



Ajuntament
de Barcelona

Desenvolupament web amb PHP Conceptes Generals

IT Academy

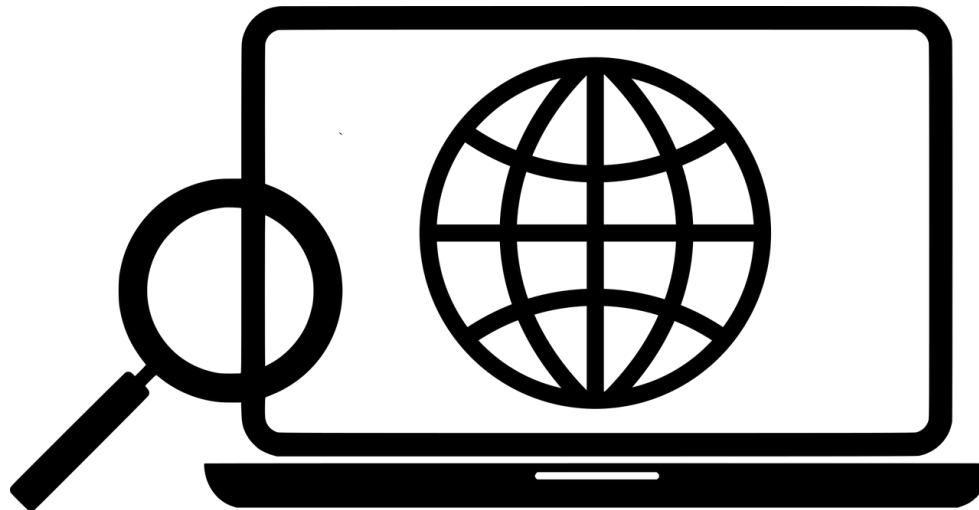
Desembre de 2020





(Molt) Breu història de la programació Web

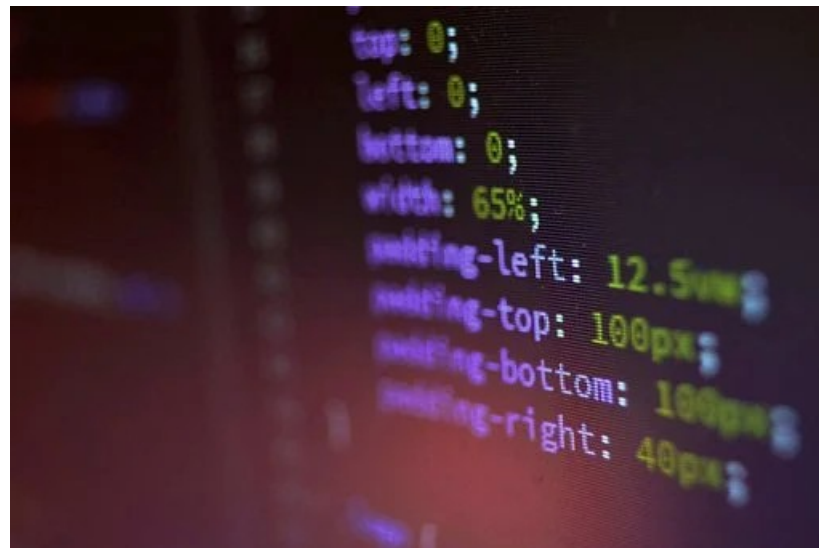
La web va nèixer com a eina de **compartició de continguts** en xarxa. El **contingut era estàtic** i per compondre-lo s'utilitzava(i s'utilitza!) el **llenguatge HTML**.





(Molt) Breu història de la programació Web

Més endavant, es va separar la **part estructurada** de la **part de format/estil**, deixant aquesta última responsabilitat a un nou llenguatge, **CSS**.





(Molt) Breu història de la programació Web

Més endavant encara, els usuaris/es es van adonar que necessitaven **quelcom més que contingut estàtic**. Necessitaven **que la web “pensés”**. Van neixer llavors nombrosos llenguatges de programació per aquest motiu.

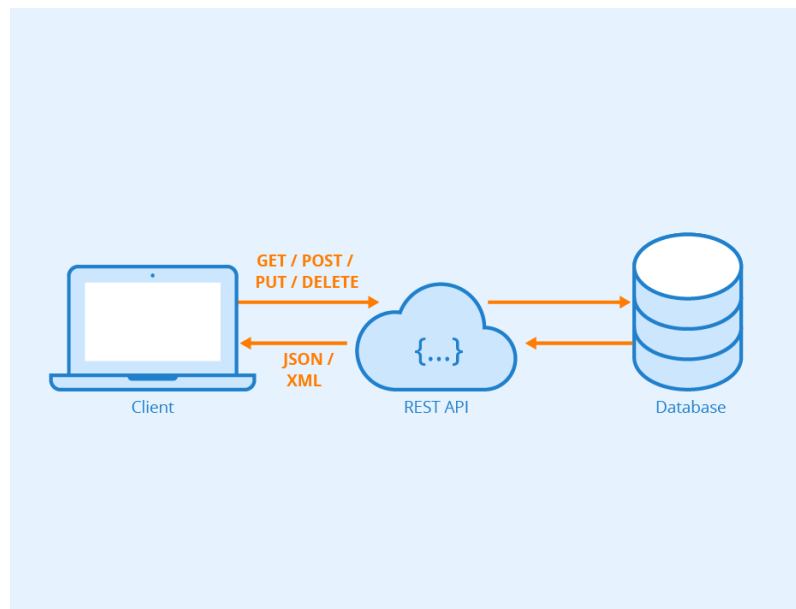


I(creieu-me) un LLARG
etcètera...



(Molt) Breu història de la programació Web

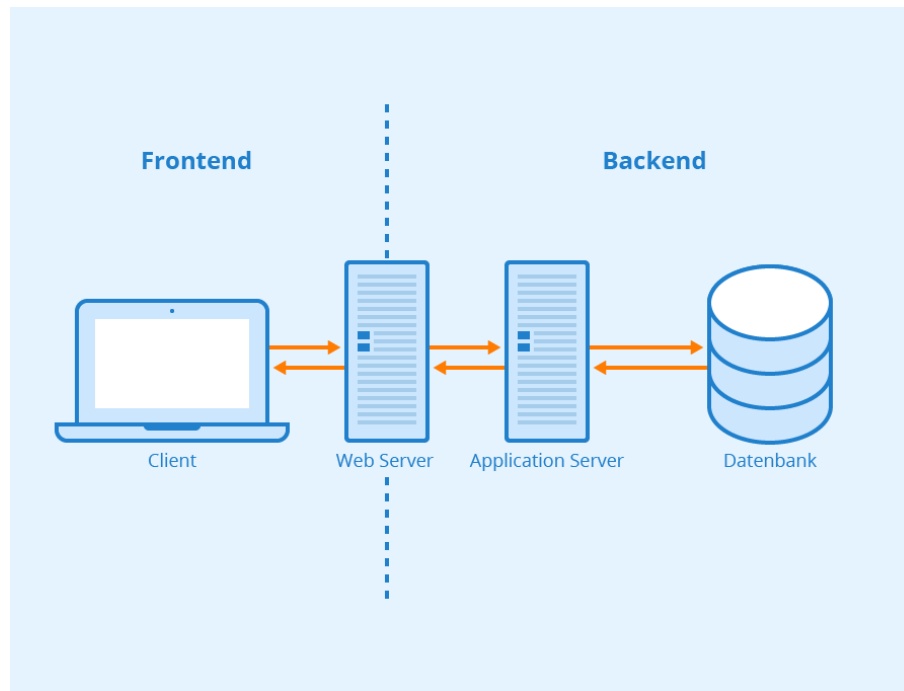
Cada cop hi havia **eines més específiques** per a **parts més concretes** del desenvolupament web, lo qual va propiciar també **arquitectures** específiques e inclús...





(Molt) Breu història de la programació Web

...perfiles de programador/a específics.





(Molt) Breu història de la programació Web

I així fins els nostres dies, on el grau de detall i profunditat de les eines que fem servir farien al·lucinar a qualsevol que hagués conegut la web en els anys 90.





Què és un/a programador/a *Frontend*?

Algú que programa literalment **tot el que ens entra pels ulls** alhora de fer una web. I no només en termes estètics, sino també en tota la vida(ànima,cervell,cor) que fa funcionar aquella aplicació web que veuen els nostres ulls.

Alguns llenguatges emprats serien:

- HTML
- CSS
- Javascript(Typescript)
- .
- .
- .

I les seves implementacions en forma de...

Libraries

- JQuery
- React
-

Frameworks

- Angular
- Vue
-



Què és un/a programador/a *Backend*?

Però **no tot és lo que entra pels ulls**. Sempre **passen coses “*behind the scenes*”**, al darrere. I sovint, aquestes coses són tan complexes i requereixen tantes tasques que es programen a part. És en aquest context on neix aquest perfil.

Alguns llenguatges emprats serien:

- PHP
- C#
- Java
- JS
- .
- .
- .



Frameworks

- Laravel, Symfony(PHP)
- Spring(Java)
- .Net(C#)
- Node(JS)
- .
- .
- .



I de quin perfil som en aquest Itinerari?

En aquest itinerari, tot i que predomina el perfil **backend** per la natura del llenguatge principal que utilitzarem(**PHP**) també treballarem llenguatges bàsics de **frontend** com serien **HTML** i **CSS**.

Frontend



Backend





Què és HTML?

Un dels llenguatges de maquetació web més emprats al món. Defineix l'**estructura** d'una plana web.



```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION
      <nav id="nav" role="navigation">
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="home-events.html">Home Events</a></li>
          <li><a href="multi-col-menu.html">Multiple Column Men
          <li class="has-children"><a href="#" class="current">
            <ul>
              <li><a href="tall-button-header.html">Tall But
              <li><a href="image-logo.html">Image Logo</a></li>
              <li class="active"><a href="tall-logo.html">Ta
            </ul>
          </li>
          <li class="has-children"><a href="#">Carousels</a>
            <ul>
              <li><a href="variable-width-slider.html">Variab
              <li><a href="variable-width-slider.html">Testimon
```



Què és CSS?

Un llenguatge pensat per a definir **format i estils** en una plana web.



Klassen gwd-iframe-vaqg i Code view

```
.gwd-iframe-vaqg {  
  position: absolute;  
  width: 661px;  
  height: 144px;  
  top: 117px;  
  left: 87px;  
}
```

Klassen gwd-iframe-vaqg i Design view

```
.gwd-iframe-vaqg {  
  ✓ position: absolute;  
  ✓ width: 661px;  
  ✓ height: 144px;  
  ✓ top: 117px;  
  ✓ left: 87px;  
  + Add property  
}
```



Què és PHP?

És un llenguatge que va nèixer “incrustat” en codi HTML, **permeten-li aplicar lògiques(“pensar”)**. Amb el temps s’ha anat refinant permetent implementar **lògiques** cada cop més complexes.



```
<?php
    include('db_connection.php');
    include('Tabla.php');
    $db_connection = new db_connection();
    $db_connection->startup("localhost", "root", "", "php_database");
    $db_connection->connect();

    $db_connection->insertarProducto("Chocolate", 3, 1.5);
    $db_connection->insertarProducto("Papel higiènic", 1, 3);

    $productos = $db_connection->getProductos();

    $tabla = new Tabla();
    $tabla->inicializar($productos);
    $tabla->dibujarTabla();

    $db_connection->disconnect();
?>
```



Què és SQL?

És un llenguatge que ens facilita la comunicació i manipulació de dades en una [base de dades](#).

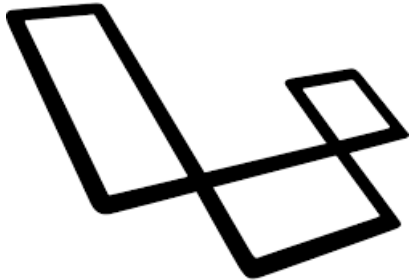


```
- insert into nulltest (cola, colb) values (NULL, NULL)
go
select * from nulltest
select count(*) aantalgelijk from nulltest where cola=colb
select count(*) aantalongelijk from nulltest where cola<>colb
go |
```



I llavors...què és Laravel?

És el **framework** de PHP que hem triat per vosaltres en aquest itinerari. Com a tal, ens facilitarà la creació de MOLTES funcionalitats d'una aplicació web.



```
class UserController extends Controller
{
    public function index(Request $request)
    {
        $this->authorize('view-any', [User::class]);

        $search = $request->search ?? '';
        $users = User::searchLatestPaginated($search);

        return view('users.index', compact('search', 'users'));
    }

    public function create()
    {
        $this->authorize('create', [User::class]);

        return view('users.create');
    }

    public function store(UserStoreRequest $request, CreateUser $create)
    {
        $this->authorize('create', User::class);

        $userData = $request->all();
        $user = $create->execute($userData);

        return redirect()->route('users.edit', compact('user'));
    }
}
```



Peró...

Com bé sabeu, abans de córrer, s'ha de caminar, o sigui que no us espanteu que veurem TOT això(i algo més) pas per pas...





Barcelona
Activa



barcelona.cat/barcelonactiva



Ajuntament
de Barcelona

Desenvolupament web amb PHP

Variables i operadors

IT Academy

Desembre de 2020





Variables: Tipus

```
<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>
```

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource



Variables: Tipus

```
1 <html>
2 <head>
3   <title>Problema</title>
4 </head>
5 <body>
6   <?php
7       $dia = 24; //Se declara una variable de tipo integer.
8       $suelo = 758.43; //Se declara una variable de tipo double.
9       $nombre = "juan"; //Se declara una variable de tipo string.
10      $exite = true; //Se declara una variable boolean.
11      echo "Variable entera:";
12      echo $dia;
13      echo "<br>";
14      echo "Variable double:";
15      echo $suelo;
16      echo "<br>";
17      echo "Variable string:";
18      echo $nombre;
19      echo "<br>";
20      echo "Variable boolean:";
21      echo $exite;
22   ?>
23 </body>
24 </html>
```



Àmbits de variables

PHP té tres àmbits de variables diferents:

- **local**
- global
- estàtic

```
<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}

myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```



Àmbits de variables

PHP té tres àmbits de variables diferents:

- **local**
- global
- estàtic

```
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```



Àmbits de variables

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();
echo $y; // outputs 15
?>
```

PHP té tres àmbits de variables diferents:

- local
- **global**
- estàtic



Àmbits de variables

PHP té tres àmbits de variables diferents:

- local
- **global**
- estàtic

```
<?php
$x = 5;
$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```




Àmbits de variables

```
<!DOCTYPE html>
<html>
<body>

<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}

myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
?>

</body>
</html>
```

0
1
2

PHP té tres àmbits de variables diferents:

- local
- global
- **estàtic**



Variables: Tipus String - Operador “.” (Concatena)

```
2  <html>
3  <head>
4    <title>Problema</title>
5  </head>
6  <body>
7
8    <?php
9        $cadena1="diego";
10       $cadena2="juan";
11       $cadena3="ana";
12       $todo=$cadena1.$cadena2.$cadena3."<br>";
13       echo $todo;
14       $edad1=24;
15       echo $cadena1." tiene $edad1 de edad";
16    ?>
17 </body>
18 </html>
```



Variables: Tipus String - Funcions

```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```

```
<?php
echo str_word_count("Hello world!"); // outputs 2
?>
```

```
<?php
echo strrev("Hello world!"); // outputs !dlrow olleH
?>
```



Variables: Tipus String - Funcions

```
<?php
echo strpos("Hello world!", "world"); // outputs 6
?>
```

```
<?php
echo str_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!
?>
```



Constants

```
define(name, value, case-insensitive)
```

```
<?php  
define("GREETING", "Welcome to W3Schools.com!");  
echo GREETING;  
?>
```

```
<?php  
define("GREETING", "Welcome to W3Schools.com!", true);  
echo greeting;  
?>
```



Constants

```
define(name, value, case-insensitive)
```

```
<?php
define("GREETING", "Welcome to W3Schools.com!");

function myTest() {
    echo GREETING;
}

myTest();
?>
```



Operadors en PHP 7

- Operadors aritmètics
- Operadors d'assignació
- Operadors de comparació
- Operadors d'increment / decreixement
- Operadors lògics
- Operadors de *strings*
- Operadors d'*arrays*
- Operadors d'assignació condicional



Operadors en PHP 7: Aritmètics

Operator	Name	Example
+	Addition	$\$x + \y
-	Subtraction	$\$x - \y
*	Multiplication	$\$x * \y
/	Division	$\$x / \y
%	Modulus	$\$x \% \y
**	Exponentiation	$\$x ** \y



Operadors en PHP 7: Assignació

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus



Operadors en PHP 7: Comparació

Operator	Name	Example
==	Equal	<code>\$x == \$y</code>
===	Identical	<code>\$x === \$y</code>
!=	Not equal	<code>\$x != \$y</code>
<>	Not equal	<code>\$x <> \$y</code>
!==	Not identical	<code>\$x !== \$y</code>
>	Greater than	<code>\$x > \$y</code>
<	Less than	<code>\$x < \$y</code>
>=	Greater than or equal to	<code>\$x >= \$y</code>
<=	Less than or equal to	<code>\$x <= \$y</code>
<=>	Spaceship	<code>\$x <=> \$y</code>



Operadors en PHP 7: Increment/Decreixement

Els operadors d'increment de PHP s'utilitzen per incrementar el valor d'una variable.

Els operadors de decreixement de PHP s'utilitzen per disminuir el valor d'una variable.

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one



Operadors en PHP 7: Increment/Decreixement

Els operadors d'increment de PHP s'utilitzen per incrementar el valor d'una variable.

Els operadors de decreixement de PHP s'utilitzen per disminuir el valor d'una variable.

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one



Operadors en PHP 7: Lògics

Els operadores lògics de PHP s'utilitzen per combinar sentències condicionals.

Operator	Name	Example	Result
and	And	<code>\$x and \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
or	Or	<code>\$x or \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
xor	Xor	<code>\$x xor \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true, but not both
<code>&&</code>	And	<code>\$x && \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
<code> </code>	Or	<code>\$x \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
<code>!</code>	Not	<code>!\$x</code>	True if <code>\$x</code> is not true



Operadors en PHP 7: Strings

PHP té dos operadors que estan especialment dissenyats per a cadenas.

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1



Operadors en PHP 7: Arrays

Els operadors d'*array* de PHP s'utilitzen per comparar *arrays*.

Operator	Name	Example	Result
+	Union	<code>\$x + \$y</code>	Union of <code>\$x</code> and <code>\$y</code>
<code>==</code>	Equality	<code>\$x == \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs
<code>===</code>	Identity	<code>\$x === \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs in the same order and of the same types
<code>!=</code>	Inequality	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code><></code>	Inequality	<code>\$x <> \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Non-identity	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not identical to <code>\$y</code>



Barcelona
Activa



barcelona.cat/barcelonactiva



Ajuntament
de Barcelona

Desenvolupament web amb PHP Arrays

IT Academy

Desembre de 2020





Arrays

“Un ***array***, es un tipus de dades estructurat que permet enmagatzemar un conjunt de dades homogeni, és a dir, on tots els son del mateix tipus i relacionats”



Arrays

Hi ha 3 maneres d'indexar(referir-nos) als diferents elements d'un *array*:

- **Numéricament indexat:** Si ens referim mitjançant un número(que començarà per 0).
- **Associativament indexat:** Si existeix un mapeig clau(paraula)-valor.
- **Multidimensional:** Si a cada índex, hi ha un altre *array*.



Arrays – Numèricament Indexats

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

```
<?php
```

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . " .";
```

```
?>
```



Arrays - Associatius

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
echo "Peter is " . $age['Peter'] . " years old."  
?>
```



Arrays - Associatius

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```



Arrays - Multidimensionals

Name	Stock	Sold
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

```
$cars = array  
(  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```



Arrays - Multidimensionals

```
<!DOCTYPE html>
<html>
<body>

<?php
$cars = array
(
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);
```

```
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";
?>
```

```
</body>
</html>
```

Volvo: In stock: 22, sold: 18.
BMW: In stock: 15, sold: 13.
Saab: In stock: 5, sold: 2.
Land Rover: In stock: 17, sold: 15.



Barcelona
Activa



barcelona.cat/barcelonactiva



Ajuntament
de Barcelona

Desenvolupament web amb PHP Funcions

IT Academy

Desembre de 2020





Funcions

A més de les funcions PHP incorporades, podem crear les nostres funcions. Una funció és un bloc d'instruccions que es pot utilitzar repetidament en un programa.

Sintaxis

```
function functionName () {  
    code to be executed ;  
}
```

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}  
  
writeMsg(); // call the function  
?>
```



“Anatomia” d’una funció

Keyword de la funció, tota funció ha de començar amb aquesta paraula “function”

```
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <?php
    function suma($s1, $s2) {
      return $s1+$s2;
    }
    echo suma(3,2);
  ?>
</body>
</html>
```

Nom : suma

Paràmetres: \$s1, \$s2

Retorn: \$s1 + \$s2

Invocació: suma(3,2);



“Anatomia” d’una funció: Retorn

Una funció pot retornar un valor o no. Si no retorna, la funció seria lo que clàssicament s’anomena **procediment**

```
function suma($s1, $s2) {  
    return $s1+$s2;  
}  
echo suma(3,2);
```



Retorna(ho anuncia precedit per la keyword **return**) en aquest cas, el valor de la suma de las variables \$s1 i \$s2

```
function suma($s1, $s2) {  
    echo $s1+$s2;  
}  
suma(3,2);
```



No retorna res(**procediment**)
La funció mostra directament el resultat per pantalla.



“Anatomia” d’una funció: Paràmetres

Una funció pot tenir 0 o més paràmetres.

```
function hello_world() {  
    echo "Hello World!";  
}
```



Sense paràmetres

```
function hello_world($who) {  
    echo "Hello ".$who;  
}  
  
hello_world("Alumne!");
```



Amb 1 paràmetre

```
function hello_world($name,$surname) {  
    echo "Hello ".$name." ".$surname;  
}  
  
hello_world("Alumne","IT");
```



Amb 2 paràmetres



“Anatomia” d’una funció: Paràmetres

Els paràmetres d’una funció poden tenir **valors per defecte**

```
function hello_world($name, $surname = "IT") {  
    echo "Hello " . $name . " " . $surname;  
}  
  
hello_world("Alumne");
```

S’invoca la funció **sense donar valor al segon paràmetre**.

La pròpia definició de la funció, defineix un valor **per defecte** en el paràmetre **\$surname**



“Anatomia” d’una funció: Nom e invocació

S’invoca de la mateixa manera que s’anomena la funció i amb els mateixos paràmetres, tret que n’hi hagi d’opcionals.

```
function suma($s1, $s2)
```

```
suma(3,2)
```




Funcions: Exemples

```
42 <html>
43 <head>
44 <title>Problema</title>
45 </head>
46 <body>
47 <?php
48     function cuadradocubo($valor,&$cuad,&$cub)
49     {
50         $cuad=$valor*$valor;
51         $cub=$valor*$valor*$valor;
52     }
53     cuadradocubo(2,$c1,$c2);
54     echo "El cuadrado de 2 es:".$c1."<br>";
55     echo "El cubo de 2 es:".$c2;
56 ?>
57 </body>
58 </html>
```



Funcions: Exemples

```
23 <html>
24 <head>
25   <title>Problema</title>
26 </head>
27 <body>
28   <?php
29       function retornarpromedio($valor1,$valor2)
30       {
31           $pro=$valor1/$valor2;
32           return $pro;
33       }
34       $v1=100;
35       $v2=50;
36       $p=retornarpromedio ($v1,$v2) ;
37       echo $p;
38   ?>
39 </body>
40 </html>
```



Barcelona
Activa



barcelona.cat/barcelonactiva



Ajuntament
de Barcelona

Desenvolupament web amb PHP

Estructures de control de fluxe de dades

IT Academy

Desembre de 2020



Estructures de control de fluxe de dades

Els programes que desenvolupem no tenen perquè ser senzillament un arranjament de línies de codi font que s'executen una darrera d'altra. A vegades és necessari expressar condicions booleanes(verdader o fals) per decidir quines línies de codi executar,o, senzillament executar una determinada lògica de forma reiterada. És per això que tenim estructures de control de fluxe de dades.

Condicionals

- If
- else
- else if
- switch

Bucles

- while
- for
- do while



Estructures condicionals: if

Fem una pregunta composta per una expressió booleana i en cas de ser cert s'executarà el codi font allotjat dins de la sentència if

Preguntem si l'edat és major o igual a 18

```
13  <?php
14
15  $edat = 19;
16  if($edat >= 18) {
17      echo "És major d'edat";
18  }
```

Si, en efecte, la variable edat conté un valor superior o igual a 18, s'executarà la línia de codi font 16



Estructures condicionals: else

En cas de que la condició resulti en **false** , podem també definir altra lògica gràcies a la sentència **else**

```
13  -> <?php
14  -> $edad = 15;
15  -> if($edad >= 18) {
16  ->     echo "És major d'edat";
17  -> }
18  -> else {
19  ->     echo "És menor d'edat";
20  -> }
21  -> ?>
```

En aquest cas, al ser el valor 15, s'executarà la lògica de la línia 19. Si no estigués definida la sentència **else**, senzillament no s'executaria res, en aquest cas, al evaluar-se la condició



Estructures condicionals: else if

Si hi ha més casos que volem tenir en compte per a evaluar una condició concreta, podem fer servir **else if**

```
<?php
    $edad = "";
    if($edad >= 18) {
        echo "És major d'edat";
    }
    else if(!is_numeric($edad)) {
        echo "No és un número!";
    }
    else {
        echo "És menor d'edat";
    }
?>
```

En aquest cas, també comprovem que no sigui un número. En cas de ser així, es mostrarà el missatge:

"No és un número"



Estructures condicionals: switch

Una altra manera d'establir condicionals en l'execució seria mitjançant la sentència **switch**

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

S'evalua el valor de la variable **\$favcolor**, i, en funció d'aquesta evaluació s'efectuarà només el cas que coincideixi.

Important la instrucció **break** per no continuar executant el condicional quan ja s'ha executat un dels casos.

El cas **default** és equivalent a **else**.



Estructures de repetició: While

En ocasions ens interessa repetir determinades instruccions i per això tenim els **bucles** o **estructures de repetició**

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

Aquesta és la condició que comprovem **a cada iteració del bucle**

Aquí per exemple, escrivim un missatge amb número de 1 al 5 que es calcula a cada iteració del bucle

A cada iteració del bucle, executem la lògica especificada dins del mateix

IMPORTANT: Hem d'establir **SEMPRE** la lògica que trencarà el bucle, o sino, deixarem l'execució del programa penjada en un **bucle infinit**. Aquí, per exemple, la línia **\$x++** s'encarrega de que, després de 5 iteracions, el bucle es trenqui



Estructures de repetició: for

Tenim altres estructures de repetició, com aquí amb el bucle **for**

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

Aquí es defineixen més elements de bucle entre parèntesis deixant només la lògica a executar dins del bucle



Estructures de repetició: do-while

Menys comú, però existent al cap i a la fi

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

La principal diferència respecte a les altres estructures de repetició és que aquí, la comprovació de la condició que trenca el bucle es fa **al final** d'una iteració



Barcelona
Activa



barcelona.cat/barcelonactiva



Ajuntament
de Barcelona

Desenvolupament web amb PHP Reportant Errors

IT Academy

Desembre de 2020





Reportant Errors

Quan es produeix un error en l'execució d'un programa en PHP, aquest pot arribar a provocar l'aturada del programa. A més, l'error pot mostrarse en pantalla amb més o menys grau de detall.

Parse error: syntax error, unexpected '}', expecting ';' or ',' in C:\xampp\htdocs\ItAcademySamples\index.php on line 23

Lo més normal en entorns de desenvolupament es que ens mostri explícitament l'error al executar el programa tal i com es veu a la imatge però...



Reportant Errors

...què fem si això ens passa en un servidor extern i no podem veure els errors per pantalla?

Doncs que habilitem la mostra d'errors explicitant-ho al mateix codi font.

```
<?php
    ini_set('display_errors', 0);
    ini_set('display_startup_errors', 0);
    error_reporting(E_ALL);
```

Existeixen més nivells d'error(E_ALL,E_ERROR,E_CORE_ERROR...) però amb aquest paràmetre es mostraran tots



Barcelona
Activa



barcelona.cat/barcelonactiva