

2.1. Bloc PHP

- Exemple bàsic de codi PHP:

```
<html>
  <head>
    <title>Exemple PHP</title>
  </head>
  <body>
    <?php
      echo 'Hello World!';
    ?>
  </body>
</html>
```

- Qualsevol programa escrit en PHP ha d'estar dins d'un bloc delimitat per:

- Opció 1

```
<?php
    //opció recomanada
?>
```

- Opció 2

```
<?
    ...//no es recomana
?>
```

- Totes les línies de codi de PHP, dins del bloc, acaben amb el símbol ; (punt i coma)
- Per enviar un missatge al navegador del client s'utilitza la funció:
`echo ".....";`

- Podem fer comentaris de 2 maneres:
 - Opció 1:

```
<?php
    //Comentaris d'una línia
?>
```

- Opció 2:

```
<?php
    /*
        Per fer comentaris de bloc
        Totes les línies entre els símbols
        són comentaris
    */
?>
```

2.2. Variables en PHP

- Totes les variables han de començar amb el caràcter \$ seguidament d'una lletra
- PHP és *case sensitive*, és a dir, distingeix entre majúscules i minúscules
- No cal declarar les variables
- Per assignar un valor a una variable sempre s'utilitza l'operador = (igual).
- **Exemple**

```
<?php  
  
$numero =25;  
$Numero =30;  
$cadena = "jaume viladoms";  
  
?>
```

2.3. Tipus de variables

- Tot i no ser declarades, les variables sempre tenen un tipus.
- Els tipus de variable que ofereix PHP són:
 - **Integer** → per representar números enters
 - **Double** → per representar números decimals
 - **Caràcter/String** → per representar caràcters i cadenes de caràcters. Poden anar entre cometes simples i cometes dobles.
 - **Boolean** → qualsevol variable numèrica diferent de 0 o qualsevol cadena no buida es considera TRUE.
 - **Array** → per representar una col·lecció de dades
- **Exemple 1**

```
<?php  
  
$n_enter = 1234;  
$n_decimal =12.34;  
$cadena = "Viladoms";  
$b = TRUE;  
  
?>
```

- Una variable no té un tipus fix, sinó que pot anar canviant el seu tipus segons la última assignació que ha rebut.
- **Exemple 2**

```
<?php
    $n=123;
    echo $n;           //tipus enter
    $n="Viladoms";
    echo $n;           //tipus string
    $n=true;
    echo $n;           //tipus booleà
?>
```

- **Observacions sobre tipus caràcters i strings:**
 - Diferències entre les cometes simples i les cometes dobles, tant en les funcions com en les variables de tipus caràcter/string
 - **Cometes simples**
 - No avaluen l'expressió
 - Mostren per pantalla el contingut íntegre del que hi ha entre cometes.
 - **Cometes dobles**
 - Avaluen l'expressió
 - Si entre cometes dobles hi ha una variable, es mostra el valor de la variable
 - **Exemple 3**

```
<?php
    $n=10;
    echo 'Mostrem el nom de la variable: $n';
    echo "Mostrem el valor de la variable: $n";

    echo "Que mostrem ara?". $n;

?>
```

- Les variables de tipus caràcter i string poden contenir els següents caràcters d'escapament o protecció:
 - `\$` -> per a poder utilitzar/mostrar el caràcter dollar
 - `\"` -> per a poder utilitzar unes cometes dobles de manera literal
 - `'` -> per a poder utilitzar unes cometes simples de manera literal
 - `\\` -> per a poder utilitzar la barra invertida de manera literal
- **Observacions sobre les variables de tipus boolean:**
 - Les variables que contenen números enters o decimals, positius i negatius diferents de 0, es poden considerar variables booleanes amb valor TRUE.
 - Les variables que contenen un valor igual a 0, també són variables booleanes amb valor FALSE.
 - Les cadenes de caràcters són considerades TRUE, excepte si estan inicialitzades a "" o "0" que aleshores són FALSE.

- **Exemple 4**

```
<?php
    $n1=10;           // és true
    $n2=0;            // és false
    $cad1="Viladoms"; // és true
    $cad2="";         // és false

?>
```

2.4. Constants

- Per crear una constant s'utilitza la funció `define(...)`
- Per invocar a una constant només fa falta posar el seu nom.
- Cal destacar que en les constants no s'utilitza el símbol `$`.
- **Exemple 5**

```
<?php  
  
    define("MAX", "10");  
    echo MAX;  
  
?>
```

2.5. Funcions relacionades amb les variables

- **isset(..)**
 - Aquesta funció permet saber si una variable existeix dins el nostre programa.
 - Si la variable existeix retorna true, sinó false.
- **unset(...)**
 - Aquesta funció permet alliberar i destruir una variable.
 - Després de fer un `unset(...)` d'una variable, es considerarà false
- **Exemple 6**

```
<?php  
  
    $v="10";  
    if(isset($v)){  
        echo 'La variable $v existeix';  
    }  
    unset($v);  
    if(isset($v)){  
        echo 'La variable $v existeix';  
    }else{  
        echo 'La variable $v ja no existeix';  
    }  
  
?>
```

- **gettype(...)**
 - Funció que retorna el tipus de la variable que se li passa com a paràmetre.
- **settype(...)**
 - Funció que permet convertir una variable a un altre tipus.
 - Si la funció no es capaç de convertir el tipus, retorna un false.
- **empty(...)**
 - Funció que comprova si una variable està buida, no existeix o el seu valor és 0.
- **is_Integer(nomVar), is_Double(nomVar), is_Array(nomVar), is_String(nomVar)**
 - Permeten saber si una variable és Integer, Double, Array...

2.6. Operadors

- **Operadors d'assignació**
=, +=, -=, *=, /=, .=
- **Operadors aritmètics**
+, -, *, /, %
- **Operadors de comparació**
==, !=, <>, <, >, <=, >=
- **Operadors lògics**
and, &&, or, ||
- **Operador de supressió d'errors**
 - Quan l'execució d'alguna funció de PHP falla, pot aparèixer pel navegador del client un missatge d'error.
 - Per evitar aquesta situació, és recomana utilitzar el caràcter @ (supressió d'errors), que permet no mostrar aquest error per pantalla.
 - **Exemple 7**

```
<?php
//si el fitxer no existeix apareixerà un error per pantalla
$fitxer=fopen("prova.txt", "r");

...

?>
```

```
<?php
//si el fitxer no existeix NO apareixerà cap error per pantalla
$fitxer=@fopen("prova.txt", "r");

...
?>
```

- **Operador de concatenació**

- Per concatenar dues variables des tipus string, o una variable amb un string, s'utilitza el símbol . (punt)
- **Exemple 8**

```
<?php

$v="Jaume Viladoms";
$n=2023;
$v=$v." " . "Cicles formatius" . $n;
echo "Sabadell: " . $v;

?>
```


2.7. Estructures de control

- Sentències condicionals

- Sintaxi if...else

```
<?
    if (condicio){
        ...
    }else [elseif(condicio)]{
        ...
    }
?>
```

- Sintaxi switch(...)

```
<?
    switch(expressió){
        case valor1:...
            break;
        ....
        default:
            ...
            break;
    }
?>
```

- Sentències iteratives

- Sintaxi while(...)

```
<?
    while(condició){
        ....
        //aquí cal actualitzar la condició
    }
```

- Sintaxi do ... while

```
<?
    Do{
        ...
        //Aquí cal actualitzar la condició
    }while(condició);
?>
```

- Sintaxi for

```
<?
    for(expressió inicial; condició de permanència; actualització){
        ...
    };
?>
```

ACTIVITAT 1

Exercici 0

- Fes la instal·lació de la última versió de WampServer en la teva màquina
- Comprova que els documents .php els has de posar en el DocumentRoot, que habitualment en Wamp és la carpeta → `c:\wamp64\www`
- Posa-hi un fitxer bàsic en PHP i comprova que et pots connectar des del navegador web i executar-lo fent → `localhost\fitxer.php`

Exercici 1

Fes un script de PHP amb les següents especificacions:

- Ha de mostrar la diferència entre utilitzar les cometes senzilles i les cometes dobles
- Ha de mostrar per pantalla un text entre cometes dobles i un altre text entre cometes senzilles
- Ha de mostrar per pantalla el caràcter \
- Ha de mostrar la utilitat de la funció `gettype(...)`
- Ha de mostrar la utilitat de la funció `settype(...)`
- Ha de mostrar que una variable que val 0 també es considera FALSE
- Ha de mostrar que una variable que val 100 també es considera TRUE

Exercici 2

Fes un script en PHP que concateni 2 variables de tipus String i les mostri per pantalla.

Exercici 3

Fes un script en PHP que mostri la diferència entre el funcionament de les funcions `isset(...)` i `empty(...)`. Recorda que també pots fer servir la funció `unset(...)`

Exercici 4

Utilitzant PHP, fes un script que solucioni equacions de 2on grau. Només pots utilitzar sentències condicionals

Exercici 5

Utilitzant PHP, fes un script que mostri tots els números parells entre l'1 i el 1000. (Has de fer servir obligatòriament un FOR)

Exercici 6

Utilitzant PHP, fes un script que mostri tots els números primers entre l'1 i el 1000

Exercici 7

Utilitzant PHP, dibuixa per pantalla una taula HTML on hi hagi les taules de multiplicar de l'1 al 10.

Observació: et serà més fàcil fer-les de manera horitzontal que de manera vertical