

# Le PHP

## 0. Présentation

---

Le **PHP**, acronyme récursif pour **PHP Hypertext Preprocessor** est un langage de scripts spécialement conçu pour le **développement d'applications web**. Il peut être intégré facilement au HTML.

Le langage PHP a été créé en 1994 par **Rasmus Lerdorf**, un programmeur groenlandais et canadien pour son site web personnel.

### Un exemple de code PHP

```
<!doctype html>
<html>
  <head>
    <title>    Exemple de variables    </title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1> Un premier programme en PHP : pour les tests</h1>

    <?php
      // Commentaires en php
      // Utilisation de variables, avec le dollard $
      $var1=5;
      $var2=7;
      $somme=$var1+$var2;
      // echo permet d'afficher sur la page
      // On peut mêler du texte et des variables
      echo "<p>La somme de $var1 et $var2 est $somme.</p>";
      $nombre1=rand(1,100); // rand(a,b) génère un entier aléatoire en
tre a et b (inclus)
      echo "<p>Le 1er nombre choisi au hasard est $nombre1.</p>";
      $nombre2=rand(1,100);
      echo "<p>Le 2e nombre choisi au hasard est $nombre2.</p>";
      $somme=$nombre1+$nombre2;
      // On affiche la somme
      echo "<p>La somme de $nombre1 et $nombre2 est $somme.</p>";
    ?>
  </body></html>
```

Ce petit exemple en PHP calcule la somme de deux variables, \$var1=5 et \$var2=7 ; qui affiche la somme avec la fonction avec la fonction `echo` ; et qui calcule et affiche la somme de deux entiers choisis aléatoirement entre 1 et 100, avec la fonction `rand(1 , 100)`.

- **Visualiser le résultat**

Un lien vers le fichier hébergé sur un serveur distant : [PHP-Exemple1.php](https://math93.com/images/pdf/NSI/premiere/html-premiere/docphp/PHP-Exemple1.php).  
<https://math93.com/images/pdf/NSI/premiere/html-premiere/docphp/PHP-Exemple1.php>

### Premières remarques sur le PHP

- L'affichage en PHP utilise la fonction `echo " texte et $variables "`. On peut afficher du texte et des variables \$var simultanément très simplement.
- Les lignes se terminent par un point virgule `;`.
- Les variables se définissent avec le dollard `$var1` et peuvent s'intégrer facilement au texte.

### Petit exercice important

1. **Affichez le code source** (Click droit) de la page [PHP-Exemple1.php](#).
2. Vous remarquez que le code source propose juste l'affichage, **sans les lignes de code PHP**.
3. **Actualisez la page**, on voit que le contenu change car la fonction `rand(1 , 100)` est appelée.
4. C'est magique, le client, c'est à dire notre ordinateur ne s'aperçoit pas qu'il y a un calcul d'effectué.
5. Recopier le code dans replit.com en choisissant le langage de programmation PHP seueur.

## Un fonctionnement distant

- Le **PHP** fonctionne en fait sur **serveur distant**.  
On ne peut pas avoir un fonctionnement immédiat sur notre disque comme avec le **HTML**, le **CSS** ou le **JavaScript**.
- Ce qui distingue **PHP** des langages de script comme le **JavaScript**, est que **le code est exécuté sur le serveur**, générant ainsi le **HTML**, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques.
- le serveur doit donc être capable de repérer les parties de codes **PHP**, cela se fera grâce à des **balises**.  
Le **code PHP** est donc inclus entre une balise de début `<?php` et une balise de fin `?>` qui permettent au **serveur web** de passer en mode **PHP**, ce que ne fait pas votre navigateur sur votre disque dur.



Vous pouvez maintenant tranquillement lire cette page qui expose le fonctionnement du PHP **Openclassrooms** : [Fonctionnement d'un site écrit en PHP](#)

## 1. Programmer en PHP

### Quelques remarques sur les fonctions, les structures conditionnels en PHP

- Les **fonctions en PHP** fonctionnent comme en Javascript. Le mot clé `function` permet de définir une fonction.
- Les paramètres sont écrits entre parenthèses précédés du symbole `$` comme toute les variables.
- Le mot clé `return` permet de renvoyer une valeur.
- Il existe de nombreuses **fonctions intégrées en PHP** vous pouvez consulter ces sites de références au fur et à mesure des besoins.
  - Liste des [fonctions PHP](#) ;
  - Un tutoriel : [https://www.w3schools.com/php/php\\_functions.asp](https://www.w3schools.com/php/php_functions.asp)
  - Un cours sur Openclassrooms : <https://openclassrooms.com>
- De même les **structures conditionnelles** `if` et `if else` fonctionnent comment en **JavaScript**.
- Par exemple cette fonction renvoie le maximum des deux nombres

```
<?php
function maximum($a,$b)
{
if ($a>$b)
return $a;
else
return $b;
}
?>
```

- De même les **structures itératives, les boucles** `for` et `while` fonctionnent comment en **JavaScript**.

## Un exemple

L'exemple ci-dessous se sert d'une boucle for pour afficher un tableau à l'écran avec une boucles `for` et une boucle `while` . C'est magique.

```
<head>
  <title>
    PHP - Exemple 2
  </title>
  <meta charset="utf-8"/>
</head>
<body>
<h1> PHP - Exemple 2 </h1>
<table border="1">
  <tr>
    <?php
    for($i=1;$i<=10;$i=$i+1) // ou ($i=1;$i<=10;$i++)
    {
      echo "<td>$i</td>";
    }
    ?>
  </tr>
</table>

<p> </p>

<table border="2">
  <tr>
    <?php
    $i = 1;
    while ($i <= 10)
    {
      echo "<td>$i</td>";
      $i=$i+1;
    }
    ?>
  </tr>
</table>

</body>
</html>
```

- Lien vers le fichier `.php` sur le serveur : [PHP-exemple2](#) / [Code source](#).

### Exercice 1

La fonction `rand(a,b)` renvoie un entier aléatoire compris entre a et b.

1. Ecrire un script qui choisit aléatoirement un nombre entre 1 et 15.
2. Et qui affiche le table de multiplication de ce nombre dans un tableau.

*Toute amélioration est la bienvenue.*

### Exercice 2

La fonction `rand(a,b)` renvoie un entier aléatoire compris entre a et b.

Ecrire un script en php qui :

1. choisit 100 nombres aléatoires entre 0 et 100 ;
2. les affiche ;
3. et affiche leur moyenne.

*Toute amélioration est la bienvenue.*

## 2. Interagir avec une page en PHP : les méthodes GET et POST

Un script PHP s'exécute sur le serveur lorsque l'utilisateur accède à un fichier par un lien. Ce script effectue un certain nombre d'instructions pour produire la page web demandée et s'arrête lorsque celle-ci est produite.

Comment faisons-nous pour passer une information d'une page à l'autre ? De l'utilisateur à la page .PHP ?

Par exemple, si je saisi mon compte utilisateur sur un site et qu'en cliquant je suis conduit vers la page de mon compte, comment cette page se souvient-elle du nom que j'ai écrit dans la page précédente ?

Il existe plusieurs solutions pour transmettre des informations d'une page aux autres :

- Cookies ;
- base de données ;
- fichiers textes ;
- la méthode GET ;
- la méthode POST.

Nous allons étudier les **méthodes GET et POST**.

## 2a. La méthode GET

### Interagir avec une page en PHP avec la méthode GET

#### 2a1. Former une URL pour envoyer des paramètres

Imaginons que votre site s'appelle **math93.com** et que vous avez une page PHP intitulée **bonjour.php**. Pour accéder à cette page, vous devez aller à l'URL suivante :

<https://www.dimension-k.com/maths/NSI/Bonjour.php>

Pour envoyer des informations à la page **bonjour.php**, on va ajouter des informations à la fin de l'URL, comme ceci :

- <https://www.dimension-k.com/maths/NSI/Bonjour.php?nom=Kergot&prenom=Julien>
- On écrit les **variables** (« nom », « prenom ») et les **valeurs** des variables (« Kergot », « Julien ») après un point d'interrogation `?`.
- On sépare les variables par le symbole `&`.
- La seule limite est la longueur de l'URL. En général il n'est pas conseillé de dépasser les **256 caractères**.

#### 2a2. Récupérer les valeurs des variables

Nous avons fait un lien vers le fichier php **bonjour.php** et à deux variables **nom et prenom**.

**bonjour.php?nom=Galois&prenom=Evariste**

Pour récupérer les valeurs des variables, on y accède via la variable `$_GET`.

- `$_GET['nom']` . à la valeur **Galois** ;
- `$_GET['prenom']` . à la valeur **Evariste** ;

Essayer de changer les noms et prénoms dans l'adresse URL.

```
<head>
  <title>
    PHP - Exemple 3
  </title>
  <meta charset="utf-8"/>
</head>
<body>
<h1> PHP - Exemple 3 </h1>
<?php
// les variables existent-t-elles ?
if(isset($_GET['nom']) and isset($_GET['prenom']))
{// oui les variables existent
    $lenom=$_GET['nom'];
    $leprenom=$_GET['prenom'];
    //echo "<p>Bonjour $_GET['prenom'] et $_GET['nom'] !</p>" ;
    echo "<p>Bonjour $leprenom $lenom</p>" ;
}
else
{
    echo "<p>Bonjour Madame ou Monsieur.</p>" ;
}
?>
</body>
</html>
```

- **Remarque** : `isset(var)` - Détermine si la variable `var` est déclarée et est différente de NULL.

#### Exercice 3

1. Ecrire une page qui prend deux nombres dans l'URL et affiche leur somme.
2. Modifier la page pour donner trois paramètres dans l'URL, deux nombres et une opération. Le résultat de l'opération sera affiché dans la page.

*Toute amélioration est la bienvenue.*

#### Exercice 4

1. Ecrire deux pages, la première choisissant un nombre au hasard entre -10 et 10

2. et contenant un lien vers la seconde qui affichera la racine carrée du nombre choisi, si cela est possible.

*Toute amélioration est la bienvenue.*

## 2a. La méthode POST

### Interagir avec une page en PHP avec la méthode POST

Il existe une autre méthode pour passer des données d'une page à l'autre sans mentionner les valeurs dans l'URL.

Il s'agit de la méthode utilisée dans les formulaires

#### Un exemple

Reprenons l'exemple 3 avec le nom et le prénom.

1. On va donc écrire deux pages, la première une page HTML que l'on va nommer ex4-HTML.html, et la deuxième une page PHP l'on va nommer ex4-PHP.php
2. Afin de pouvoir accéder aux contenus envoyés par la méthode POST, il faut donner à chaque élément dont on veut récupérer les valeurs l'attribut `name` .
3. Pour récupérer les valeurs des variables, on y accède via la variable `$_POST`.
  - `$_POST['nom']` . à la valeur du nom entré ;
  - `$_POST['prenom']` . à la valeur du prénom entré ;

```
<!doctype html>
<html>
<head>
  <title>
    PHP - Exemple 4 - Méthode POST Fichier HTML
  </title>
  <meta charset="utf-8"/>
</head>
<body>
<h1> PHP - Exemple 4 - Méthode POST Fichier HTML</h1>
<form action="ex4-PHP.php" method="post">
```

```
<label for="nom">Votre Nom :</label>
<input type="text" name="nom">
<div> et </div>
<label for="prenom">Votre Prénom :</label>
<input type="text" name="prenom">
<input type="submit" value="Valider">
</form>
</body>
</html>
```

```
<!doctype html>
<html>
<head>
  <title>
    PHP - Exemple 4 - Méthode POST Fichier PHP
  </title>
  <meta charset="utf-8"/>
</head>
<body>
<h1> PHP - Exemple 4 - Méthode POST Fichier PHP </h1>
<?php
// les variables existent-t-elles ?
if(isset($_POST['nom']) and isset($_POST['prenom']))
{ // oui les variables existent
  $lenom=$_POST['nom'];
  $leprenom=$_POST['prenom'];
  //echo "<p>Bonjour $_POST['prenom'] et $_POST['nom'] !</p>" ;
  echo "<p>Bonjour $leprenom $lenom</p>" ;
}
else
{
  echo "<p>Bonjour Madame ou Monsieur.</p>" ;
}
?>
</body>
```

- Lien vers le fichier `.html` sur le serveur : [ex4-HTML.html](#) .
- **Remarque** : Tous les paramètres de `input` : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>

## Exercice 5

Reprendre les exercices 3 et 4 mais en utilisant la méthode POST et un formulaire.

*Toute amélioration est la bienvenue.*

## Exercice 6 : un QCM

1. Ecrire deux pages, la première contenant un QCM avec au moins trois questions.

2. Les réponses seront proposées et l'utilisateur devra faire son choix par des cases à cocher.

3. La seconde page affichera le score obtenu.

Aide : Pour récupérer l'information "case cochée, c'est assez simple : la variable n'existe dans le post que si la case a été cochée.

*Toute amélioration est la bienvenue.*

**Aide :**

- <https://www.zerocms.fr/PHP/formulaires-POST-url-GET-PHP.php>
- [Check-box et récupération 1](#)
- <https://apcpedagogie.com/recuperer-les-valeurs-des-checkbox-avec-php/>