

PHP Basics

An introduction to php



Què veurem?

- Introducció de PHP
- Practicar amb exercicis

Introducció

- És un llenguatge a nivell de servidor
- El fem servir per generar contingut dinàmic
- Orientat a objectes
- El codi s'executa de forma seqüencial
- Sintaxis molt semblant al C++
- Podem executar PHP des de la nostra consola o bé en un servidor web

Sintaxis

- Tot el codi PHP ha d'anar encapsulat dins els tags `<?php ?>`
- Cada declaració ha d'acabar amb un `;`

Comentaris

- Tot el codi PHP ha d'anar encapsulat dins els tags `<?php ?>`
- Cada declaració ha d'acabar amb un `;`

```
<?php

// Single line comment
# Another single line comment

/**
 * Multi
 * line
 * comment
 */
```

Print data

- Tenim diverses funcions que ens permeten *printar* informació
- La més utilitzada és la funció **echo**

```
<?php
```

```
echo 'Hello world!'; // Prints Hello world!
```

Debug

Per poder *debugar* una variable podem fer servir la funció **var_dump**:

```
<?php  
  
var_dump('Hello world!');
```

Variables

Escalars	Compostes	Especials
boolean integer float string	array object	null

Boolean

Poden tenir dos valors: **true** o **false**

```
<?php  
  
$true = true;  
$false = false  
;
```

Integer

- Qualsevol nombre enter $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
- Es poden especificar en base 10 (decimal), en base 16 (hexadecimal), en octal o en binari
- Podem forçar el *cast* d'una variable a un valor enter amb la funció **intval()** o directament amb els casts **(int)** o **(integer)**

Float

Qualsevol nombre decimal

```
<?php  
  
$a = 1.234; // 1.234  
$b = 1.2e3; // 1200  
$c = 7E-3; // 0.007
```

String

Conjunt de caràcters

```
<?php  
  
$name = 'Jaume';  
$singleQuote = 'Hello Jaume';  
$doubleQuote = "Hello $name"; // Hello Jaume
```

Arrays

- Un array és un conjunt de valors ordenats
- Cada valor té associat una clau
- PHP especifica una clau per defecte per a cada valor.
- No necessitem demanar memòria per crear l'array

```
<?php
```

```
$names = [  
    'Jaume',  
    'Marc',  
];
```

```
$users = [  
    [  
        'name' => 'Jaume',  
        'age' => 20  
    ],  
    [  
        'name' => 'Marc',  
        'age' => 19  
    ],  
]
```

```
echo $users[0]['name']; //Prints Jaume
```

Array - implode

```
<?php  
  
$names = ['Jaume', 'Marc'];  
  
echo implode(', ', $names); // Prints 'Jaume,Marc'
```

Arrays - explode

```
<?php

$ingredients = "tomato, cheese, meat";

explode(',', $ingredients); // ['tomato', 'cheese', 'meat']
```


Arrays - unset

Per eliminar un element del array (no cambia les claus internes)

```
<?php  
  
$names = ['Jaume', 'Marc'];  
  
unset($names[1]); // ['Jaume']
```

Altres funcions interessants...

Podeu consultar totes les funcions disponibles al següent [l·listat](#).

Objectes

- Agrupen dades + comportament
- Els objectes són instàncies d'una classe
- Les classes defineixen propietats, mètodes i el comportament dels objectes

```
<?php

class User
{
    private $name;

    public function __construct(string $name)
    {
        $this->name = $name;
    }

    public function getName(): string
    {
        return $this->name;
    }
}

$user = new User('Jaume');

$user->name; // Error

echo $user->getName(); // Prints Jaume
```

Null

Representa una variable sense valor.

Declaració de variables

Per declarar una variable en PHP hem de fer servir el símbol `$` seguit del nom de la variable.

```
<?php  
  
$name = 'Jaume';  
  
echo $name; // Prints Jaume
```

Superglobals

- Variables predefinides.
- Algunes variables només són disponibles en un cert context.

Superglobals
<code>\$_SERVER</code>
<code>\$_GET, \$_POST</code>
<code>\$_SESSION, \$_COOKIE</code>

Mètodes predefinit

Method	Action
isset	Return true if the variable exist
is_null	Return true if the variable is null
...	

Constants

- El seu valor no pot variar en temps d'execució
- Case sensitive
- Per convenció els noms sempre en *upper case*

```
<?php  
  
define('MIN', 0);  
  
echo MIN; // Prints 0
```

Magic constants

- Hi han 8 constants màgiques que prenen un valor diferent segons on han sigut utilitzades.

Comparacions

<code>\$a == \$b</code>	Equal
<code>\$a === \$b</code>	Identical
<code>\$a != \$b</code>	Not equal
<code>\$a <> \$b</code>	Not equal
<code>\$a !== \$b</code>	Not identical
<code>\$a < \$b</code>	Less than
<code>\$a > \$b</code>	Greater than
<code>\$a <= \$b</code>	Less than or equal to
<code>\$a >= \$b</code>	Greater than or equal to

Estructuras de control

if	else	elseif	while	do-while
for	foreach	break	continue	switch
return	require	include	require_once	include_once

Funcions (1/4)

- Ens permeten organitzar i reutilitzar el nostre codi
- No cal declarar una funció abans de fer-la servir (ha d'estar en el mateix script php)

```
<?php

function sayHello()
{
    echo "Hello!";
}

sayHello() // Prints 'Hello!'
```

Funcions (2/4)

Podem passar paràmetres a les funcions per valor o per referència.

```
<?php

function sayHello($name)
{
    echo "Hello $name!";
}

function sayHelloByReference(&$name)
{
    $name = 'Other';
    echo "Hello $name!";
}

$name = 'Jaume';

sayHello($name); // Prints 'Hello Jaume!'
sayHelloByReference($name); // Prints 'Hello Other!'
sayHello($name); // Prints 'Hello Other!'
```

Funcions (3/4)

Podem especificar valors per defecte.

```
<?php

function sayHello($name = 'Jaume')
{
    echo "Hello $name!";
}

sayHello(); // Prints 'Hello Jaume!'
```

Funcions (4/4)

Podem fer servir la paraula reservada **return** per fer que la funció retorni un valor.

```
<?php

function sum(int $a, int $b): int
{
    return $a + $b;
}

echo sum(1, 1); // Prints 2
```


Hands on

Implementar els [exercicis](#) 1 - 5.

In the next episode

- Formularis web
- Creació i manipulació de fitxers