PHP

Fonctions et Importations

Dans ce module

- Fonctions
  - *Fonction simple*
  - *Fonction avec paramètres*
  - *Fonctions avec return*
  - *Fonction avec typage fort*
  - *Fonction anonyme en Callback*
  - *Scope des variables de fonctions*
- Importations
  - *Importer des scripts PHP*
- Pratique
- Découverte : GET et POST
- Découverte : Cookies et Sessions

Fonction simple

- Une fonction va nous permettre de conserver un petit programme ou un code simple.
- On pourra ainsi répéter le code d'une fonction où l'on veut et autant de fois que l'on veut tant que la fonction est créée ou importée avant d'être utilisée.
- Une fonction doit être appelée pour être utilisée, donc on lui donnera un nom unique dans le scope de son utilisation. On peut aussi stocker une fonction directement dans une variable mais l'utilisation est moins commun car limité dans sa porté d'utilisation. On parlera aussi de fonction anonyme dans ce cas.
- Dans la pratique, on sépare les fonctions dans des scripts (fichiers) PHP à part. Il faudra alors les importer pour les utiliser. Ces scripts ne contiennent alors que du PHP.

```php
function NAME()
{
    // Code
}
```

Définition d'une fonction classique (sans paramètre)

```php
$VAR = function() {
    // Code
};
```

Définition d'une fonction dans une variable (sans paramètre).
Usage « local » à la résolution d'un problème complexe

Fonction simple - Exemple

```php
// Définition
function bonjour()
{
    echo 'Bonjour !<br>';
}

// Appel
bonjour();
bonjour();
bonjour();
```

Bonjour !
Bonjour !
Bonjour !

```php
// Définition
$var = function() {
    echo 'Bonjour !<br>';
};

// Appel
$var();
$var();
$var();
```

Bonjour !
Bonjour !
Bonjour !

Fonction avec paramètres

- Une fonction peut demander un ou des paramètres pour la réalisation de son code.
- Les paramètres demandés sont alors obligatoire pour utiliser la fonction.
- Les paramètres peuvent être définie avec une valeur par défaut, ce qui les rends optionnels lors de l'appel de la fonction.
- Les paramètres peuvent être de n'importe quels type et en nombre théoriquement infinis.

```php
function NAME(PARM1, PARM2, PARM3)
{
    // CODE
}
```

```php
$VAR = function(PARM1, PARM2, PARM3) {
    // CODE
};
```

Fonction avec paramètre - Exemple

```php
function html_($text = '')
{
    echo $text.'<br>';
}

function html_h4($text)
{
    echo '<h4>' . $text . '</h4>';
}

function html_p($text, $br = false)
{
    echo '<p>' . $text . '</p>' . ($br ? '<br>' : '');
}

html_h4('Titre h4’);
html_();
html_p('Paragraphe 1', true);
html_p('Paragraphe 2');
```

Titre h4

Paragraphe 1

Paragraphe 2

Fonction avec return

- Une fonction ne sert pas qu'à afficher des données, elle peut aussi renvoyer une valeur.
- Pour cela on utilise l'instruction « return » suivie de la valeur à retourner.
- A savoir que return arrête aussi la fonction (comme « break » pour une boucle).
- Une fonction peut retourner la valeur d'une variable ou directement une valeur.
- Une fonction ne peut retourner qu'une seule valeur mais la valeur peut être de n'importe quels types.

```php
function NAME()
{
    // CODE

    return VAL;
}
```

Définition d'une fonction avec retour de valeur

```php
$VAR = function() {
    // CODE

    return VAL;
};
```

Définition d'une fonction dans une variable avec retour de valeur

Fonction avec return - Exemple

```php
function priceTTC($price, $vat = 20)
{
    return round(
        $price * (1 + $vat/100)
    );
}

// Valeur de retour récupéré dans une variable
$price_ttc = priceTTC(109.09, 10); // 120
$price_ttc = priceTTC(45.83); // 55

// Valeur de retour non récupéré
priceTTC(99);
```

```php
priceTTC(); // ! ERROR ! 1er paramètre obligatoire !
```

Fonction avec return - Exemple

```php
// Définition
function priceTTC($price, $vat = 20)
{
    $vatAccepted = [20, 10, 5.5];

    if ( !in_array($vat, $vatAccepted) ) {
        echo '<br>! ERROR ! VAT value is not accepted !<br>';
        return; // return null;
    }

    return round(
        $price * (1 + $vat/100)
    );
}

// Appel
$price_ttc = priceTTC(45.83, 30);
if ($price_ttc === null) {
    // ...
}
```

Fonction avec typage fort

- On peut imposer un type précis pour un paramètre (on parle alors de **typage fort**).
- On peut imposer un type précis pour la valeur de retour (on parle alors de **typage fort**).

```
function NAME(TYPE PARM1, PARM2, TYPE PARM3)
{
    // CODE
}
```

Fonction avec paramètres :

1. PARM1 obligatoire (type obligatoire)
2. PARM2 obligatoire
3. PARM3 optionnel car valeur par défaut

```
function NAME(PARM1, PARM2, PARM3) : TYPE
{
    // CODE
}
```

Fonction avec typage fort
pour le valeur du return

Fonction avec typage fort - Exemple

```php
function priceTTC(float $price, int $vat = 20) : float
{
    return round(
        $price * (1 + $vat/100)
    );
}
```

Fonction anonyme en callback

- Les fonctions anonymes, aussi appelées « fermetures » ou « closures » permettent la création de fonctions sans préciser leur nom. Elles sont particulièrement utiles comme fonctions de rappel « callback », mais leur utilisation n'est pas limitée à ce seul usage.
- Elles peuvent être contenu dans une variable.
- Elles peuvent être utilisée comme valeurs dans les paramètres d’une fonction appelée.

```
function() {
    // CODE
};
```

Fonction anonyme

```
(function() {
    // CODE
})();
```

Fonction anonyme auto-appelé

```
$VAR = function() {
    // Code
};
```

Fonction anonyme comme valeur de variable.

Fonction anonyme en callback - Exemple

```php
$arr = [2, 4, 8, 16, 32];

$arr = array_map(

    function($value) { // Callback function
        return $value**2;
    },

    $arr // Array to work

); // $arr = [4, 16, 64, 256, 1024]
```

→ « array_map » est une fonction PHP utilisant une fonction anonyme en 1er paramètre pour modifié les valeurs d'un tableau en 2ème paramètre.

```php
$arr = [2, 4, 8, 16, 32];

$arr = array_map(function($value) {
    return $value**2;
}, $arr);

// $arr = [4, 16, 64, 256, 1024]
```

→ Véritable apparence normalisée PSR-2

Scope des variables de fonctions

- Les variables créées dans une fonction n'existe que la fonction. Elles sont automatiquement détruites lorsque la fonction s'arrête. Cela inclut les paramètres de la fonction.
- Les paramètres d'une fonction sont par défaut des copies de valeurs. Modifier un paramètre ne modifie pas la variable/valeur d'origine qui a été envoyée en paramètre.
- Pour modifier une variable envoyée en paramètre d'une fonction sans return, il faut que dans la définition de la fonction le paramètre soit déclaré en « référence » avec le symbole « & » avant le « $ ».

```php
function foo($a)
{
    $a = 'hello';
}

$a = 5;
foo($a);
// $a = 5
```

```php
function foo($a)
{
    $a = 'hello';
    return $a;
}

$a = 5;
$a = foo($a);
// $a = 'hello'
```

```php
function foo(&$a)
{
    $a = 99;
}

$a = 5;
foo($a);
// $a = 99
```

Importer des scripts PHP

- Une importation d'un script PHP c'est comme si le code du script importé avait été écrit là où l'importation est faite.
- Une importation de code entre script se fait avec (PATH est un chemin à partir du script appelant) :
  - *« include(PATH) » : **import avec warning** si le script cible n'existe pas. **Erreur** si appels doublon !*
  - *« include_once(PATH) » : **import avec warning** si le script cible n'existe pas. **Gère** les appels doublon.*
  - *« require(PATH) » **: import avec erreur** si le script cible n'existe pas. **Erreur** si appels doublon !*
  - *« require_once(PATH) » : **import avec erreur** si le script cible n'existe pas. **Gère** les appels doublon.*

```
/-- mysite.com
    /-- functions
        -- math_functions.php
        -- html_functions.php
    -- index.php
    -- functions.php
```

```php
// Dans index.php
require('functions.php');
```

```php
// Dans functions.php
require_once('functions/math_functions.php');
require_once('functions/html_functions.php');
```

Pratique

- Créer une page « **functions.php** » pour y mettre toutes les fonctions de l'exercice. Pour simplifier, nous appellerons directement les fonctions dans le script fonctions. Par la suite, nous importerons « functions.php » pour l'utiliser dans d'autres scripts PHP.
- Dans votre script « **functions.php** » réaliser les exercices suivants :
  1. Reprendre les exercices corrigés (ou votre version à vous) sur les conditions et les boucles. Vous devez donc copier les corrigées dans des fonctions individuelles, que vous nommerez. Les variables de départ utilisé dans les exercices devront maintenant être des paramètres.
  2. Réaliser une fonction pour calculer un prix HT à partir d'un prix TTC. (Inspirez-vous de l'exemple similaire dans le cours).
  3. Réaliser une fonction nommée « **dd** » qui prendra 1 paramètre quelconque et l'affichera avec un « **var_dump()** ». La fonction devra arrêter le code PHP avec « **exit();** ». Elle servira a comme débogage avec arrêt du code PHP.
  4. Réaliser une fonction demandant trois nombres entier **x, y, z** et qui les triera par ordre croissant. Vous ne devez pas utiliser de fonctions toute faite mais simplement des conditions.
  5. Réaliser une fonction affichant une suite de nombres pairs dans une plage déterminée par 2 paramètres de la fonction : un minimum et un maximum. La fonction les affichera dans un ordre croissant, sans inclure les bornes.

Pratique

- Dans un nouveau dossier, créer un mini-site avec une page « index.php », « menu.php » et « footer.php ». Dans « index.php » ajouter le minima HTML et avec PHP importer « menu.php » au début de « body » puis importer « footer.php » avant la fin de « body ».
- Créer 2 autres pages comme « index.php » : « cv.php » et « contact.php ».
- Dans « menu.php », insérer une balise « nav » avec des balises liens HTML « a » pour entre les 3 pages : index, cv et contact.
- Dans « footer.php », insérer une balise « footer » avec quelques liens « facebook », « twitter » ou « linkedin ».
- Tester votre mini-site avec un menu et un footer centralisés et dynamiquement insérés.

Découvertes

- **GET et POST**
  1. http://creersonsiteweb.net/page-php-get-post
  2. https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/912799-transmettez-des-donnees-avec-lurl

- **Cookies et Sessions**
  1. http://creersonsiteweb.net/page-php-cookies
  2. http://creersonsiteweb.net/page-php-sessions
  3. https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/4239476-session-cookies

PHP

Fonctions et Importations

Fin du module