



PHP : Introduction

Culture générale

PHP est un langage de programmation coté serveur. En effet, PHP à la différence de JavaScript, sert à interagir avec le serveur, la base de donnée pour récupérer des informations, les réécrire, insérer ou même supprimer et les afficher coté **client** sur une page **Html**.

La version de PHP est la 7.4.0 mais évolue constamment (une version beta est déjà disponible v8.1.0)

Avec **PHP**, nous allons pouvoir **dynamiser notre Html** pour le rendre un peu plus intelligent, nous le verrons dans la création de notre projet commun : « Amazon from Scratch » Un site web présentant des produits ainsi qu'un BackOffice et des connexions entant qu'utilisateur / administrateur !

Par ailleurs, le créateur de **PHP** est **Rasmus Lerdorf en 1994**.

Il y a 5 ans, **PHP** en était encore à la version 5.3. Mais la version 7.0 était déjà disponible en **beta** (utilisable mais en cours de création et de test).

Les bases

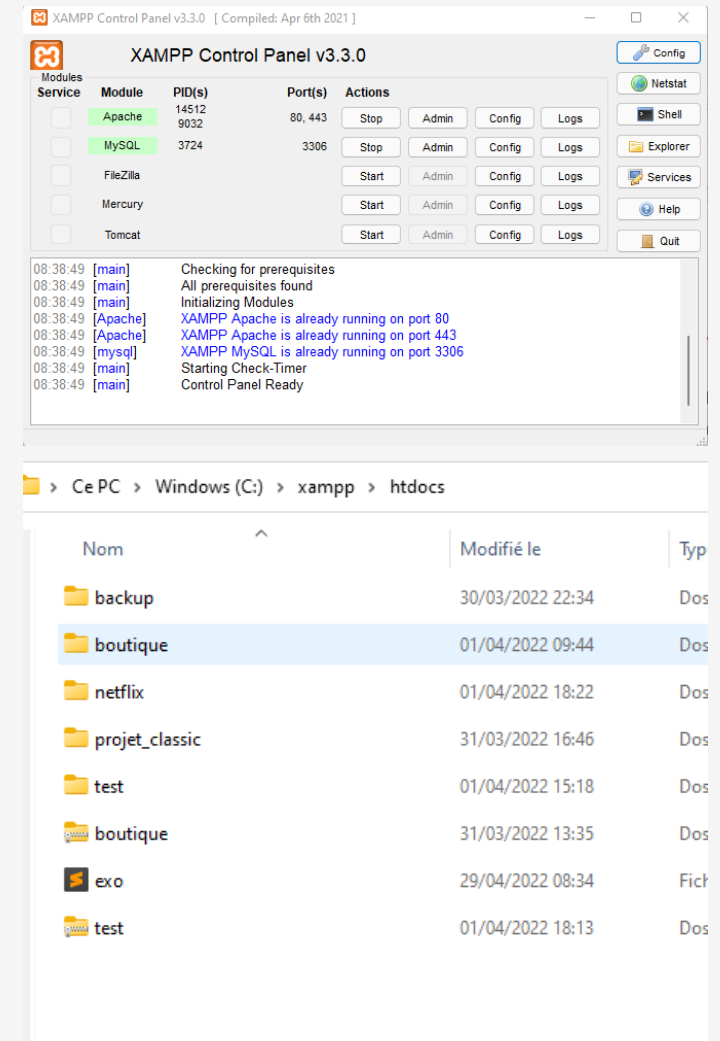
Pour travailler avec **PHP**, vous êtes dans l'obligation de travailler dans le dossier **HTDOCS** de votre serveur **APACHE** (Xampp / Lampp ect..).

Sans quoi, votre code ne sera perçu que comme du texte ou de l'html. Donc, direction :

C:\xampp\htdocs\votre_projet pour commencer à codé en **PHP** !

Parceque, oui, votre serveur **Apache** possède un compilateur caché dans le dossier **htdocs** servant à convertir le **code PHP** en code **HTML** ou en **Instruction Serveur**.

PHP reprend les bases de la programmation, création de **variable**, de **condition**, de **boucle**, de **fonction**, **programmation en orienté objet** ect.. Seul la syntax change ainsi que les fonctions natives de PHP nous donnant la possibilité de créer / modifier des fichiers sur le serveur ou interagir avec la base de donnée (chose impossible via JavaScript !)



Introduction : echo

Bien, essayons d'écrire quelque chose en **PHP** pour comprendre comment marche ce formidable langage.

PHP est un langage pouvant **créer ou render dynamique de l'Html**. Alors pour commencer affichons un texte.

Pour cela, nous avons besoin d'ouvrir une balise PHP :

`<?php`

, puis écrire une instruction qui affiche un message (comme sur le screen shot)
et enfin terminé le code par

`?>`

Ici, l'instruction qui permet d'afficher quelque chose en HTML via php est :
`echo "blabla";`

Veiller à ne jamais oublier le point-virgule à la fin, Php risqué de ne pas être content sinon

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
    <title></title>
</head>
<body>

    <h1>Mon Titre</h1>

    <p>Mon texte</p>

    <strong>Sous-titre en gras</strong>

    <?php

        echo "voici mon texte en PHP !";

    ?>

</body>
</html>
```

Introduction : echo

Ok, vous avez ajouté du texte grâce à “**echo**”, mais il est **possible de créer une balise html à la place d’un simple texte !**

Exemple :

```
<?php  
  
    echo "<p> Hello ! </p>";  
  
?>
```

Et oui, nous pouvons aussi bien mettre des balises Html dans nos “**echo**”, c’est l’utilité de **PHP après tout !**

Introduction : Variable

Nous avons vu comment créer une variable en JavaScript, vous découvrirez qu'en **PHP** la difference n'est pas énorme !

```
<?php  
  
$prenom = "Ragnar";  
$nom = "LeBreton";  
$age = 19;  
$mariage = TRUE;  
  
?>
```

En javascript, la syntax était :

```
var prenom = "Ragnar "  
var nom = "LeBreton";
```

En PHP, nous aurons besoin d'écrire le signe \$ avant le nom de la variable, à ne pas confondre avec JQuery !

Essayer !

Introduction : Variable

Vous avez la possibilité d'afficher le contenu d'une variable dans un ***echo*** !

Comme en programmation vous pouvez faire des calculs, programmes des instructions pour les déclencher à une action précise ect...

```
$nombreA = 29;  
$nombreB = 123;  
  
echo $nombreA + $nombreB;  
echo $nombreB / $nombreA;  
echo $nombreA + ( $nombreB * 19 );
```

```
<?php  
  
$prenom = "Ragnar";  
$nom = "LeBreton";  
$age = 19;  
$mariage = TRUE;  
  
echo "Bienvenue MR.".$nom;  
  
?>
```

Introduction : Condition

Il est impossible de programmer sans utiliser les conditions.

La syntax ne change toujours pas, nous retrouvons les termes : **IF / ELSE** et **ELSEIF** (SI , alors , Sinon)

N'oublions pas de terminer chaque instruction par un point-virgule !

```
if( $age < 18 ){  
    echo "vous êtes mineur. accès refusé !";  
}else{  
    echo "vous êtes majeur. Bienvenue";  
}  
  
if( $vehicule == "voiture" ){  
    echo "je conduis une voiture";  
}elseif ( $vehicule == "moto" ) {  
    echo "je conduis une moto";  
}else{  
    echo "j'ai des jambes";  
}
```


Introduction : Fonctions

Une fois encore, en JS nous avons l'habitude d'ajouter un **attribut** entre les parenthèses, en **PHP** nous procédons de la même façon, nous ajoutons un \$ avant le nom de l'attribut pour rappeler que c'est une variable.

Puis nous executons la fonction en appelant le nom de la fonction

Welcome('Gaston');

```
<body>

<?php

    function welcome( $client_name ){

        echo "Welcome Mister. ".$client_name;

    }

    welcome("Gaston");


    function ajout_bloc_rouge (){

        echo " <div> ALERTE </div> ";

    }

    ajout_bloc_rouge();

?>

</body>
```