

Funcions PHP

- **Què són les funcions?**

- Una funció és un conjunt d'instruccions que es poden utilitzar repetidament en un programa.
- És per això que aquest conjunt d'instruccions s'agrupen en una funció.
- Una funció s'executarà mitjançant una crida a la funció.
- PHP té les seves funcions integrades (predefinides) i, a més, es poden crear funcions personalitzades.

Funcions PHP

- **Com es crea una funció?**

Una declaració de funció definida per l'usuari comença amb la paraula clau **function**, seguida del nom de la funció:

```
function myMessage () {  
    echo "Hello world!";  
}
```

Nota: el nom d'una funció ha de començar amb una lletra o un guió baix. Els noms de les funcions NO distingeixen entre majúscules i minúscules.

Consell: Donar un nom a la funció que reflecteixi el que fa la funció!

Funcions PHP

- **Com es crida una funció?**

- Per cridar la funció, només cal escriure el seu nom seguit de parèntesis ():

```
function myMessage () {  
    echo "Hello world!";  
}  
myMessage ();
```

- En l'exemple, es crea una funció anomenada myMessage().
- La clau d'obertura { indica l'inici del codi de funció, i la clau de tancament } indica el final de la funció.
- La funció mostra "Hello World!".

Funcions PHP

- **Arguments de la funció PHP (I)**

- La informació es pot passar a les funcions mitjançant arguments. Un argument és com una variable.
- Els arguments s'especifiquen després del nom de la funció, dins dels parèntesis.
- Es poden afegir tants arguments com es vulgui, s'han de separar amb una coma.
- L'exemple següent té una funció amb un argument (\$name). Quan es crida la funció, també es passa un nom, p. ex. ("Jani"), i el nom s'utilitza dins de la funció, que genera diversos noms diferents, però un cognom igual:

```
function familyName($name) {  
    echo "$name Refsnes.<br>";  
}  
  
familyName("Jani");  
familyName("Hege");  
familyName("Stale");
```

Funcions PHP

- **Arguments de la funció PHP (II)**

- L'exemple següent té una funció amb dos arguments (\$name, \$year):

```
function familyName($name, $year) {  
    echo "$name Refsnes. Born in $year <br>";  
}  
  
familyName("Hege", "1975");  
familyName("Stale", "1978");  
familyName("Kai Jim", "1983");
```

Resultat esperat:

```
Hege Refsnes. Born in 1975  
Stale Refsnes. Born in 1978  
Kai Jim Refsnes. Born in 1983
```

Funcions PHP

- **Valors d'arguments predeterminats**

- Es pot fer servir un paràmetre predeterminat.
- En aquests casos, quan es crida la funció sense arguments, aquesta pren el valor per defecte com a argument

```
function setHeight($minheight = 50) {  
    echo "The height is : $minheight <br>";  
}  
  
setHeight(350);  
setHeight(); // will use the default value of 50  
setHeight(135);  
setHeight(80);
```

Funcions PHP

- **Valors de retorn**

- Per permetre que una funció retorni un valor, s'ha d'utilitzar la paraula return.

```
function sum($x, $y) {  
    $z = $x + $y;  
    return $z;  
}  
  
echo "5 + 10 = " . sum(5, 10) . "<br>";  
echo "7 + 13 = " . sum(7, 13) . "<br>";  
echo "2 + 4 = " . sum(2, 4);
```

Funcions PHP

- **Nombre variable d'arguments (I)**

- En utilitzar l'operador ... davant del paràmetre de funció, la funció accepta un nombre desconegut d'arguments.
- Això també s'anomena funció variàdica.
- L'argument de la funció variàdica es converteix en una matriu.

```
function sumMyNumbers(...$x) {  
    $n = 0;  
    $len = count($x);  
  
    for($i = 0; $i < $len; $i++) {  
        $n += $x[$i];  
    }  
    return $n;  
}  
$a = sumMyNumbers(5, 2, 6, 2, 7, 7);  
echo $a;
```


Funcions PHP

- **Nombre variable d'arguments (II)**

- En cas de funcions **amb dos o més arguments**, només es pot tenir un argument amb longitud variable, i aquest ha de ser l'últim argument.

```
function myFamily($lastname, ...$firstname) {  
    $txt = "";  
    $len = count($firstname);  
    for($i = 0; $i < $len; $i++) {  
        $txt = $txt."Hi, $firstname[$i] $lastname.<br>";  
    }  
    return $txt;  
}  
  
$a = myFamily("Doe", "Jane", "John", "Joey");  
echo $a;
```

- Tenir l'operador ... en el primer dels dos arguments, generarà un error:

```
function myFamily(...$firstname, $lastname) {
```

Funcions PHP

- **Suggeriment de tipus dels arguments**

- PHP és un llenguatge de tipatge dinàmic (associa automàticament un tipus de dades a la variable, en funció del seu valor).
- A PHP s'han afegit les declaracions de tipus, podent especificar el tipus de dades esperat en declarar una funció
- Si s'afegeix la declaració **strict**, generarà un "Error fatal" si el tipus de dades no coincideix.

```
<?php
declare(strict_types=1); // strict requirement
function addNumbers(int $a, int $b) {
    return $a + $b;
}
echo addNumbers(5, "5 days");
?>
```

Funcions PHP

- **Suggeriment de tipus per al valor de retorn**

- Per especificar el tipus d'un valor de retorn per a una funció, afegiu el tipus després de la capçalera de la funció com aquesta:

```
function my_function(type $param1, type $param2, ...) : type
```

- Exemple en que es defineix la funció add() que accepta dos nombres enters i retorna un nombre enter:

```
function add(int $x, int $y): int
{
    return $x + $y;
}
echo add(10, 20);
```

Funcions PHP

- **Funcions dinàmiques**

- Les funcions dinàmiques permeten cridar funcions a través d'una variable que conté el nom de la funció.
- S'assigna el nom de la funció a una variable (nom exacte d'una funció existent).

```
function sumar($a, $b) {  
    return $a + $b;  
}  
function restar($a, $b) {  
    return $a - $b;  
}  
$operacio = "sumar"; // Variable que conté el nom de la funció a executar  
// Crida dinàmica  
$resultat = $operacio(5, 3); // Equival a cridar sumar(5, 3)  
echo $resultat; // Sortida: Resultat: 8
```

Funcions PHP

- **Funcions predefinides**

- PHP ofereix una àmplia varietat de funcions predefinides.
- Aquestes funcions estan organitzades per categories o tipus, segons la seva funcionalitat.
- Els tipus més comuns són:
 1. Funcions de cadenes de text
 2. Funcions de gestió d'arrays
 3. Funcions matemàtiques
 4. Funcions de data i hora
 5. Funcions de fitxers
 6. Funcions de validació i seguretat
 7. Funcions de xarxes

Funcions PHP

- **Funcions predefinides: Cadenes de text**

- Permeten manipular i processar cadenes de text. Alguns exemples són:
 - `strlen($cadena)`: Retorna la longitud d'una cadena.
 - `str_replace($cerca, $substituir, $cadena)`: Substitueix parts d'una cadena.
 - `strpos($cadena, $buscar)`: Retorna la posició d'una subcadena dins d'una cadena.
 - `trim($cadena)`: Elimina espais en blanc al principi i al final d'una cadena.

```
$text = "    Hola món!    ";  
$net = trim($text);  
echo strlen($net); // Sortida: 9 (sense els espais inicials i finals)
```

```
$original = "PHP és fantàstic!";  
echo str_replace("fantàstic", "increïble", $original); // Sortida: PHP és increïble!
```

Funcions PHP

- **Funcions predefinides: Gestió d'arrays**

- Permeten treballar amb llistes de dades. Alguns exemples són:

- `count($array)`: Retorna el nombre d'elements d'un array.
- `array_push($array, $valor)`: Afegeix un element al final de l'array.
- `array_pop($array)`: Elimina i retorna l'últim element de l'array.
- `sort($array)`: Ordena un array en ordre ascendent.

```
$fruites = ["Poma", "Plàtan", "Raïm"];
```

```
array_push($fruites, "Maduixa");  
print_r($fruites);
```

- ```
sort($fruites);
print_r($fruites); // Sortida: Array ([0] => Maduixa, [1] => Plàtan, [2] =>
Poma, [3] => Raïm)
```

# Funcions PHP

- **Funcions predefinides: Matemàtiques**

- Permeten la resolució de càlculs i operacions matemàtiques. Alguns exemples són:

- `abs($valor)`: Retorna el valor absolut.
    - `round($valor, $precisio)`: Arrodoneix un nombre a la precisió indicada.
    - `rand($min, $max)`: Genera un nombre aleatori entre un mínim i un màxim.
    - `sqrt($valor)`: Retorna l'arrel quadrada d'un nombre.

```
$nombre = -15;
echo abs($nombre); // Sortida: 15
```

```
$saleatori = rand(1, 100);
echo "Nombre aleatori: $saleatori"; // Sortida: Nombre aleatori: (un nombre
entre 1 i 100)
```



# Funcions PHP

- **Funcions predefinides: Data i hora**

- Permeten la gestió de dates i hores. Alguns exemples són:
  - `date($format)`: Aquesta funció retorna una cadena que representa la data/hora actual, formatada segons el format especificat.
  - `time()`: Retorna el temps actual en segons des del 1 de gener de 1970.
  - `strtotime($string)`: Converteix una cadena de text a timestamp.

```
echo date("Y-m-d H:i:s"); // Sortida: Data i hora actual (ex: 2024-11-12 14:30:45)
```

```
$actual = time();
echo "Timestamp actual: $actual"; // Sortida: 1732030446 (pot variar)
```

```
$timestamp = strtotime("next Monday");
echo date("Y-m-d", $timestamp); // Sortida: Data del proper dilluns
```

# Funcions PHP

- **Funcions predefinides: Fitxers**

- Permeten la gestió de fitxers i directoris. Alguns exemples són:
  - `file_get_contents($nomFitxer)`: Llegeix el contingut d'un fitxer.
  - `file_put_contents($nomFitxer, $contingut)`: Escriu contingut en un fitxer.
  - `unlink($nomFitxer)`: Esborra un fitxer.
  - `fopen($nomFitxer, $mode)`: Obre un fitxer en un mode específic (r, w, etc.).

```
$fitxer = "exemple.txt";
file_put_contents($fitxer, "Hola, món!\n");
```

```
$contingut = file_get_contents($fitxer);
echo $contingut; // Sortida: Hola, món!
```

# Funcions PHP

- **Funcions predefinides: Validació i Seguretat**

- Ajuden a validar dades i assegurar que són segures. Alguns exemples són:
  - `filter_var($valor, $filtre)`: Valida dades segons el filtre especificat (correus, URLs, etc.).
  - `htmlspecialchars($cadena)`: Escapa caràcters especials per prevenir XSS.
  - `password_hash($contrasenya, PASSWORD_DEFAULT)`: Crea un hash segur d'una contrasenya.
  - `password_verify($contrasenya, $hash)`: Verifica una contrasenya amb el hash generat.

```
$email = "exemple@correu.com";
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
 echo "El correu electrònic és vàlid.";
} else {
 echo "El correu electrònic no és vàlid.";
}
```

# Funcions PHP

- **Funcions predefinides: Xarxes**

- Tenen relació amb connexions a la xarxa o recuperació de dades. Alguns exemples són:

- `get_headers($url)`: Retorna els encapçalaments HTTP d'una URL.
- `fsockopen($host, $port)`: Obre una connexió de xarxa.

```
$url = "https://www.google.com";
$headers = get_headers($url);
print_r($headers); // Mostra els encapçalaments HTTP de la URL
```