

# PHP

---

## A quoi sert-il ?

---

C'est un langage interprété qui sert principalement à générer des documents (HTML, PDF, images, ...)

## Comment utiliser php

---

1. En console, avec la commande `php fichier.php`
2. Avec un serveur web (Apache, NGINX, ..)

## PHP est interprété

---

PHP interprète uniquement ce qui se trouve entre les balises ouvrante `<?php` et fermante `?>`

## Hello world en PHP

---

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

## PHP en console

---

Pour tester vos exercices, vous devez exécuter vos fichiers PHP. Sachez que +80% des sites internet utilisent PHP, malgré cela, nous pouvons exécuter les fichiers en console, et c'est par là que nous commencerons.

Exemple, si votre répertoire de travail est `php_level_0` (vous pouvez le vérifier avec la commande `pwd` )

Vous devrez exécuter la commande suivante

```
php exo00/01_symbol.php
```

# Votre premier fichier PHP

---

Tout script PHP doit se trouver entre les balises ouvrantes et fermantes, déclarant que du PHP s'y trouve.

```
<?php
// Bonjour je suis un commentaire
?>
```

Pour éclaircir immédiatement l'histoire, sachez que vous pouvez y mettre du texte de tout type (en dehors des balises). Ce texte ne sera pas interprété par le moteur PHP.

Exemple:

```
Hello<?php
// Bonjour je suis un commentaire
?>
World
Test
```

Le résultat de l'exécution de ce fichier sera :

```
HelloWorld
Test
```

## Comment écrire dans le flux de sortie en PHP

---

### Le flux de sortie c'est quoi ?

Si on parle d'exécuter des fichiers en console, cela signifie afficher/écrire dans la console

On peut le faire avec différentes fonctions, en général vous utiliserez :

- echo
- print
- printf
- sprintf

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

Vous constaterez que toutes les instructions PHP se terminent pas ;

## L'avantage de PHP

---

- Fonctionne sur tous les OS
- Simple à prendre en main
- Syntaxe familière
- Vieux, beaucoup de doc en ligne

## La bonne doc

---

- [PHP.net](https://www.php.net)
- [W3School](https://www.w3schools.com/php/)
- [OpenClassRoom](https://www.openclassroom.com/fr/)

## Les variables en PHP

---

Les variables sont dynamiques (elles sont non-typées et peuvent changer de type en cours d'exécution du script).

Elles sont obligatoirement déclarées et utilisées avec le symbole \$

Exemple:

```
<?php
$number = 5;
$hello = 'World';

// Les chaines de caracteres avec des double guillemets permettent l'interpolation de '
echo "Hello $hello, $number";
// Vous pouvez également concatener vos variables et chaines de caractères:
echo "Hello " . $hello . ", " . $number;
// Vous remarquerez que depuis le début de ce fichier nous avons utiliser les guilleme

// Essayez donc les 2 lignes précédentes avec des guillemets simples
```

```
echo 'Hello $hello, $number';  
echo 'Hello ' . $hello . ', ' . $number;  
?>
```

## Explorer le contenu des variables (pour DEBUG)

---

Parfois vous ne savez pas ce que contient une variable, le meilleur moyen d'en savoir plus, ou de débiter votre script est d'utiliser ces fonctions:

- var\_dump
- print\_r

```
<?php  
$maVar = "test";  
var_dump($maVar);  
  
$maVar = 42;  
var_dump($maVar);  
  
$maVar = array("Bonjour", 'Hello');  
var_dump($maVar);  
  
$maVar = false;  
var_dump($maVar);  
/*  
- Pareil avec print_r  
- Vous pouvez passer plusieurs variables à var_dump($var1, $var2, ...)  
*/  
?>
```

## Les constantes en PHP

---

La première constante que nous utiliserons sera `PHP_EOL` (pour PHP\_end of line).

Comment faire si je vous demande d'écrire

```
Hello  
World
```

avec un seul `echo` ?

# Les conditions

---

Les conditions (ou structure conditionnelles) sont assez simple à réaliser:

```
<?php

if (true != false) {
    // bonjour on arrivera tout le temps ici
}
elseif (false != false) {
    // nous ne passerons jamais ici
}
else {
    // test
}

// le raccourci suivant est disponible
if (true != false)
    echo 'Helloo';

// Vous pouvez coupler les conditions avec des 'ET' et 'OU'
if (1 < 2 && 2 < 3) {
    // 1 est plus petit que 2 et 2 est plus petit que 3
}

if (10 < 100 AND 100 < 1000) {
    // fonctionne également
}

?>
```

# Les tableaux

---

## Les tableaux 'simple'

```
<?php

$arr_a = [1, 2, 3, 4, 5];
$arr_b = array(1, 2, 3, 4, 5);

echo $arr_a[0];
echo $arr_b[4];
echo 'Longueur du tableau A = ' . count($arr_a);
```

?>

## Les tableaux 'associatifs'

```
<?php

$arr_as = [
    'pomme' => 10,
    'test' => 20
];

$arr_bs = array(
    'pomme' => 10,
    'test' => 20
);

echo $arr_as['pomme'];
echo $arr_bs['test'];
echo 'Longueur du tableau AS = ' . count($arr_as);

?>
```

## S'assurer que la clé existe / contient une valeur dans le tableau

---

```
<?php
$arr_as = [
    'pomme' => 10,
    'test' => 20
];

if (!isset($arr_as['poire'])) {
    echo 'Tableau ne contient pas de poire';
}
else {
    echo 'Poire(s) : ' . $arr_as['poire'];
}

?>
```

La fonction `empty` existe également, vous êtes invité à lire la documentation à ce sujet

- [empty](#)
- [isset](#)

