



Cours PHP

Maîtriser les bases du développement avec PHP

Décembre 2018 - E2 - Guillaume Lelasseur

Sommaire

Sommaire	1
Installation	4
Vérifier si PHP est déjà installé	4
Linux Debian	4
Mac	5
Windows	5
Bases	6
Serveur de développement	6
Déclarer du PHP	6
Si la page se termine par du PHP	7
Les instructions	7
Les commentaires en PHP	8
Les types en PHP	8
Les variables en PHP	9
La concaténation en PHP	9
Les conditions en PHP	9
if, elseif & else	9
switch	10
Les boucles en PHP	11
Boucle while	11
Boucle for	11
Les fonctions en PHP	12
Tableaux	13

Déclarer un tableau	13
Tableau à plusieurs dimensions	13
Variables super globales	14
\$_SERVER	14
\$_GET	14
\$_POST	14
Bases de données	15
Installation	15
PHPMyAdmin	15
Installation	15
Connexion à la base de données	15
Select	16
Simple select	16
Select avec paramètres	16
Insert	16
Objets	16
Définition d'une classe	17
Créer un objet avec le mot-clé new	18
La pseudo-variable \$this	18
Méthode constructeur	19
Méthode destructeur	19
L'héritage avec le mot-clé extends	20
visibilité	20
Visibilité publique	20
Visibilité privée	20
Visibilité protégée	21
Visibilité des propriétés	21
Visibilité des méthodes	21

Installation

Vérifier si PHP est déjà installé

Lancer la commande suivante dans un terminal et vérifier que PHP 7 est bien installé :

```
php -v
```

Si une version supérieure ou égale à 7 de PHP est retournée, alors ce chapitre peut être sauté.

La console retournera alors un résultat similaire à celui-ci :

```
[guillaume@guillaume-manjaro ~]$ php -v
PHP 7.1.19 (cli) (built: Aug 17 2018 18:03:17) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.1.0, Copyright (c) 1998-2018 Zend
Technologies
```

Linux Debian

❑ Installer le paquet php :

```
apt install php
```

Mac

- Installer le gestionnaire de paquet Homebrew
- Installer le paquet php :

```
brew install php
```

Windows

- Télécharger la dernière version stable disponible de PHP (au minimum PHP7) sur : <https://windows.php.net/download>
- Extraire le fichier compressé téléchargé
- Créer un dossier php dans C:\Program Files (x86) et y placer le dossier décompressé
- Créer une variable d'environnement pointant vers ce dossier

Bases

Serveur de développement

Pour créer un serveur de développement en PHP dans un terminal, il est nécessaire de se placer dans le dossier souhaité et de lancer la commande suivante :

```
cd /home/julie/workspace/projet_php  
php -S 0.0.0.0:8080
```

Déclarer du PHP

On place le code PHP entre des balises spécifiques :

```
<?php  
... ici on placera du code php ...  
?>
```

🔥🔥🔥 On peut placer du PHP où on le souhaite au sein du code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Ma page HTML</title>
</head>
<body>
  <h1>Bienvenue sur mon site</h1>
  <p>
    <?php echo "Voici mon site" ?>
  </p>
</body>
</html>
```

Si la page se termine par du PHP

Dans ce cas, la fermeture de la balise PHP à la fin est facultative.

☀️ Par convention, on ne ferme pas la balise de fin pour les pages comportant seulement du PHP.

Les instructions

Chaque instruction doit être terminée par un point virgule en PHP :

```
<?php
echo "Bonjour";
?>
```


Les commentaires en PHP

Comme en C, il est possible de faire des commentaires sur une ligne et plusieurs lignes, pour cela il faut utiliser la syntaxe suivante :

Pour faire un commentaire sur une seule ligne, on utilisera la syntaxe suivante :

```
// Un commentaire sur une seule ligne
```

Quand au commentaire sur plusieurs ligne, on utilisera la syntaxe suivante :

```
/* Un commentaire  
sur plusieurs  
lignes */
```

Les types en PHP

```
$variable = TRUE; // un booléen  
$variable = "foo"; // une chaîne de caractères  
$variable = 'foo'; // une chaîne de caractères  
$variable = 12; // un entier  
$variable = 12.3; // un double
```

Pour connaître le type d'une variable on utilise la fonction `gettype()` :

```
echo gettype($variable);
```

Les variables en PHP

```
$prenom = "Julie";  
$nom = 'Dupont';
```

La concaténation en PHP

La concaténation est le fait de fusionner le contenu de plusieurs variables et ou chaînes. Pour y parvenir on utilisera le « . » pour séparer les différents éléments que l'on souhaite fusionner. Ainsi on utilisera la syntaxe suivante :

```
$nom_complet = $prenom . ' ' . $nom;
```

```
// Concaténation et affichage  
echo 'Bonjour ' . $nom_complet;
```

Les conditions en PHP

if, elseif & else

```
if ($a > $b)  
{  
    echo "a est plus grand que b";  
}  
elseif ($a == $b)  
{
```

```
    echo "a est égal à b";
}
else
{
    echo "a est plus petit que b";
}
```

switch

```
switch ($i)
{
    case 0:
        echo "i égal 0";
        break;
    case 1:
        echo "i égal 1";
        break;
    case 2:
        echo "i égal 2";
        break;
}

// Équivalent avec des if et elseif
```

```
if ($i == 0)
{
    echo "i égal 0";
}
elseif ($i == 1)
{
    echo "i égal 1";
}
elseif ($i == 2)
{
    echo "i égal 2";
}
```

Les boucles en PHP

Boucle while

```
$i = 1;
while ($i <= 10)
{
    echo $i++;
}
```

Boucle for

```
for ($i = 1; $i <= 10; $i++)
{
    echo $i;
}
```

Les fonctions en PHP

```
function nom_fonction($parametre1, $parametre2)
{
    ... code de la fonction ...
}

// Appel de la fonction
nom_fonction(2, "Lorem Ipsum");
```

Tableaux

Déclarer un tableau

Créer un tableau avec la fonction `array()` :

```
$tableau = array("Premier élément du tableau", "Deuxième élément");
```

Créer un tableau avec la nouvelle syntaxe introduite avec PHP 7 :

```
$tableau = ["Premier élément du tableau", "Deuxième élément"];
```

Définir une clé spécifique :

```
$tableau[2] = "Valeur";
```

Accéder aux éléments d'un tableau :

```
echo $tableau[0];
```

Tableau à plusieurs dimensions

```
$classes = [];  
$classes["E1A"] = ["Julie", "Nadia", "Paul", "Sophie"];  
$classes["E1B"] = ["Antoine", "Mélanie", "Melissa"];  
echo $classes["E1A"][1]; // Affiche Nadia
```

Variables super globales

\$_SERVER

Cette variable super globale permet notamment d'obtenir une multitude d'informations sur le serveur et sur le client.

\$_GET

Tableau généré par PHP qui retourne les variables d'URL.

\$_POST

Tableau généré par PHP qui retourne les variables contenues dans le corps d'une requête POST.

Bases de données

Installation

PHPMyAdmin

Installation

Utiliser MAMP sur Mac ou XAMMP sur Windows pour obtenir PHPMyAdmin.
Créer les bases de données avec.

Connexion à la base de données

```
<?php
$dsn = 'mysql:dbname=testdb;host=127.0.0.1';
$user = 'dbuser';
$password = 'dbpass';

try {
    // Connexion à la base de données
    $bdd = new PDO($dsn, $user, $password);

    // Affichage des erreurs
```



```
$bdd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch (PDOException $e)
{
    echo 'Échec lors de la connexion : ' . $e->getMessage();
}
```

Select

Simple select

```
$req = $bdd->query('SELECT nom, couleur, calories FROM fruits');
$tous_les_fruits = $req->fetchAll();

echo $tous_les_fruits[0]['nom']; // Affiche le nom du premier fruit
```

Select avec paramètres

```
$prep = $bdd->prepare('SELECT nom, couleur, calories
FROM fruit
WHERE calories < ? AND couleur = ?');
$prep->execute([150, 'rouge']);
$fruits = $prep->fetchAll();

echo $fruits[1]['couleur']; // Affiche la couleur du deuxième fruit
```

Insert

```
$prep = $bdd->prepare('INSERT INTO item(title, link) VALUES(?, ?)');
$prep->execute(["Lorem Ipsum", "http://gnu.org"]);
```

Objets

Définition d'une classe

- On commence par le mot clé classe.
- On définit le nom de la classe
 - Par convention, on commence par une majuscule
- Le contenu de la classe, les propriétés et les méthodes de celle-ci est placé entre des accolades

```
<?php

class SuperClasse {
    // Déclaration d'une propriété
    public $toto = 'Valeur par défaut';

    // Déclaration des méthodes
    public function afficheBonjour() {
        echo "Bonjour";
    }
}
```

Créer un objet avec le mot-clé new

Pour créer un objet, on utilisera le mot-clé `new` :

```
// On créé un objet à partir de la classe
$superObjet = new SuperClasse();

// On utilise une méthode de notre objet
$superObjet->afficheBonjour();
```

La pseudo-variable `$this`

La pseudo-variable `$this` permet d'accéder aux propriétés et méthodes de ce même objet :

```
class SuperClasse {
    // Déclaration d'une propriété
    public $toto = 'Valeur par défaut';

    // Déclaration des méthodes
    public function afficheBonjour() {
        echo $this->toto;
    }
}
```

Méthode constructeur

Il est possible de préciser une méthode constructeur en créant une méthode nommée `__construct()`. À chaque fois qu'un objet sera créé, cette méthode sera appelée :

```
class SuperClasse {
    // Quand l'objet est créé, cette méthode s'exécute :
    function __construct($param1, $param2) {
        echo $param1; // Affiche « Valeur 1 »
    }
}

$superObjet = new SuperClasse("Valeur 1", "Valeur 2");
```

Méthode destructeur

Il est possible de préciser une méthode destructeur en créant une méthode nommée `__destruct()`. À chaque fois qu'un objet est détruit, cette méthode sera appelée :

```
class SuperClasse {
    // Quand l'objet est détruit, cette méthode s'exécute :
    function __destruct() {
        echo 'Objet détruit';
    }
}
```

L'héritage avec le mot-clé extends

Il est possible de faire hériter les méthodes et propriétés d'une classe auprès d'une autre classe en utilisant le mot-clé « *extends* » dans la déclaration. Une classe ne peut pas hériter de plusieurs classes.

Les méthodes et propriétés hérités peuvent être surchargés en les redéclarant avec le même nom que dans la classe parente. En revanche, si une méthode a été définie dans la classe parente avec le mot-clé « *final* », il ne sera pas possible de surcharger celle-ci.

visibilité

Une méthode ou propriété peut être définie en préfixant sa déclaration avec le mot clé : *public*, *private* ou *protected*.

Visibilité publique

Les éléments déclarés comme publics peuvent être utilisés dans n'importe quelle partie du code.

Visibilité privée

Les éléments déclarés comme privés sont uniquement accessibles au sein de la classe dans laquelle ils ont été définis.

Visibilité protégée

Les éléments déclarés comme protégés sont accessible au sein de la classe dans laquelle ils ont été définis mais également au sein des classes qui en héritent.

Visibilité des propriétés

La visibilité des propriétés doit être obligatoirement définis en préfixant leur déclaration.

Visibilité des méthodes

La visibilité des méthodes est définie en préfixant leur déclaration. Si aucun mot-clé n'est défini au préalable, alors la méthode sera définie comme publique.