

Les fonctions en PHP

Contenu

- **Le concept de fonction**
- **Exemples de fonctions incluses dans PHP**
- **Créer ses propres fonctions**
- **L'utilité des fonctions**
- **Les fonctions *include*, *require*, *include_once* et *require_once***

Le concept de fonction

Rappel de vos maths du secondaire

- En mathématiques, une **fonction** fait correspondre une variable (**x**) à une autre (**y**)
 - Ex: La fonction « **y** = 2**x** »
 - Si **x** = 3, alors **y** = 6
 - Si **x** = 25, alors **y** = 50
 - Etc.
- On peut donc dire qu'une fonction prend une valeur d'entrée (**x**) et produit une valeur de sortie (**y**)

Les fonctions en programmation

- **Une fonction, en programmation:**

- Peut prendre une ou des variables en entrée, qu'on appelle des **paramètres**
- Peut produire (**retourner**) une valeur en sortie (un **résultat**)
- Peut effectuer des actions ayant un impact direct sur l'environnement dans lequel s'exécute la fonction (**effet de bord** ou *side effect*)
- Ex: Afficher du texte à l'écran

La fonction « Grille-pain d'hôtel »



Photo par Wtshymanski sur Wikimedia Commons (CC BY-SA 3.0)

Paramètre:
Une tranche de pain

Résultat:
Une toast!

Effet de bord: Dégage de la chaleur

Appel de fonction

- On dit qu'on **appelle** une fonction lorsqu'on l'exécute en lui passant ou non des paramètres
- La syntaxe en PHP est la suivante:

\$resultat = nomDeLaFonction(paramètres);

Exemples de fonctions incluses dans PHP

La fonction *isset*

- **isset(\$variable)**

- Paramètre: une variable dont on veut vérifier l'existence
- Résultat: **true** si la variable existe, **false** si elle n'existe pas
- Effet de bord: aucun

La fonction *array_push*

- **array_push(\$tableau, \$valeur);**
 - Paramètres:
 - Le tableau auquel ajouter une valeur
 - La valeur à ajouter
 - Résultat: L'indice de l'élément qui a été ajouté au tableau
 - Effet de bord: ajoute la valeur à la fin du tableau passé en paramètre

La fonction *count*

- **count(\$tableau);**

- Paramètre: un tableau
- Résultat: le nombre d'éléments dans le tableau
- Effet de bord: aucun

La fonction *htmlspecialchars*

- **htmlspecialchars(\$chaineDeCaracteres);**
 - Paramètre: une chaîne de caractères
 - Résultat: une nouvelle chaîne de caractères correspondant à celle reçue en paramètre dont certains caractères utilisés en HTML ont été remplacés (ex: < devient <)
 - Effet de bord: aucun

La fonction *strtolower*

- ***strtolower(\$chaineDeCaracteres);***

- Paramètre: une chaîne de caractères
- Résultat: une nouvelle chaîne de caractères correspondant à celle reçue en paramètre dont on aurait converti toutes les lettres en minuscules
- Ex:
 - "Bonjour" => "bonjour"
 - "Une Pomme" => "une pomme"
 - "BONJOUR123" => "bonjour123"
- Effet de bord: aucun

La fonction *strtoupper*

- **strtoupper(\$chaineDeCaracteres);**
 - Paramètre: une chaîne de caractères
 - Résultat: une nouvelle chaîne de caractères correspondant à celle reçu en paramètre dont on aurait converti toutes les lettres en majuscules
 - Ex:
 - "Bonjour" => "BONJOUR"
 - "Une Pomme" => "UNE POMME"
 - "bonjour123" => "BONJOUR123"
 - Effet de bord: aucun

La fonction *intval*

- **intval(\$variable);**

- Paramètre: une variable quelconque
- Résultat: la variable convertie en nombre entier s'il ne s'agissait pas déjà d'un nombre entier, ex:
 - 13.5 => 13
 - "17" => 17
 - "allo" => 0
 - "17allo" => 17
- Effet de bord: aucun

La fonction *in_array*

- **in_array(\$valeur, \$tableau)**

- Paramètres: une valeur et un tableau
- Résultat: **true** si la valeur est présente dans le tableau, **false** si elle n'est pas présente
- Effet de bord: aucun

La fonction *asort*

- **asort(\$tableau);**

- Paramètre: un tableau
- Résultat: **true** si l'opération a réussi, **false** si elle a échoué
- Effet de bord: Modifie le tableau pour placer ses éléments en ordre croissant

Autres fonctions

- **PHP inclut de nombreuses autres fonctions**
- **Elles sont décrites dans une section de la documentation du langage**
 - <https://www.php.net/manual/fr/funcref.php>

Créer ses propres fonctions

Fonction sans paramètre et sans résultat

```
// Définir la fonction  
function direBonjour() {  
    echo 'Bonjour';  
}
```

```
// Appeler la fonction  
direBonjour();
```

```
// Effet de bord: Affiche « Bonjour » à  
l'écran
```

Fonction avec un paramètre

```
// Définir la fonction  
function direBonjour($nom) {  
    echo "Bonjour $nom";  
}
```

```
// Appeler la fonction  
direBonjour("Lisa");
```

```
// Effet de bord: Affiche « Bonjour Lisa » à  
l'écran
```

Fonction avec deux paramètres

```
// Définir la fonction  
function direBonjour($prenom, $nom) {  
    echo "Bonjour $prenom $nom";  
}  
  
// Appeler la fonction  
direBonjour("Lisa", "Simpson");  
  
// Effet de bord: Affiche « Bonjour Lisa Simpson » à l'écran
```

Fonction renvoyant un résultat

```
// Définir la fonction
function genererMessageBonjour($prenom, $nom) {
    return "Bonjour $prenom $nom";
}

// Appeler la fonction
$messageBonjour = genererMessageBonjour("Lisa",
"Simpson");

echo $messageBonjour; // Affiche « Bonjour Lisa
Simpson »
```

Fonction renvoyant un résultat

- L'instruction **return** met fin à la fonction
- On peut en mettre plusieurs dans une fonction (par exemple dans des conditions)
 - Seul le premier à s'exécuter aura un effet

```
function quoiPorter($temperature) {  
    if ($temperature < 5) {  
        return "manteau d'hiver";  
    }  
    if ($temperature < 15) {  
        return "manteau d'automne";  
    }  
    return "t-shirt";  
}
```

L'utilité des fonctions

Réutilisation du code

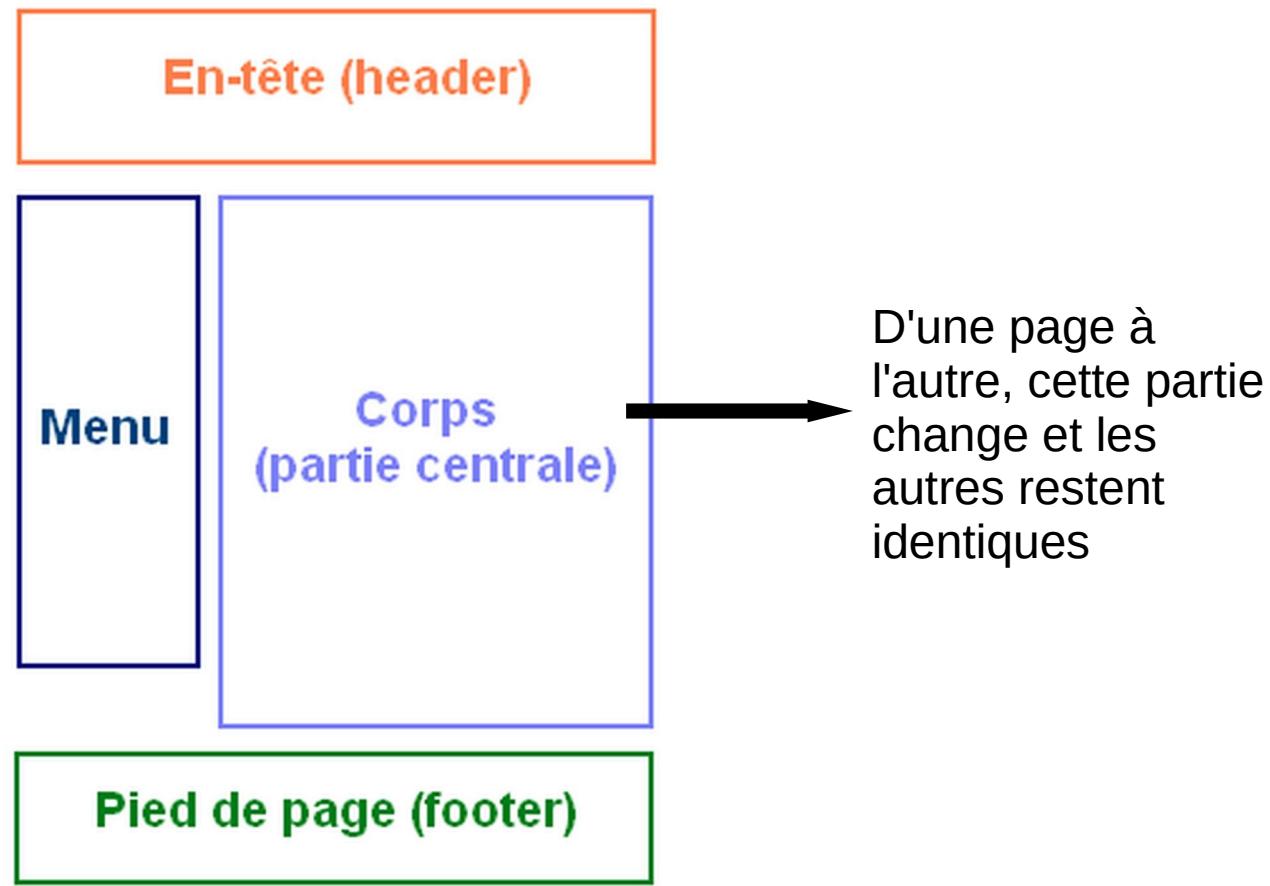
- **Les fonctions permettent de réutiliser du code facilement**
- **C'est pour ça que PHP inclut d'emblée une panoplie de fonctions permettant d'effectuer des opérations courantes**
 - Ex: La fonction `strtolower` évite à tout le monde de devoir réécrire du code permettant de convertir une chaîne de caractères en minuscules

Clarté du code

- **Les fonctions permettent de rendre le code plus clair**
 - Ex: si je vois une ligne de code « afficherFormulaire(); », c'est assez clair qu'elle affiche un formulaire!
 - Code de plus haut niveau

Les fonctions *include*, *require*, *include_once* et *require_once*

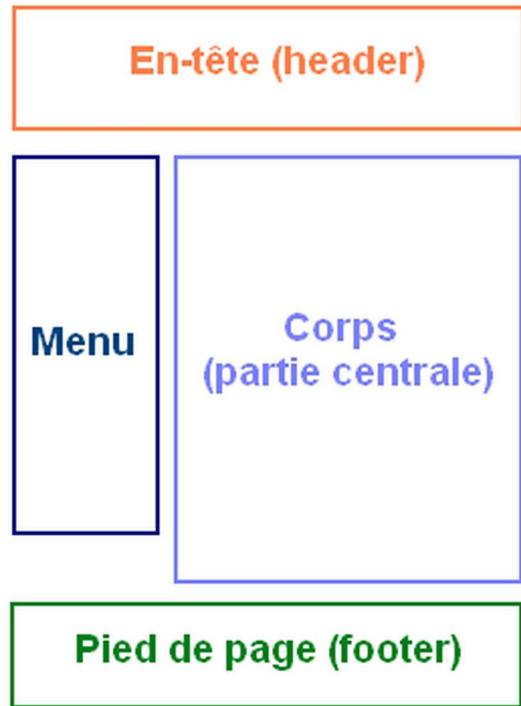
Exemple de structure des pages d'un site Web



OpenClassrooms (CC BY-SA 4.0)

La fonction *include*

- La fonction *include* permet de copier le contenu provenant d'un autre fichier PHP ou HTML



- On pourrait avoir les fichiers suivants:
 - **entete.php**
 - **menu.php**
 - **pied.php**
- Puis, dans chacune de nos pages:

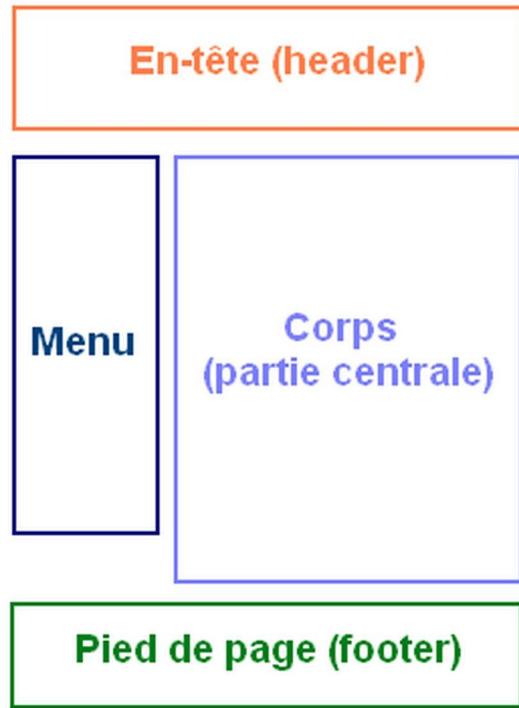
```
include("entete.php");  
include("menu.php");
```

```
// Mettre le code du corps ici
```

```
include("pied.php");
```

La fonction *require*

- La fonction *require* fait la même chose que *include*, mais lance une erreur et arrête l'exécution si le fichier à inclure est introuvable



- On pourrait avoir les fichiers suivants:
 - **entete.php**
 - **menu.php**
 - **pied.php**
- Puis, dans chacune de nos pages:

```
require("entete.php");  
require("menu.php");
```

```
// Mettre le code du corps ici
```

```
require("pied.php");
```

Fichiers de fonctions

- **On peut aussi utiliser *include* pour avoir accès à des fonctions définies dans un autre fichier**
- **On pourrait donc définir des fonctions à un seul endroit et les réutiliser n'importe où dans notre projet**

include_once et *require_once*

- Les fonctions *include_once* et *require_once* se comportent respectivement comme *include* et *require*, mais s'assurent qu'un même fichier n'est jamais inclus plus d'une fois
- Exemple:
 - **fichierA.php**
 - `require_once("fichierB.php");`
 - **fichierC.php**
 - `require_once("fichierA.php");`
 - `require_once("fichierB.php");`
 - `// ne fait rien, car déjà inclus dans fichierA.php`

Fin de la présentation

Des questions?



Photo par Simone Secci sur Unsplash