

PHP 10

Funcions

```
1 function calculate(x, y){
2     //convert toNumber
3     let a = x * 1;
4     let b = y * 1;
5
6     //Sum
7     function sum(a, b){
8         return (a + b);
9     }
10
11    //Subtract
12    function subtract(a, b){
13        return (a - b);
14    }
15
16    return {
17        plus : sum,
18        minus :subtract
19    }
20 }
```

snappify.io

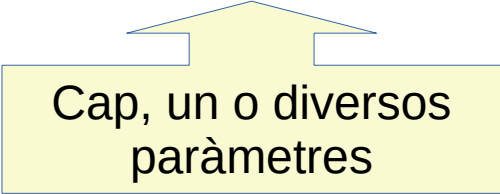
14)Funcions

- **Funcions**: Una de les **eines** més **importantes** en qualsevol llenguatge de programació.
- **Funció**: **conjunt d'instruccions** que al llarg del programa van a ser **executades multitud de vegades**.
 - **Estalvia** línies de codi: evita repetir el mateix codi una i altra vegada.
 - Es fa més **llegible**: més fàcil d'entendre quan es revisa el codi.
 - Podrà ser **cridada** i **executada** des de **qualsevol punt** del programa.

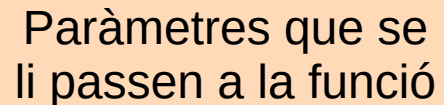
- Una funció pot rebre **paràmetres externs** dels quals depenga el seu resultat.
 - Segons el **paràmetre** o **paràmetres** amb els quals s'**invoque la funció**:
 - Aquesta retornarà **un resultat o un altre**.
- Les funcions han d'estar **definides abans de realitzar la crida** a la funció.
- Totes les funcions i classes de PHP tenen **àmbit global**.
 - Es poden **cridar des de fora d'una funció** fins i tot si van ser **definides dins**, i viceversa.

- **Sintaxi:**

```
function nom (paràmetre1, paràmetre2, ..., paràmetreN) {  
    instrucció1;  
    instrucció2;  
    ...  
    instruccióN;  
}
```



Cap, un o diversos
paràmetres



Paràmetres que se
li passen a la funció

- **Crida** (execució):

```
...  
nom (par1, par2, par3, ..., parn);  
...
```

- **Exemple 1:**

<?php // Exemple funció

// Declaració de funcions

```
function mostrarText($text) {  
    echo "<strong> El text a mostrar és el següent: </strong>" ;  
    echo $text;  
    echo "<br/>";  
}
```

// Fi de declaració de funcions

// crida a la funció

```
mostrarText("IES Simarro");
```

?>

Funció amb nom mostrarText
Espera que se li passe un paràmetre:
No s'indica el tipus (int, float...)

En acabar, torna al punt on va ser invocada

- Podem crear **funcions que retornen dades** (valors **concrets**).
- Executen un codi i al final inclouen una **sentència return seguida del resultat** de la funció.
- La sentència **return** indica que **s'ha arribat al final de la funció** i es **torna** com a **resultat** de la mateixa el contingut especificat a continuació del **return**.
 - Pot retornar una **variable**, un **nombre**, una **cadena** de text, etc.

• Per **exemple**:

```
function f1(){  
    return "bon dia";  
}
```

```
function f2($valor){  
    $calcul = $valor/10;  
    return $calcul;  
}
```

- **Exemple:**

```
function f3($usuari, $emailAdministrador){  
  
    return "Malauradament ".$usuari." no disposa de permisos. Per a  
        sol·licitar informació pot escriure a ".$emailAdministrador;  
  
}
```


- **Exemple 2:**

```
<?php //Exemple funcions
//inici declaració de funcions
function operacions ($n1, $n2, $operacio) {
    $resultat = 0;
    if ($operacio == "Sumar" ) {
        $resultat = $n1 + $n2;
    } else if ($operacio == "Restar" ) {
        $resultat = $n1 - $n2;
    } else if ($operacio == "Multiplicar" ) {
        $resultat = $n1 * $n2;
    }
    return $resultat; // Tornar el resultat
}
//final de la declaració de funcions
```

Requereix
3 paràmetres

Se substitueix la seua invocació
allà on apareix pel valor que retorna.

```
// Crida a la funció operacions:
//podem cridar i després mostrar per pantalla
$valorResultat = operacions (5, 7, "Sumar" );
echo "Resultat: ".$valorResultat."<br/>" ;
// O podem mostrar per pantalla directament
echo "Resultat: ".operacions(15, 8, "Restar" )."<br/>" ;
```

Seria com escriure:
echo (15-8) o
echo 7
directament

?>

- **Exemple 3:**

No Requereix
cap paràmetre

```
<?php // Exemple funcions  
    function mostrarTextError {  
        echo "<strong> S'ha produït un error </strong><br/>" ;  
    }  
  
//final de declaració de funcions  
  
//invocació  
mostrarTextError();  
?>
```

- PHP **no admet la sobrecàrrega** de funcions.
- **Tampoc** és possible '**desdefinir**' ni **redefinir** funcions prèviament declarades.
- Els **noms** de les funcions són **insensibles a majúscules-minúscules**, :
 - És convenient **cridar** a les funcions **tal com apareixen** en les seues **declaracions**.

- En PHP és possible també fer crides a funcions **recursives**.
- Les crides a funcions recursives amb més de 100-200 nivells de recursivitat poden **esgotar la pila** i ocasionar la finalització de l'script en curs.
 - **Recursivitats infinites**: un **error greu** de programació.

- **Exemple 4:**

```
<?php
function recursivitat($a){
    if ($a < 20) {
        echo "$a <br/>";
        recursivitat($a + 1);
    }
}
?>
```