Apprendre le PHP

PHP (PHP : Hypertext Preprocessor) est un langage de programmation de scripting généraliste. Sa principale vocation est l'envoi de réponse HTTP que se soit une page HTML générée en PHP ou des données JSON.

Le PHP permet l'utilisation des principes de la **POO** (héritage, polymorphisme, interface) mais n'oblige pas son utilisation comme Java ou dans une moindre mesure le JavaScript. **Comme tout langage de scripting PHP néccessite un interpréteur pour être executé.**

Documentations

Le manuel du langage PHP est disponible sur le site officel du langage : https://www.php.net/manual/fr/index.php

La réference des fonctions et objets du langage sont disponible via la barre de recherche du site officel : https://www.php.net/docs.php

La réference de la syntaxe du langage est dispoible ici : https://www.php.net/manual/fr/langref.php

Comme d'habitude **W3schools** est également là : https://www.w3schools.com/php/php_syntax.asp

Les possibilités du PHP

PHP est le premier langage de programmation du web et comme son nom l'indique PHP est un *Préprocesseur d'Hypertext*. Concrètement, un hypertext est un fichier HTML et PHP permet de d'exectuer un algorithme avant l'envoi de l'hypertext au client pour envoyer du contenu dynamique.

> Aujourd'hui on n'envoi pas execlusivement de l'hypertext (html) mais également du JSON ou du XML.

Avec PHP vous pourrez entre autre :

- **Créer un site web dynamique** sans l'utilisation de JavaScript.
- **Accéder à une base de données SQL**.
- Créer un **système d'autentification**.

- **Concevoir une API REST** accéssible depuis un front-end, par exemple en JavaScript via la méthode `fetch()` ou depuis n'importe quel client HTTP.
- Récupérer le contenu des champs d'un formulaire HTML.
- Gérer les **cookies**

Installation de l'intérpreteur PHP

L'intérpreteur php est le programme qui va executer nos scripts php.

Windows

- https://windows.php.net/downloads/releases/php-8.3.1-Win32-vs16-x64.zip
- Décompresser l'archive dans un dossier nommé php et placé le dans le dossier Programme de votre disque dur.
- Ajouter le chemin vers php.exe dans le PATH Windows
- Décommenté les lignes suivantes dans le fichier php.ini-developpement
  - extension=mysqli
  - extension=pdo_mysql
- Renommé le fichier php.ini
- Lancer le serveur web php

```
le/dossier/de/mon/serveur$ php -S localhost:8080
```

Linux

```
apt install php
le/dossier/de/mon/serveur$ php -S localhost:8080
```

Mac

```
brew install php
le/dossier/de/mon/serveur$ php -S localhost:8080
```

Lancez un serveur web compatible avec PHP

Une fois l'intérpréteur installé il faut lancer un serveur http local grâce à une simple commande disponible avec l'intérpréteur php. A la différence de serveurs HTTP lancés avec python, ce serveur local va executer les scripts php si un client demande un fichier `".php"` .

Dans une console :
Rendez vous dans le dossier dans lequel vous souhaitez mettre vos futurs sites web PHP et ouvrez une console à cet endroit.

Puis écrivez ceci pour lancer le serveur en `localhost` sur le port `8080` .

```
php - S localhost:8080
```

> Si vous souhaitez que votre serveur soit accéssible à tout les PC du réseau local de chez vous : remplacez `localhost` par l'adresse ipv4 de votre pc.

Apache2

Dans l'industrie on ne fournit pas un site PHP grâce à cette commande le plus souvent on utilise un serveur apache.

Hello World !

Tout code `php` doit être contenu dans un fichier .php et entre les balises php : `<?php ... ?>`

Dans une fichier **index.php**:

```php
<?php
echo "Hello World";
?>
```

`echo` est une fonction qui na pas besoin de parentèses et qui écrit un texte dans le HTML de la page à l'endroit où elle est placée.

```
<h1>Le magnifique site de <?php echo "Jeff"; ?> !</h1>
<p>
    <?php
        echo "Bienvenue tout le monde !";
    ?>
</p>
```

Avec la syntaxe alternative `<?= ?>` il est possible de ne pas écrire le `echo` .

```
<h1>Le magnifique site de <?= "Jeff" ?> !</h1>
<p>
    <?= "Bienvenue tout le monde !" ?>
</p>
```

Ces deux codes sont les mêmes.

Allez plus loin avec une variable

```
<?php
$prenom = "Jeff";
?>
<h1>Le magnifique site de <?php echo $prenom; ?> !</h1>
<p>
    <?php
        echo "Bienvenue tout le monde !";
    ?>
</p>
```

> **Précision sur `echo`**
> Pour être plus précis `echo` écrit enfaite dans le body de la réponse HTTP à envoyer au client. Et oui PHP est un langage back-end qui a pour but final d'envoyer une réponse HTTP ! Tout texte écrit en dehors des balises php sera placé dans le body de la réponse HTTP.

Syntaxe basique du PHP

En PHP toutes instructions se fini par un point virgule `;` .

Type de données

En php il existe plusieurs types de données :

- Integer, les nombres entiers
- Float, les nombres à virgules
- String, les textes
- Boolean, les valeurs binaires : `true` ou `false`
- Array, les tableaux
- Object, les instances de classes.
- NULL, la valeur d'une variable non déclarée.

A propos de la concaténation

En PHP la concaténation de deux string se fait via l'opérateur point : `.` .

```
$age = 24;
echo "J'ai ".$age." ans !";     // => J'ai 24 ans !
```

> **Note**
> Dans une string il est possible de placer directement une variable sans concaténation. N'oublie pas le dollar ! `$` .
>
> ```
> $age = 24;
> echo "J'ai $age ans !";
> ```

Les opérateurs en PHP

Les opérateurs en `PHP` respecte la syntaxe classique de tout les langages qui hérite du C.
Parmis les opérateurs les plus commun on retrouve :

Opérateur	Opération	Résultat
=	L'affectation	Une variable
.	La concaténation : `"J'ai ".$age." ans !"`	String
> < >= <= == !=	La comparaison	Boolean

Opérateur	Opération	Résultat
%	modulo	Integer
+ * - /	Arithmétique	Float ou Integer
++	Incrémentation	Float ou Integer
--	Décrémentation	Float ou Integer

Variables

Une variable est défini par un nom et un type de données parmis les types de données du PHP décrit plus haut.
Ont utilise le symbole dollar **$** pour déclarer une variable et l'opérateur **=** pour l'affectation.

```
$age = 24;                  // Nombre
$prenom = "Massinissa";     // String
$isMajeur = $age >= 18;     // Boolean
```

Pour utiliser une varible il faut toujours placer le dollar devant son nom.

```
echo $prenom;          // => Massinissa
echo gettype($age);    // => Integer
```

Les Tableaux

En php l'index d'un tableau s'appelle là `key` ou clé en français.
Il existe deux genres de tableaux : les listes et les `map` (ou dictionnaire) en français. La `key` d'une liste est un `Integer` alors que la `key` d'un dictonnaire est une `string` .

Les Array - Liste

```
$fruits = ["cerise","pomme","poire"];
echo $fruits[0];        // => cerise , la key 0 est un Integer.
```

Connaître la taille du tableau

La fonction `count()` fournit le nombre d'élément d'un tableau, c'est très utile pour écrire une boucle `for` par exemple.

```
count($fruits);         // 3
```

Ajouter

```
$fruits[] = "orange";   // Ajoute à la fin du tableau.
```

Modifier

```
echo $fruits[2];        // => poire
$fruits[2] = "banane";   // Ajoute à la fin du tableau.
echo $fruits[2];        // => banane
```

Supprimer des éléments

`array_splice` permet de supprimer des éléments dans un intervale donnée.

Définition :

```
array_splice(array $table,int startDeletingAt,int nbElementToDelete);
```

Paramètres :

- `array` le tableau à modifier
- `int` l'index à partir d'où commence la suppression des éléments
- `int` le nombre d'élément à supprimer

Exemple :

Supprimer l'élément à l'index 0 :

```
$fruits = ["cerise","pomme","poire"];
array_splice($fruits,0,1);      // Starts at 1 Ends at 0
echo $fruits[0];        // => pomme
```

Supprimer 2 éléments à partir de l'index 1 :

```php
$fruits = ["cerise","pomme","poire"];
array_splice($fruits,1,2);
echo $fruits[0];        // => cerise
echo $fruits[1];        // => NULL
echo $fruits[2];        // => NULL
```

Supprimer tout les éléments :

```php
$fruits = ["cerise","pomme","poire"];
array_splice($fruits,1);     // J'omets le nombre d'élément donc je supprime tout
echo count($fruits);        // 0
```

> Si ont omet le troisième paramètre, ont vide le tableau.

Les Array - Map

Les Map possède des clés textuelles et sont très utilisées pour récupérer les lignes d'une base de donnée SQL par exemple.

```php
// Array à key numérique
$eleve = [
    "name" => "Thomas",
    "lastname" => "Canal",
    "age" => 27,
];
echo $eleve["name"];       // => Thomas , la key "name" est une string.
```

Ajouter

```php
$fruits["job"] = "web dev";   // Ajoute à la clé job l'élément "web dev"
```

Modifier la valeur d'une clé

Modifier et ajouter une clé répond à la même syntaxe.

```php
$fruits["name"] = "Mathieu";
```

Supprimer une clé

La fonction `unset()` permet de supprimer une clé d'un `Map`, elle permet également de supprimer n'importe quelle variable.

```
unset($fruits["lastname"]);
```

> la fonction `var_dump()` permet d'afficher le détail d'une variable comme les éléments d'un tableau par exemple.
>
> ```
> var_dump($fruits);     // Ecrit le contenu du tableau dans le HTML
> ```

Les Conditions

Les conditions en PHP sont similaire au condition des autres langages de programmation.

if...else

```
if($age >= 18){
    echo "Majeur";
}
else{
    echo "Mineur";
}
```

L'opérateur ternaire ? :

L'opéretur ternaire un un raccourcis du if else qui permet de renvoyer une valeur si le test est vrai ou bien une autre si le test est fausse.

```
echo $age >=18 ? "Majeur" : "Mineur";    // Equivalent au code plus haut.
```

Syntaxe:

```
test ? valueIfTrue : valueIfFalse
```

switch

Le switch permet d'éviter d'écrire une longue suite de `if else if` imbriqués.
Ce code :

```php
$user = [
    "name" => "Thomas",
    "role" => "ADMIN"
];
if($user["role"]=="ADMIN"){
    echo "Welcome administrator ".$user["name"];
}elseif($user["role"]=="USER"){
    echo "Welcome ".$user["name"];
}else{
    echo "Welcome guest.";
}
```

Produit le même résultat que ce code :

```php
$user = [
    "name" => "Thomas",
    "role" => "ADMIN"
];
switch ($user["role"]) {
    case 'ADMIN':
        echo "Welcome administrator ".$user["name"];
    break;      // break défini la fin du case.

    case 'USER':
        echo "Welcome ".$user["name"];
    break;

    default:
        echo "Welcome guest.";
    break;
}
```

Les boucles

La boucle while

```
while (test) {

}
```

La boucle for

Le PHP supporte la synxtaxe classique du `for` :

```
$fruits = ["cerise","pomme","poire"];
for ($i=0; $i < count($fruits); $i++) {
    echo $fruits[$i];
}
/**
 * cerise
 * pomme
 * poire
*/
```

Le foreach

La plupart du temps la syntaxe classique n'est pas utilisée car le `for` permet surtout de parcourir un tableau, pour ceci ont préfère utiliser `foreach` .

```
$fruits = ["cerise","pomme","poire"];
foreach($fruits as $fruit){
    echo $fruit;        // J'affiche l'itérateur $fruit
}
/**
 * cerise
 * pomme
 * poire
*/
```

foreach - Utiliser la clé d'un tableau Map

Dans le cas où mon tableau est un `Map` je veux pouvoir récupérer la clé associée aux valeurs, j'utilise la syntaxe complète du `foreach` .

```php
$eleve = [
    "name" => "Thomas",
    "lastname" => "Canal",
    "age" => 27,
];
foreach($eleve as $key => $value){
    echo $key." : ".$value;
}
/**
 * name : Thomas
 * lastname : Canal
 * age : 27
 * */
```

Les fonctions en PHP

Les fonctions sont des blocs de code paramètrables et réutilisables qui peuvent renvoyer une valeur de retour. **Une fonction est défini par son nom, ses paramètres et le type de sa valeur de retour.**

> La valeur de retour d'une fonction qui ne renvoi rien est `NULL` .

Syntaxe :

```php
function functionName(type param1, type param2, ...) : returnType{
    // Code
    return value;
}
```

Exemple simple :

```php
function Somme(float $a,float $b) : float{
    return $a+$b;
}
```

Exemple réel :

```php
/***
 * Converti un tableau en string
 * @param array $table : le tableau à convertir en string
 * @param string $separator : séparateur utilisé entre chaque élément du tableau, par de
 * @return string : la string qui contient tout les éléments du tableau.
 */
function arrayToString(array $table,string $separator = ",") : string{
    $joinedString = "";

    foreach($table as $key=>$value){
        $joinedString = $key==0 ? $joinedString.$value :$joinedString.$separator.$value;
    }

    return $joinedString;
}

echo arrayToString(["J'ai",24,"ans"]," ");      // => J'ai 24 ans
```

En PHP si l'ont précise le type d'un paramètre ou d'une valeur de retour et qu'une variable du mauvais type est passée une erreur survient et le script s'arrète.

> Note : `string $separator = ","`
> Le paramètre d'une fonction peut avoir une valeur par défaut, ce qui rend le paramètre optionnel.

Les fonctions lambda

En PHP il est possible de créer des fonctions sans nom, ces fonctions sont souvent utilisées en tant que fonctions callback. Ont les appelles les fonctions lambda.

Syntaxe :

```php
function(type param):returnType{
    return value;
}
```

Exemple :

```php
$somme = function Somme(float $a,float $b) : float{
    return $a+$b;
}
```

Les fonctions lambda sont utilisés dans l'appel des fonctions de traitement d'un array comme :

- `array_filter()` : https://www.php.net/manual/fr/function.array-filter.php
- `array_map()` : https://www.php.net/manual/fr/function.array-map.php
- `array_reduce()` : https://www.php.net/manual/fr/function.array-reduce.php

POO - Orienté objet en PHP

Le PHP permet l'utilisation de classes, interfaces, héritage et polymorphisme.

Déclarer une classe

La déclaration d'une classe se fait via le mot clé `class` .

Syntaxe :

```
class ClassName{

    public $attribut1;
    public $attribut2;
    private $attribut3;

    function __construct(type $param){
        $this->attribut1 = $param;
    }
    public function publicMethodName(){

    }
    public function privateMethodName(){

    }
}
```

Exemple :

```php
class User{
    private $name;
    public function __construct(string $name){
        $this->name = $name;
    }
    getName():string{
        return $this->name;
    }
    setName(string $newName){
        $this->name = $newName;
    }
}
```

Instancier un objet

Le mot clé `new` permet d'instancier une classe.

```php
$user = new User("Massinissa");
```

Importer des éléments entre plusieurs fichiers PHP

Il est possible de déclarer des fonctions, des constantes ou des classes dans un ficher php puis d'y accéder dans un autre. Cela permet d'organiser son projet.

Il y a deux moyens d'importer des éléments :

- **via la fonction `require_once()`** qui copie colle TOUT le contenu d'un fichier php dans le fichier dans lequel il est appellé.
- **via le gestionnaire de paquet `Composer`** , cette option est l'objet d'un cours à part. `Composer` est primoridal dans l'utilisation de modules tiers et de framework comme `Symfony` ou `Laravel` .

Import avec `require_once()`

J'écris du code dans un fichier php.
other.php

```php
<?php
$fruit = "pomme";

class User{
    public $name;
    public function __construct(string $name){
        $this->name = $name;
    }
}

function hello(){
    echo "Hello";
}
```

Je l'importe dans un autre.

```php
<?php
require_once("other.php");

$user = new User("Massinissa");
echo $fruit;
hello();
```

Lorsque j'importe j'effectue liéttéralement un copié-collé du script PHP à l'endroit de l'appel du `require_once` . Attention donc à rester organiser et ne pas importer n'importe quoi.

Session PHP

En PHP on peut sauvegarder des données pendant toute la session de l'utilisateur, de cette manière même si l'utilisateur charge de nouvelles pages et donc de nouveaux scripts php les données persistes.

Une utilisation typique des sessions c'est l'autentification d'un utilisateur.
On démarre la session et on peuple les clées de la variable `$_SESSION` avec `session_start()` et ont supprime les variables de la session avec `session_destroy()` .

> Attention il ne faut appeller `session_start()` qu'une fois par fichier au tout début.
> **Syntaxe :**

```php
session_start();                // Je lance la session
$_SESSION["userid"] = 1;        // Je créer une variable de session dans le tableau de s
$_SESSION["role"] = "ADMIN";
```

Exemple - Autentification

Le header "Location: url" permet de rediriger la page vers une autre page.

> ATTENTION IL FAUT COLLER `Location` et le `:` `"Location: autrepage.php"` sinon ça ne marche pas.

index.php

```php
<form action="/profil.php" method="post">
    <input type="email" name="email">
    <input type="password" name="password">
    <input type="submit" value="Se Connecter">
</form>
```

profil.php

```php
<?php
session_start();
$email = "good@mail.com";
$password = "0000";
if($_POST["email"] == $email && $_POST["password"] == $password){
    $_SESSION["user_id"] = 3;       // Habituellement ont récupére cet id d'une BDD.
}

if(!$_SESSION["user_id"]){
    header("Location: index.php");  // redirige vers le formulaire
}
?>
<h1>Bienvenue !</h1>
<a href="logout.php">Se déconnecter</a>
```

logout.php

```php
<?php
session_destroy();

header("Location:index.php");
?>
```