

Introduction au **PHP** : les basiques





Quelques mots...

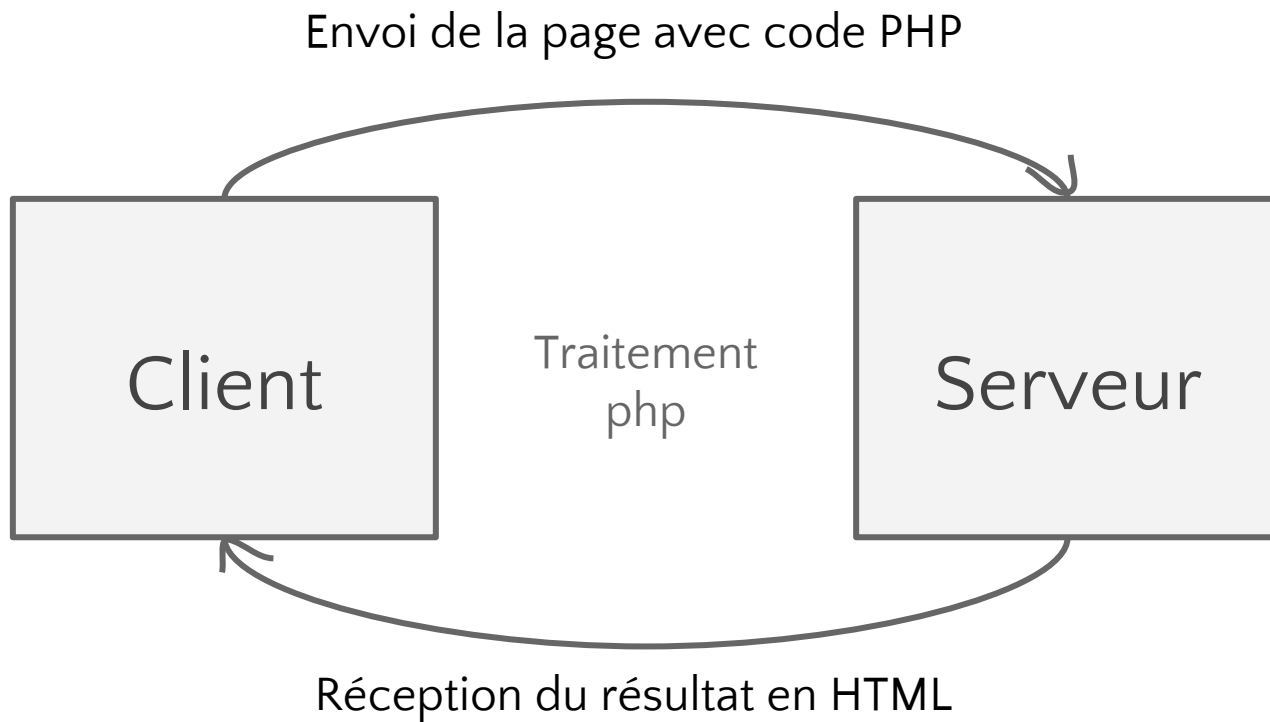
Inventé par Rasmus Lerdorf en 1994: voulait faire le suivi des visiteurs sur son site web.

PHP: Hypertext Preprocessor, c'est une solution Open Source.

À l'heure d'aujourd'hui, la dernière release est PHP 7.2.2



Côté serveur





Initialisation

Il faut initialiser chaque morceau de code contenant du PHP avec des balises spécifiques :

```
<?php
    $mystring = "12";
    $myinteger = 20;
?>
```



Initialisation

Vous pouvez inclure des fichiers PHP externes.

```
<?php
    include("monfichier.php");
?>
```



Types de données

String	« abc »	« Vive le PHP »
Integer	30	-20
Float	4.2	1.00023
Boolean	true	false
Array	variable spéciale	
Object	variable + complexe	



Forcer un type

PHP va automatiquement convertir les types des données autant que nécessaire, mais si besoin vous pouvez spécifier un type précis pour passer au dessus du type défini par défaut avec PHP.

```
<?php
    $myString = "Vive le pâté";
    $myInteger = (integer)$myString;
    var_dump($myInteger)
?>
```



isset()

isset() sert à vérifier si une variable est bien définie or not.
Très utilisé pour la suite de vos projets. Permet également d'éviter les warning avec PHP !

```
$petiteVar = 1;  
if (isset($petiteVar)) {  
    echo "Définie \n";  
} else {  
    echo "Pas définie... \n »;  
}
```




NULL / empty



0 vs NULL



Constante

Une constante est une variable qui restera inchangée car elle ne peut être modifiée plus tard dans le code, contrairement aux variables.

Une constante prend deux paramètres :
un nom et une valeur.

```
<?php
    define('YEAR', '2014');
    define("CURRENT_TIME", time());
?>
```



Superglobales

<code>\$_GLOBALS</code>	Toutes les variables globales dont les variables d'environnements
<code>\$_GET</code>	Envoyées avec une requête GET
<code>\$_POST</code>	Envoyées avec une requête POST
<code>\$_FILES</code>	Envoyées avec une requêtes pour l'upload de fichiers
<code>\$_COOKIE</code>	Stocke les variables de cookies
<code>\$_SESSION</code>	Stocke les variables de sessions
<code>\$_ENV</code>	Stocke les variables d'environnements



Comparaisons

$\$a == \b	Égaux
$\$a === \b	Identiques
$\$a != \b	Pas égaux
$\$a <> \b	Pas égaux
$\$a !== \b	Pas identiques
$\$a < \b	Plus petit
$\$a > \b	Plus grand
$\$a <= \b	Plus petit ou égal
$\$a >= \b	Plus grand ou égal



Fonction `var_dump()`

`var_dump()` est un bon moyen de voir ce qu'il y a à l'intérieur d'une variable. C'est un bon réflexe à prendre en main pour debugger notamment !

```
<?php
    $capitalcities['England'] =
    array("Capital"=>"London",
    "Population"=>40000000,
    "NationalSport"=>"Cricket");
    var_dump($capitalcities);
?>
```



Conditions

```
<?php
    if ($a > $b){
        echo "a est plus grand que b ";
    }
?>
```



Conditions

```
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";  
}
```



Boucles : for

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




Boucles : while

```
<?php
$nombre_de_lignes = 1;

while ($nombre_de_lignes <= 100){
    echo $nombre_de_lignes;
    $nombre_de_lignes++; // $nombre_de_lignes = $nombre_de_lignes + 1
}
?>
```

Il faut TOUJOURS s'assurer que la condition sera fausse au moins une fois. Si elle ne l'est jamais, alors la boucle s'exécutera à l'infini !



Boucles : foreach

Pratique pour parcourir un tableau. On affichera les éléments d'un tableau ainsi que les clefs correspondantes comme ça :

```
<?php
foreach ($tableau as $cle => $valeur) {
    echo $cle.' - '.$valeur.'<br />'. "\n";
}
?>
```



Boucles : foreach

Admettons qu'on ne veuille que les valeurs, alors on peut aussi faire comme ça :

```
<?php
foreach ($tableau as $valeur) {
    echo $valeur.'<br />'. "\n";
}
?>
```



Fonction utilisateur

```
<?php
    function sayHello($name) {
        return "Hello $name";
    }
    print sayHello("Maxime");
?>
```



Paramètres par défaut

```
<?php
    function coucou($name = "Maxime") {
        return "Hey $name!\n";
    }
    coucou();
    coucou("Morgan");
    coucou("Enzo");
?>
```



Ranger des tableaux

sort()	Tri les éléments dans l'ordre alphabétique - Réécrit les clés
rsort()	Tri les éléments dans l'ordre inverse - Réécrit les clés
asort()	Tri les éléments dans l'ordre alphabétique - Garde les clés
arsort()	Tri les éléments dans l'ordre inverse - Garde les clés
ksort()	Tri les clés dans l'ordre alphabétique
krsort()	Tri les clés dans l'ordre inverse



Ranger des tableaux

```
array(3) {  
    ["a"]=>  
    string(6) "Pommes"  
    ["b"]=>  
    string(7) "Oranges"  
    ["c"]=>  
    string(5) "Paires"  
}
```



Tableaux associatifs

```
<?php
    $myarray = array("a"=>"Pommes", "b"=>"Oranges",
        "c"=>"Poires");
    var_dump($myarray);
?>
```

On appelle tableau associatif la relation clé - valeur. Vous pouvez spécifier à chaque fois un nom de clé spécifique (sauf de type float).



Tableaux multidimensionnels

```
<?php
    $capitalcities['England'] = array("Capital"=>"London",
    "Population"=>40000000, "NationalSport"=>"Cricket");
    $capitalcities['France'] = array("Capital"=>"Paris",
    "Population"=>2440000, "NationalSport"=>"Football");
?>
```



Tableaux multidimensionnels

```
array(2) {  
    ["England"]=>  
    array(3) {  
        ["Capital"]=>  
        string(6) "London"  
        ["Population"]=>  
        int(40000000)  
        ["NationalSport"]=>  
        string(7) "Cricket"  
    }  
}
```

```
    ["France"]=>  
    array(3) {  
        ["Capital"]=>  
        string(7) "Paris"  
        ["Population"]=>  
        int(2440000)  
        ["NationalSport"]=>  
        string(5) "Football"  
    }  
}
```



Sauvegarder un array

```
$array["a"] = "Oh";  
$array["b"] = "Ha";  
$array["c"] = "Hey";  
// serialize un array, permet son enregistrement dans un  
fichier par exemple  
$str = serialize($array);  
// encode pour le convertir en une version safe pour le  
passer sur le web  
$strenc = urlencode($str);
```



Sauvegarder un array

```
// decode  
$arr = unserialize(urldecode($strenc));  
var_dump($arr);
```



Afficher les erreurs en local

```
// Afficher les erreurs à l'écran
ini_set('display_errors', 1);

// Enregistrer les erreurs dans un fichier de log
ini_set('log_errors', 1);

// Nom du fichier qui enregistre les logs (attention
aux droits à l'écriture)
ini_set('error_log', dirname(__file__) . '/
log_error_php.txt');
```



Quelques ressources

- ◉ [Hacking with PHP](#)
- ◉ La Bible de PHP => [PHP.net](#)
- ◉ [TehPlayground](#)
- ◉ Sololearn - [Learn PHP](#)