - a una variabile un valore può essere assegnato:
    - direttamente (esempio **\$base = 5**)
    - o a seguito di una operazione, ad esempio una moltiplicazione (esempio **\$area = \$base \* \$altezza** )
  - **Non è necessario inizializzare o dichiarare** una variabile prima di utilizzarla



## Variabili (2)

- Le variabili possono anche contenere stringhe

```
<?php  
$messaggio = "Questa è una stringa";  
echo $messaggio;  
?>
```

- Con le variabili ECHO non usa le virgolette.



# Programmazione Lato Client

- Il PHP ha origine dal linguaggio C => molta sintassi è simile.

OPERATORI di CONFRONTO	
<b>==</b>	uguale
<b>&gt;=</b>	Maggiore o uguale
<b>&lt;=</b>	Minore o uguale
<b>!=</b>	Diverso
<b>&lt;</b>	Minore
<b>&gt;</b>	Maggiore





# Operatori

OPERATORI LOGICI	
<b>!</b> (punto esclamativo)	NOT
<b>  </b> (shift + backslash)	OR
<b>&amp;&amp;</b> (shift + 6)	AND



# Selezione

- Segue le stesse regole del linguaggio C, con i vari blocchi tra parentesi graffe:

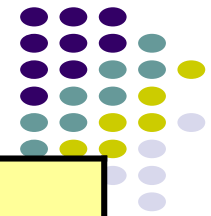
```
If (condizione)
{
    istruzioni
}
elseif (condizione)
{
    istruzioni
}
else
{
    istruzioni
}
```



# Selezione: patenti ed età

```
<?php
$eta=20;
$patente='a';
if (($eta>=18) && ($patente="b"))
{
    echo "patentato";
}
elseif (($eta>=18) && ($patente="a"))
{
    echo "puoi guidare la moto";
}
//se eta minore di 18 e patente non è ne A ne B
else
{
    echo "nessuna patente";
}
?>
```

# Breve introduzione a CICLI e CONTROLLI



WHILE	DO-WHILE	SWITCH
<pre>While(cond) {   istruzioni }</pre>	<pre>Do {   istruzioni }while(condizione);</pre>	<pre>Switch(var) {   case valore:   {     istruzioni;     Break;   }   .....   default:   {     istruzioni;     Break;   } }</pre>



# Esempio1 : maggiore di 3 numeri

(pag.90 - Eser1.php)

- Scopo: funzionamento del server web e del motore php.
- Testo:
  - inserire da form 3 numeri
  - Il form rimanda alla stessa pagina tramite le 3 variabili tramite la variabile globale `$PHP_SELF`
  - Delle 3 variabile stampa a video il Maggiore
- Notare come si interrompe il codice PHP per mettere in mezzo codice HTML



# Esempio 1: soluzione

```
<?php
    echo "<form name=modulo method=post action=".$PHP_SELF.">";
?>
<p>primo numero
    <input name="n1" type="text">
</p>
<p>secondo numero
    <input type="text" name="n2">
</p>
<p>terzo numero
    <input type="text" name="n3">
</p>
<p>
    <input type="submit" name="Submit" value="calcola maggiore e minore">
</p>
</form>
<?php
if (($n1>$n2) && ($n1>$n3))
{
    $mag=$n1;
}
else
{
    if (($n2>$n3) && ($n2>$n1))
    {
        $mag=$n2;
    }
    else
    {
        $mag=$n3;
    }
}
echo "MAggiore: ".$mag;
?>
```



# Esempio 2: tabella di numeri pari

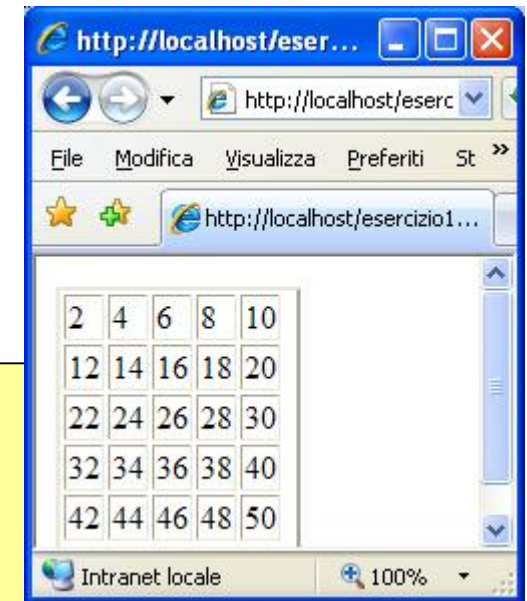
(pag 91, eser2.php)

- Generare una tabella di  $n/2$  numeri divisibili per 2
- Usare un do-while finchè il contatore  $i < n$
- Il resto si ottiene col modulo  $\%$

# soluzione

```
<?php //eser2.php
echo "<TABLE border=2><TR><TD>";
$n=50;
$i=1;
do
{
    if (($i % 2)==0) //% serve per calcolare il resto della divisione
    {
        echo $i."<TD>"; //stampa e passa alla colonna successiva
    }
    if ($i % 10==0) //se decima colonna, salto di riga
    {
        echo "<TR><TD>"; //salta alla riga nuova
    }
    $i=$i+1; //incremento del contatore
} while ($i<=$n); //fino alla riga ennesima

echo "</TABLE>";
?>
```







# Esercizi di applicazione

1. Scrivere una pagina php che stampi il tuo nome e cognome e salvarla in una cartella qualsiasi => modificare i parametri di Apache per farla riconoscere da Easyphp
2. Scrivere una pagina php che stampi una scacchiera di 20x20 caselle
3. Scrivere una pagina php che stampi i primi 100 numeri interi
4. Scrivere una pagina php che stampi i primi 100 numeri interi partendo da un valore immesso dall'utente via form



# Soluzione scacchiera

```
<?php
echo "<table>";
for($i=0;$i<20;$i++)
    {
    echo "<tr>";
    for($j=0;$j<20;$j++)
        if(($j%2)==0)
            {
            echo "<td BGCOLOR='black'
width=10 height=10> </td>";
            }
        else
            {
            echo "<td BGCOLOR='white'
width=10 height=10> </td>";
            }
        echo "</tr>";
    }
echo "</table>";
?>
```