



Desenvolupament de llocs web amb PHP y MySQL

Tema 2: Llenguatge PHP bàsic

- **1.Sintaxis bàsica**
- **2.Tipus de dades**
- **3.Variables**
- **4.Constants**
- **5.Expresions y operadors**
- **6.Estructures de control**
- **7.Funcions**
- **8.Taules**
- **9.Biblioteques de funcions**

Sintaxis bàsica

- **PHP és sensible a les majúscules**
- **Com s'incrusta en la pàgina web?**

<?PHP ... ?>

recomanat, sempre disponible

<?= expressió ?>

equival a <? echo expressió ?>

- **Les instruccions es separen amb un ; como en C. La marca final ?> implica un ;**
- **·Comentari: com en C, /* ... */ (varies línees) y //(una línea)**

/* Comentari de

varies línees */

echo "hola"; /// Comentari de una línea

- **Per imprimir: echo i print**

·echo: mostra una o més cadenes

echo cadena1 [, cadena2...];

echo "Hola món";

echo "Hola ", "món"; //El echo 50% més ràpid

print: mostra una cadena

printcadena;

print "Hola món";

print "Hola " . "món"; // El . produeix una concatenació

Exemple:

<HTML>

<HEAD>

<TITLE>EL meu primer programa en PHP</TITLE>

</HEAD>

```
<BODY>
<?PHP
echo "Hola món";
?>
</BODY>
</HTML>
```

- Us de \n per a generar codi HTML legible
- Primer Sin

Codi PHP

```
echo "<P>Pàrrafo 1</P>";
echo "<P>Pàrrafo 2</P>";
```

Codi HTML

```
<P>Pàrrafo 1</P><P>Pàrrafo 2</P>
```

Sortida

```
Pàrrafo 1

Pàrrafo 2
```

- Us de \n per generar codi HTML legible
- Ara amb \n

Codi PHP

```
echo "<P>Pàrrafo 1</P>\n";  
echo "<P>Pàrrafo 2</P>\n";
```

Codi HTML

```
<P>Pàrrafo 1</P>  
<P>Pàrrafo 2</P>
```

Sortida

```
Pàrrafo 1  
  
Pàrrafo 2
```

- **Exercici 1: programa que mostra un missatge**

Il·lustra com incrustar codi PHP en un document HTML y com imprimir desde PHP

- **Inclusió de fitxers externs:**

–include()

–require()

- Els dos inclouen i avaluen el fitxer especificat
- Diferència: en cas d'error include() produeix un warning i require() un error fatal
- S'utilitzarà require() si al produir-se un error te que interrompirse la carga de la pàgina

- **Exemple:**

```
<HTML>
<HEAD>
<TITLE>Titulo</TITLE>
<?PHP
// Incloure biblioteques de funcions
require("conecta.php");
require("data.php");
require("cadena.php");
require("globals.php");
?>
</HEAD>
<BODY>
<?PHP
include("capsalera.html");
?>
// Codi HTML + PHP
...
<?PHP
include("pie.html");
?>
</BODY>
</HTML>
```

Tipus de dades

- **PHP suporta 8 tipus de dades primitius:**

Tipus escalats: boolean, integer, double, string

Tipus compostos: array, object

Tipus especials: resource, NULL

- **El tipus d'una variable no es sol especificar. Si decideix en temps de execució en funció del context i pot variar**

- **Funcions d'interès:**

La funció `gettype()` retorna el tipus d'una variable

Les funcions *is_type* comprova si una variable és d'un tipus donat:

`is_array()`, `is_bool()`, `is_float()`, `is_integer()`, `is_null()`, `is_numeric()`, `is_object()`, `is_resource()`, `is_scalar()`,

is_string()

La funció `var_dump()` mostra el tipus i el valor d'una variable. És especialment interessant amb els arrays

- **Tipus integer(números sencers)**

-27, -5, 0

- **Tipus double(números reals)**

-1.234, -5.33

- **Tipus boolean o (lògic)**

-Valors: *true*, *false* (insensibles a las majúscules)

-El 0 i la cadena buida tenen valor *false*

- **Tipus string:**

-Les cadenes es tanquen entre cometes simples o dobles:

- 'simples': admet els caràcters d'escape `\` (cometes simples) y `\\`(barra). Les variables **NO** s'expandeixen

- "dobles": admet més caràcters d'escape, com `\n`, `\r`, `\t`, `\\`, `\$`, `\"`. Els nombres de variables **Si** s'expandeixen

- Exemples:

```
$a= 9;
echo "a val".$a;
///mostra a val $a
echo"a val $a";
/// mostra a val 9 y avança una línia
echo"<IMG SRC='logogrif'>";
/// mostra <IMG SRC='logogrif'>
echo"<IMG SRC=\"logogrif\">";
```

/// mostra

–Accés a un caràcter de la cadena:

- La forma és \$inicial= \$nombre{0};

Variables

- Les variables sempre van precedides de un \$
- El nombre és sensible a las majúscules
- Comencen per lletra o subratllat, seguit de lletres, números o subratllat
- Variables predefinides:

\$GLOBALS, \$_SERVER, \$_GET, \$_POST, \$_COOKIES, \$_FILES,
\$_ENV, \$_REQUEST, \$_SESSION

- Àmbit: globals al fitxer (excepte funcions) o locals a una funció
- Exemple:

```
$valor= 5;  
echo"El valor és: " . $valor . " ";  
echo"El valor és: $valor"; // atenció: cometes dobles  
Resultat:  
El valor es: 5 El valor es: 5
```

- Variables variables

–És poden crear nombres de variables dinàmicament

–La variable variable pren el seu nom del valor d'una altre variable prèviament declarada

–Exemple:

```
$a= "hola";  
$$a= "món";  
hecho "$a $hola\n";
```

```
echo "$a ${$a}";
```

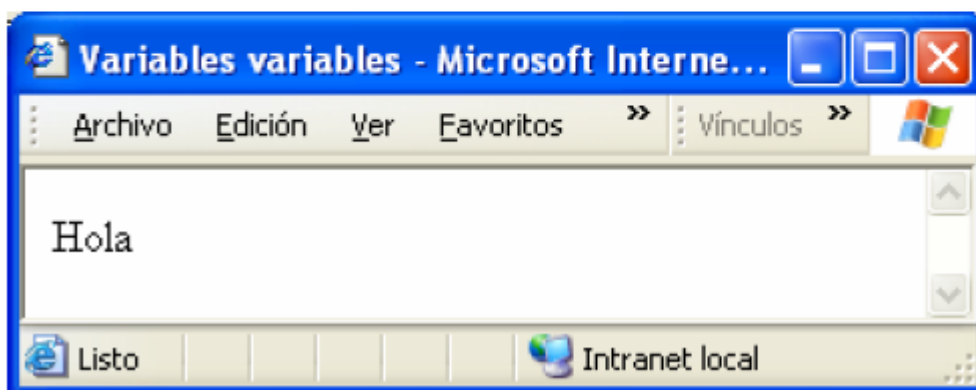
Resultat:

hola món

hola món

- **Exemple de variables variables: pàgina internacionalizada**

```
<?PHP
    $mensaje_es="Hola";
    $mensaje_en="Hello";
    $idioma = "es";
    $mensaje = "mensaje_" . $idioma;
    echo    $$mensaje;
?>
```



- Exemple de variables variables: pàgina internacionalizada (2)

```
<?PHP
    $mensaje_es="Hola";
    $mensaje_en="Hello";
    $idioma = "en";
    $mensaje = "mensaje_" . $idioma;
    echo    $$mensaje;
?>
```



Constants

- Definició de constants:

```
defineix ("CONSTANT", "hola");
echo CONSTANT;
```

- No porten \$ davant
- Només es poden definir constants dels tipus escalars (boolean, integer, double, string)



Expressions y operadors

- **Operadors aritmètics:**

+, -, *, /, %, ++, --

- **Operadors d'assignació:**

=

operadors combinats: .=", +=, etc

\$a= 3; \$a+= 5; a val 8

\$b = "hola "; \$b .= "món"; b val "hola món"

Equival a \$b = \$b . "món";

- **Operadors de comparació:**

==, !=, <, >, <=, >= i altres

- **Operador de control d'error: @.** Anteposat a una expressió, evita qualsevol missatge d'error que pugui ser generat per la expressió

- **Operadors lògics:**

and(&&), or(| |), !, xor

and/&& y or/| | tenen diferents prioritats

- **Operadors de cadena:**

concatenació: . (punt)

assignació amb concatenació: .=



- **Precedència d'operadors (de major a menor):**

++, --
*, /, %
+,-
<, <=, >, >=
==, !=
&&
||
and
or

Estructures de control

- **Estructures selectives:**

–if-else

–switch

- **Estructures repetitives:**

–while

–for

–foreach

- **Estructura selectiva if-else**

```
if (condición)
    sentencia
```

```
if (condición)
    sentencia 1
else
    sentencia 2
```

```
if (condición1)
    sentencia 1
else if (condición2)
    sentencia 2
...
else if (condición n)
    sentencia n
else
    sentencia n+1
```

- **Mateix comportament que en C**
- **Les sentències compostes es tanquen entre claus**
- **Else if pot anar tot junt**
- **Exemple d'estructura selectiva if-else:**

```
<?PHP
if($sexe== 'M')
$salutacio= "Benvinguda, ";
else
$salutacio= "Benvingut, ";
$salutacio= $salutacio. $nom;
Echo "$salutacio";
?>
```



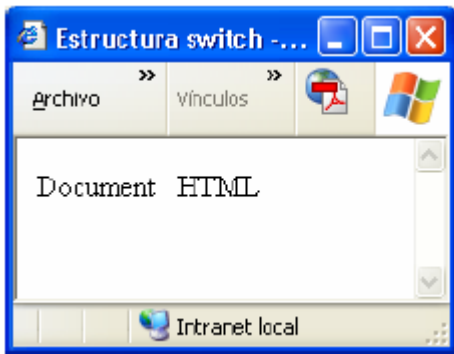
- **Estructura selectiva switch**

```
switch(expressió)
{
case valor_1:
sentència 1
break;
case valor_2:
sentència 2
break;
...
case valor_n:
sentència n
break;
default
sentència n+1
}
```

Mateix comportament que en C, només que l'expressió del case pot ser integer, float o string

- **Exemple d'estructura selectiva switch:**

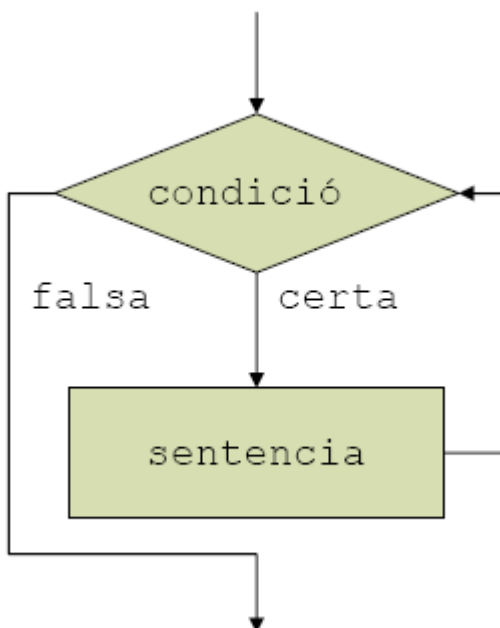
```
switch($extensió)
{
case ("PDF"):
$tipo= "Document Adobe PDF";
break;
case ("TXT"):
$tipus= "Document de texto";
break;
case ("HTML"):
case ("HTM"):
$tipus= "Document HTML";
break;
default:
$tipus= "Arxiu " . $extension;
}
Echo $tipo;
```



- **Estructura repetitiva while**

While (condició)
sentència

- **Mateix comportament que en C**



- **Exemple d'estructura repetitiva while:**

```
<?PHP
Echo "<UL>\n";
$i=1;
while($i<= 5)
{
```

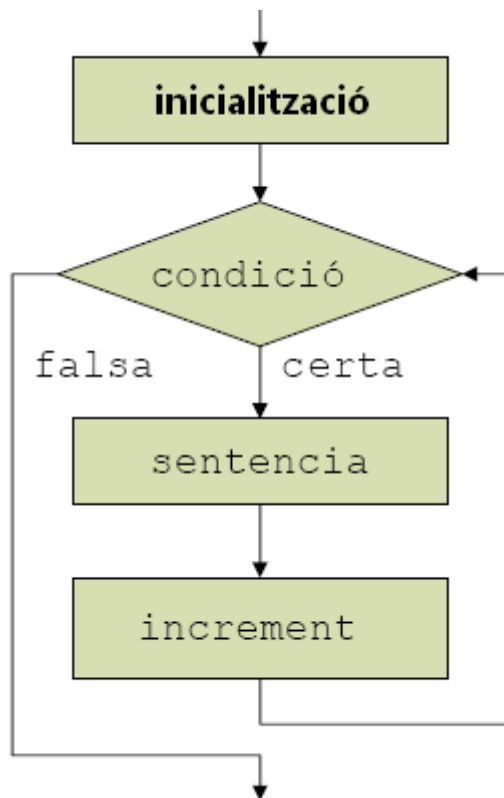
```
Echo "<LI>Element $i</LI>\n";  
$i++;  
}  
Echo "</UL>\n";  
?>
```



- **Estructura repetitiva for**

for(inicialització; condició; increment)
sentència

- **Mateix comportament que en c**



- **Exemple d'estructura repetitiva for:**

```
<?PHP
Echo "<UL>";
for($i=1; $i<=5; $i++)
echo "<LI>Element ".$i."</LI>";
echo "</UL>\n";
?>
```



- **Exercici 2: programa que calcula una taula de multiplicar**

–Il·lustra com manejar variables i com utilitzar bucles

Funcions

- **Exemple:**

```
function suma ($x, $y)
{
  $s= $x+ $y;
  return $s;
}
$a=1;
$b=2;
$c=suma($a, $b);
echo $c;
```

- **Per defecte els paràmetres es passen per valor**

- **Per referència:**

```
function incrementa ($a)
{
  $a= $a+ 1;
}
$a=1;
incrementa ($a);
echo $a; // Mostra un 2
```

- **Arguments per defecte**

```
function muestranombre($titulo= "Sr.")
{
  echo "Estimat $titulo:\n";
}
muestranombre();
muestranombre("Prof.");
```

- **Sortida:**



Estimat Sr.:

Estimada Prof.:

- **Els arguments amb valors per defecte tenen que ser sempre els últims:**

```
function muestranom($nom, $titul="Sr.")  
{  
    echo "Estimat $titulo$nom:\n";  
}  
muestranom("Fernández");  
muestranom("Fernández", "Prof.");
```

- **Sortida:**

Estimat Sr. Fernández:

Estimat Prof. Fernández:

Taules

- **Sintaxis:**

```
array([clau =>] valor, ...)
```

- **La clau és una cadena o un sencer no negatiu. El valor pot ser de qualsevol tipus vàlid en PHP, incluint un altre array**

- **Exemples:**

```
$color= array('vermell'=>101, 'verd'=>51, 'blau'=>255);  
$mesures= array(10, 25, 15);
```

- **Accés:**

```
$color['vermell'] ///No oblidar les cometes  
$medidas[0]
```

- **El primer element és el 0**



- **L'estructura de control foreach permet iterar sobre arrays**

- **Sintaxis:**

```
foreach(expressió_array as $valor)  
sentència  
foreach(expressió_array as $clave=> $valor)  
sentència
```

- **Exemples:**

```
foreach($colors $valor)  
echo "Valor: ".$valor."<BR>\n";  
foreach($colors $clave=> $valor)  
echo "Clau: ".$clave." Valor: ".$valor."<BR>\n";
```

- **Sortida:**

```
Valor: 101  
Valor: 51  
Valor: 255  
Clau: rojo Valor: 101  
Clau: verde Valor: 51  
Clau: azul Valor: 255
```

Biblioteques de funcions

- **Existeixen moltes biblioteques de funcions en PHP**

- **Alguns exemples:**

–Funcions de manipulació de cadenes

–Funcions de data i hora

–Funcions de arrays



–Funcions de fitxers

–Funcions matemàtiques

–Funcions de bases de dades

–Funcions de xarxa

- **Algunes biblioteques requereixen la instal·lació de components addicionals**
- **Totes les funcions de biblioteca estan comentades en la documentació de PHP**
- **Funciones de manipulació de cadenes**

–explode()

- Divideix una cadena en subcadena
- array **explode**(stringseparator, stringstring[, intlmit])

–rtrim(), ltrim(), trim()

- Eliminen caràcters a la dreta, a la esquerra o per els dos costats d'una cadena
- string**rtrim**(stringstr[, stringcharlist])

–strstr()

- Busca la primera ocurrencia d'una subcadena

–strtolower() / strtoupper()

- Converteix una cadena a minúscula / majúscula

–strcmp() / strcasecmp()

- Compara dos cadenes amb/sense distinció de majúscules

–strlen()



- Calcula la longitud d'una cadena

- **Funcions de data i hora**

–date()

- Formateja una data segons un format donat

- Exemple:

```
$data= date ("j/n/Y H:i");  
Echo $data;  
Resultat:  
26/9/2005 17:36
```

–strtotime()

- Converteix una data en un *timestamp* de UNIX

- Exemple:

```
$data= date ("j/n/Y", strtotime("5 april 2001"));  
Echo $fecha;  
Resultat:  
5/4/2001
```

- **Funcions de arrays**

–array_count_values()

- Calcula la freqüència de cada un dels elements d'un array

–array_search()

- Busca un element en un array

–count()

- Compta els elements d'un array



–sort(), rsort()

- Ordena i reindexa un array(r=decreixent)

–ksort(), krsort()

- Ordena per referències clau un array(r=decreixent)

- **Exercici 3: programa que mostra la data actual**

–Il·lustra com utilitzar comentaris, taules i funcions (propies i de biblioteca). També com utilitzar el manual de PHP