

# **PHP 01**

**Variables**  
**Instrucció echo**  
**Estructures**  
**condicionals**

- 1) **Variables**
- 2) Instrucció echo
- 3) Estructures condicionals

- **Variable**: espai de la memòria de l'ordinador (en aquest cas del **servidor**)
  - Li assignem un **contingut** que a grans trets pot ser un **valor**:
    - **Numèric** : només números, amb el seu valor de càlcul.
    - **Alfanumèric** : només text o text amb números, sense valor de càlcul.

- **Variables** en PHP: signe de dòlar (\$) seguit pel nom de la variable.
- Nom **sensible a minúscules i majúscules**.
- Les següents **declaracions** de variables són **diferents** entre elles:  
**\$variable, \$Variable, \$variAble, \$VariAble**

- **Nom de variable vàlid:**
  - **Comença** per:
    - Una lletra
    - Un caràcter de subratllat.
  - **Seguit** de qualsevol nombre de:
    - Lletres.
    - Números.
    - Caràcters de subratllat.
    - **Sense espais.**

- Exemples de noms de variables:

| Noms Vàlids                               | Noms Invàlids             |
|---|---------------------------|
| <code>\$_4variable</code>                 | <code>variable_A</code>   |
| <code>\$Variable4</code>                  | <code>\$4variable</code>  |
| <code>\$Altra</code>                      | <code>\$1_altra</code>    |
| <code>\$Variable_de_nom_molt_llarg</code> | <code>\$A B C</code>      |
| <code>\$ABC</code>                        | <code>\$Variable-A</code> |
| <code>\$A_y_B_x_C</code>                  |                           |

- **Intèrpret PHP** detecta **errada de sintaxi** en declarar una **variable**.
- Si s'ha configurat el fitxer **php.ini** correctament:
  - Si accedim a la pàgina php: apareix un **missatge d'error**.
- **Exemple**: declarem en la línia 20 la variable **\$A B C = 5; //nom invàlid**

***Parse error: syntax error, unexpected identifier "B" in  
/var/www/html/entorns/index.php on line 20***

## Diferències entre els operadors d'assignació i igualtat.

### Assignació (=)

- = en programació **no té el mateix sentit que en matemàtiques.**
- Significa:
  - El valor de l'expressió de la dreta s'estableix a l'operand de l'esquerra.

<?php

***\$A = 3;** // Assignem el valor 3 a la variable \$A*

***\$B = "cadena";** // Assignem el valor "cadena" a la variable \$B*

?>



## Diferències entre els operadors d'assignació i igualtat.

### Operador d'igualtat (==)

- == compara dos valors o variables: no assigna valors.
- Significa:
  - El valor de l'expressió de la dreta **és igual a** valor de l'expressió de l'esquerra?
  - El resultat és un valor **booleà** que s'avalua a **true** o **false**

Exemple operador comparació ==

```
<?php
    $A = 3; // Assignació
    $B = 3; // Assignació
    $A == $B; // Resultat de la comparació: true perquè $A és igual a $A
    $A = 3; // Assignació
    $B = 4; // Assignació
    $A == $B; // Resultat de la comparació: false perquè $A no és igual a $B
?>
```

## Tipus de dades

- En PHP **no és obligatori indicar el tipus de dades** a què pertany una **variable**.
- Els **tipus** de dades són **establerts directament per l'interpret PHP**.
- En qualsevol cas, hi ha 4 tipus bàsics en PHP:
  - **integer**
  - **float o double**
  - **string**
  - **boolean**

- **integer**: enters o sencers, poden tenir diferents valors numèrics sencers que s'expressen amb diferents notacions:

```
$Variable = 18; // Nombre enter positiu
```

```
$Variable = -18; // Nombre enter negatiu
```

```
$Variable = 0x12; // Notació hexadecimal
```

- **float o double**: números de coma flotant (nombres decimals)
  - Tots dos tenen molta precisió, per double és el més precís (permet més decimals).

```
$Variable = 9.876;
```

- **string**: cadena de caràcters, s'expressa amb la següent sintaxi:  
`$Variable = "Cadena de text";`
- **boolean**: tipus lògic, possibles valors: **true** (vertader) o **false** (false).  
`$Variable = true;`  
`$Variable = false;`

# ENTORNS DE DESENVOLUPAMENT

## PHP 01: Variables, echo, condicionals

- 1) Variables
- 2) **Instrucció echo**
- 3) Estructures condicionals

- La instrucció PHP echo:
  - Es diu que serveix perquè es **mostre per pantalla**:
    - Un text.
    - Una variable
    - Una combinació de text i variables.
- **Realment no és així** (encara que podem usar-la per això):
  - Serveix per a **inserir text dins del document HTML subjacent al codi PHP**.

```
echo "eixida per pantalla.";
```

- Insertarà aquest text en el document html.
- Si ho fem entre les etiquetes `<body>` i `</body>` :
  - **Ho veurem per pantalla.**
  - No perquè la instrucció `echo` done lloc al fet que es mostre per pantalla.



```
echo "<h1>eixida per pantalla.</h1>";
```

- No farà que es mostren per pantalla les etiquetes h1 i el text.
- **Introduirà en el document HTML les etiquetes amb el text.**
- Les etiquetes funcionaran com a **codi HTML**.
- El que es mostrarà per pantalla serà **Eixida per pantalla**
  - Amb format de títol **h1** (lletres més grans del normal).

- En general en php són vàlides tant les cometes dobles com les cometes simples.
- Es pot escriure tant:

```
echo "<h1> Eixida per pantalla. </h1>";
```

com:

```
echo '<h1>Eixida per pantalla. </h1> ';
```

- Els parèntesis no són necessaris:
  - **echo** en realitat **no és una funció** sinó una "**construcció del llenguatge**".
  - **Es poden incloure** si es desitja.
  - Exemple:

```
echo ('amb parèntesis i cometes simples');
```

- **Diferència** fonamental entre utilitzar **cometes simples i dobles**:
  - Cometes dobles: permet **introduir variables PHP** definides prèviament:
    - Seran **substitudes pel seu valor corresponent**.
- **Exemple**: substituir el valor de la variable **\$cadena** pel que s'ha indicat:

```
<?php
```

```
$cadena = "Hola";  
echo "$cadena món!";
```

```
?>
```

Hola món

```
<?php
```

```
$cadena = "Hola";  
echo '$cadena món!';
```

```
?>
```

\$cadena món

```
<?php
```

```
echo "eixida per pantalla.";
$A = 5;
echo $A;
echo "El valor de la variable A és $A.";
```

eixida per pantalla.5El valor de la variable A és 5.

```
?>
```

```
<?php
```

```
echo "eixida per pantalla.";
$A = 5;
echo $A; // per mostrar $A com un text enlloc de com una variable:
echo "El valor de la variable \$A és $A."; // \ caràcter d'escapament
```

eixida per pantalla.5El valor de la variable \$A és 5.

```
?>
```

```
<?php
    echo 'amb cometes simples <br>';
    echo ('amb parntesis i cometes simples <br>'); // <br> no es mostra
    echo ( "amb parntesis i cometes dobles <br>"); // és codi HTML
    echo '<h1> Eixida per pantalla </h1> <br>';
    echo "eixida per pantalla.";
    $A = 5;
    echo $A;
    echo "El valor de la variable A s $A.";
?>
```

amb cometes simples  
amb parèntesis i cometes simples  
amb parèntesis i cometes dobles

**Eixida per pantalla**

eixida per pantalla.5El valor de la variable A és 5.

- El navegador rep l'HTML "**equivalent**" generat pel motor PHP del servidor i l'interpreta.
- Dona lloc als resultats que es mostren per pantalla.

```
<body>  
<? php  
echo '<h1> Hola </h1>';  
?>  
</body>
```

Equival a

```
<body>  
<h1> Hola </h1>  
</body>
```

# ENTORNS DE DESENVOLUPAMENT

## PHP 01: Variables, echo, conditionals

- 1) Variables
- 2) Instrucció echo
- 3) **Estructures condicionals**



- En **PHP**: instruccions que permeten prendre **decisions lògiques** quan programem.
  - Són comunes a diferents llenguatges de programació.
  - Se solen anomenar de forma general "**conditionals**".
- 
- `if ... else`
  - `if ... elseif ... else`
  - `switch`

## Condicional if ... else

- Sintaxi:

```
if (expressió_condició) {  
    Sentència1;  
    Sentència2;  
    Sentència3; ...  
} else {  
    Sentència A;  
    Sentència B;  
    Sentència C; ...  
}
```

- Si **expressió\_condició** és certa (s'avalua a **true**)
  - S'executen les sentències 1, 2, 3 ...
- En cas que l'expressió siga falsa (s'avalua a **false**)
  - S'executarien les sentències que estan dins de l'**else**:
    - És a dir, les sentències A , B, C ...

### Exemple if ... else

```
<?php
    $galetes = true;
    if($galetes == true) { // cert
        echo 'Hi ha galetes';
    } else {
        echo 'No hi ha galetes';
    }
?>
```

- En el navegador:

Hi ha galetes

### Exemple if ... else

```
<?php
    $galetes = true;
    if ($galetes == true) {
?>
<br>
<hr>
<?php
    echo 'Hi ha galetes';
    echo '<br>';
    echo '<br>';
    echo '<hr>';
    } else {
        echo 'No hi ha galetes';
    }
?>
```

- Per **introduir html** cal:
  - **O tancar un bloc php** .
  - O utilitzar una **instrucció echo dins del bloc php**.
- La instrucció **echo** significa:  
"donar eixida html al que va a continuació".
- En el navegador:

|               |
|---------------|
|               |
| Hi ha galetes |
|               |

### Exemple if ... else

```
<?php
    $galetes = true;
    if ($galetes == true) {
<br>
<br>
        echo 'Hi ha galetes';
        echo '<br>';
        echo '<br>';
        echo '<br>';
    } else {
        echo 'No hi ha galetes';
    }
?>
```



- En aquest cas, sense tancar el bloc php:
  - Error.
- En el navegador:

**Parse error:** syntax error, unexpected token "<" in /var/www/html/entorns/index.php on line 22

- Sintaxi alternativa: **if**

```
<?php
$a = 5;
if ($a == 5):
?>

<p>A és igual 5</p>

<?php
endif;
?>
```

Exemple amb sintaxi **alternativa**

```
<?php
$a = 5;
if ($a == 5){
?>

<p>A és igual 5</p>

<?php
}
?>
```

Exemple amb sintaxi **tradicional**

- Sintaxi alternativa: **if .. else**

```
<?php
    $a = 4;
    if ($a == 5):
?>

    <p>A és igual 5</p>
<?php
    else:
?>

    <p>A no s igual a 5</p>
<?php
    endif;
?>
```

Exemple amb sintaxi **alternativa**

```
<?php
    $a = 4;
    if ($a == 5){
?>

    <p>A és igual 5</p>
<?php
    } else {
?>

    <p>A no s igual a 5</p>
<?php
    }
?>
```

Exemple amb sintaxi **tradicional**

### Conditionals if ... else if ... else

- Sintaxi:

```
if (expressió_condició1) {  
    Sentència1; Sentència2; ...  
} else if (expressió_condició2) {  
    SentènciaA; SentènciaB; ...  
} else {  
    SentènciaM; SentènciaN; ...  
}
```

- Si **expressió\_condició1** és certa (s'avalua a **true**)
  - S'executen les sentències 1, 2, ...
- Si **expressió\_condició1** és falsa (s'avalua a **false**)
  - Si **expressió\_condició2** és certa (s'avalua a **true**)
    - S'executen les sentències A, B, ...
  - Si **expressió\_condició2** és falsa (s'avalua a **false**)  
(En general, si totes les condicions anteriors són falses)
- S'executen les sentències que estan a l'**else** final:
  - En aquest cas les sentències M, N, ...



### Exemple if ... else if ... else

```
<?php
```

```
$test = 33; // volem comprovar si és més gran que 40 o 35 o 30?
```

```
if ($test > 40) {
```

```
    echo "Sí, $test és més gran que 40.";
```

```
} else if ($test > 35) {
```

```
    echo "Sí, $test és més gran que 35.";
```

```
} else if ($test > 30) {
```

```
    echo "Sí, $test és més gran que 30.";
```

```
} else {
```

```
    echo "No, $test és menor que 40, 35 i 30.";
```

```
}
```

```
?>
```

- En el navegador:

Sí, 33 és més gran que 30.

- Sintaxi alternativa: **if .. else if .. else**

```
<?php
$a = 7;
if ($a == 5):
    echo "<h3>a val 5</h3>";
elseif ($a == 6): // elseif junt!!!
    echo "<h3>a val 6</h3>";
else:
    echo "<h3>a no val ni 6 ni 7</h3>";
endif;
?>
```

Exemple amb sintaxi **alternativa**

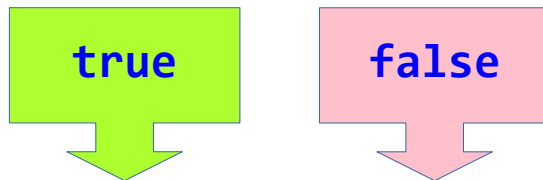
```
<?php
$a = 7;
if ($a == 5){
    echo "<h3>a val 5</h3>";
} else if ($a == 6){ // else if separat!!!
    echo "<h3>a val 6</h3>";
}else{
    echo "<h3>a no val ni 6 ni 7</h3>";
}
?>
```

Exemple amb sintaxi **tradicional**

## Operador ternari

- Sintaxi:

**expressióAmbValorBooleà? Expressió1: Expressió2;**



- Si l'**expressióAmbValorBooleà** és certa s'executa l'**Expressió1**.
- En cas contrari s'executa l'**Expressió2**.
- L'operador ternari pot ser **inserir en sentències d'execució on no es permet la inserció d'if:**
  - Operació d'assignació.
  - Variable de control d'un bucle **for**.

### Exemples operador ternari

- Suposem que tenim quatre variables:  $\$A = 5$ ,  $\$B = 3$ ,  $\$C = -7$  i  $\$D = 5$

| expressió  | resultat   |
|--|--|
| $\$A == \$5 ? \text{dispara} () : \text{espera} ();$                                     | S'executa dispara ()                             |
| $\$A < \$B ? \text{dispara} () : \text{espera} ();$                                      | S'executa espera ()                              |
| $\$B < \$C ? \text{dispara} () : \text{espera} ();$                                      | S'executa espera ()                              |
| $\$A < \$B \ \&\& \ \$B > \$C ? \text{dispara} () : \text{espera} ();$                   | S'executa espera ()                              |
| $(\$A < \$B \ \&\& \ \$B > \$C) \    \ \$B == 3 ? \text{dispara} () : \text{espera} ();$ | S'executa dispara ()                             |
| $\$A == 5 ? \$A = 20 : \$A = 1;$   | Si A valia 5 ara val 20, cas contrari ara val 1. |

**Avantatges i inconvenients**

| <b>Condicional ternari</b>   | <b>if tradicional</b>   |
|--|---|
| <b>Permet l'escriptura compacta, permetent estalviar escriptura de codi.</b>   | <b>Obliga a escriure més sentncies per aconseguir el mateix resultat.</b> |
| <b>Resulta més difícil de llegir, entendre i depurar</b>                       | <b>Resulta ms fàcil de llegir, entendre i depurar</b>                     |
| <b>No tots els programadors el fan servir, alguns ni tan sols el coneixen.</b> | <b>Tots els programadors el fan servir i el coneixen.</b>                 |

**Avantatges i inconvenients**

| Condicional ternari   | if tradicional  |
|---|---|
| S'admet en la sintaxi dels llenguatges en llocs on no s'admet la sentència if | No és vàlid en certes ubicacions on només s'admeten expressions, però pot fer-se l'avaluació abans del punt on siga necessari el condicional. |
| Criticat per alguns experts, adorat per altres                                | Ús i acceptació quasi-universal   |

### Exemple

```
<?php
```

```
$A = 5; $B = 3; $C = -7; $D = 5;
```

```
$msg = '';
```

```
$A == 5? $msg = 'A és 5 <br/>': $msg = 'A no és 5 <br/>';
```

```
echo $msg;
```

```
$A < $B? $msg = 'A menor que B <br/>': $msg = 'A major o igual que B <br/>';
```

```
echo $msg;
```

```
$B < $C? $msg = 'B menor que C <br/>': $msg = 'B major o igual que C <br/>';
```

```
echo $msg;
```

```
?>
```

- En el navegador:

```
A és 5  
A major o igual que B  
B major o igual que C
```