

PHP

FONCTIONS & PROCEDURES

Les fonctions

- PHP fournit des centaines de fonctions intégrées
Ex : `strlen()`, `if()`, `date()`, `print()`, etc.
- C'est les parenthèses qui indique que c'est une fonction et non une constante !
- Mais on peut aussi en créer !

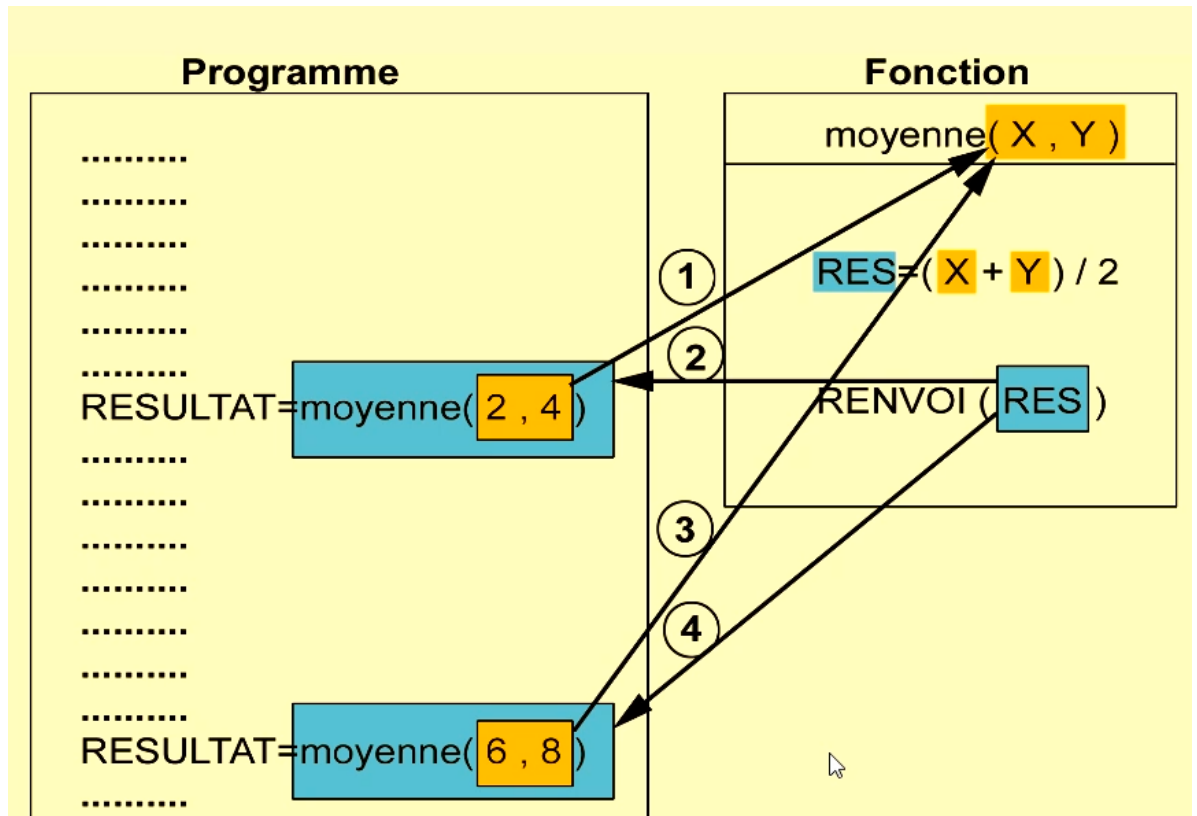
fonction

- Le mot clef en PHP est **function()**
- Elle peut prendre aucun ou plusieurs paramètres
 - exemple : `phpinfo()`
- Elle retourne une seule valeur
- Le nom d'une fonction suit les mêmes règles que pour les variables
- Le nom des fonctions n'est pas sensible à la casse.
- Le mot clef pour retourner une valeur est **return**

La fonction : moyenne

Prend deux paramètres X et Y

Retourne un seul résultat : RES



Implémentation en PHP

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      // déclaration de la fonction moyenne
      function moyenne($x,$y)
      {
        $res=($x+$y)/2;
        return $res;
      }

      echo "Moyenne de 2 et 4 est ".moyenne(2,4)."<br/>";
      echo "Moyenne de 6 et 8 est ".moyenne(6,8)."<br/>";

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

Avantages d'une fonction

- Factorisation du code
- Maintenance plus simple

Plusieurs valeurs à retourner

- On peut utiliser un tableau !

```
<?php
// déclaration de la fonction moyenne
function calcul($x,$y)
{
    $somme=($x+$y);
    $moyenne=$somme/2;
    $res=array($somme,$moyenne);
    return $res;
}
$resultat=calcul(2,4);
echo "Somme de 2 et 4 est ".$resultat[0]."<br/>";
echo "Moyenne de 2 et 4 est ".$resultat[1]."<br/>";

?>
```

- On peut passer plusieurs éléments (pas de limite)

Paramètre optionnel

```
<?php
// déclaration de la fonction moyenne
function calcul($x,$y,$unite=" euros")
{
    $res=($x+$y);
    // Concaténation
    $res.=$unite;
    return $res;
}
$resultat=calcul(2,4);
echo "Somme de 2 et 4 est ".calcul(2,4)."<br/>";
echo "Somme de 3 et 4 est "calcul(2,4," dollars")."<br/>";
?>
```

résultat gardé
dans une
variable



résultat
affiché puis
perdu !



Dans les paramètres \$unite à une valeur par défaut et devient donc optionnel lors de l'appel de la fonction.

Externaliser les fonctions

Intérêt : pouvoir créer des fonctions et les utiliser dans plusieurs programmes

Factorisation encore du code

Maintenance simplifiée (un seul fichier à modifier)

Principe : mettre les fonctions dans un fichier différent

Ces fichiers se terminent (par convention) par **inc.php**

Externaliser les fonctions

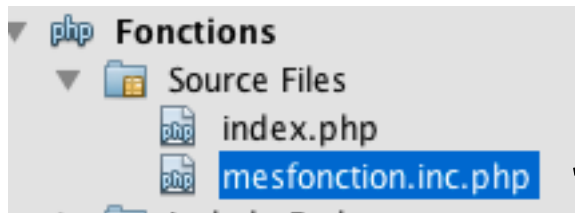
The screenshot shows a 'New PHP File' dialog box with the following fields and values:

- File Name:** mesfonction.inc.php
- Project:** Fonctions
- Folder:** (empty field) with a **Browse...** button next to it.
- Created File:** /Applications/MAMP/htdocs/Fonctions/mesfonction.inc.php

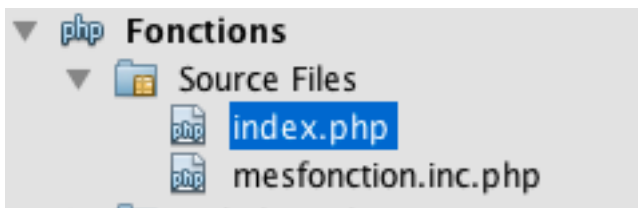
On the left side of the dialog, there is a 'Steps' section with a list:

1. Choose File Type
2. **Name and Location**

Externaliser les fonctions



```
<?php
function calcul($x,$y,$unite=" euros")
{
    $res=($x+$y);
    // Concaténation
    $res.=$unite;
    return $res;
}
?>
```



```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <?php
    require("mesfonction.inc.php");
    echo "Somme de 2 et 4 est ".calcul(2,4)."<br/>";
    echo "Somme de 3 et 4 est ".calcul(2,4," dollars")."<br/>";
    ?>
  </body>
</html>
```

Permet d'inclure le code