

Cours 3 : Écrivez votre premier script

Maintenant, nous allons passer à la pratique et réaliser notre toute première page web en PHP !

Ne vous attendez pas à un résultat extraordinaire, mais cela va vous permettre de prendre vos marques. Vous allez en particulier comprendre comment on sépare le code HTML classique du code PHP.

Vous êtes prêt ? Allons-y !

Utilisez des balises PHP

Vous savez que le code source d'une page HTML est constitué de **balises** (aussi appelées **tags**, en anglais). Par exemple, `` est une balise.

Le code PHP viendra s'insérer au milieu du code HTML. On va progressivement placer dans nos pages web des morceaux de code PHP à l'intérieur du HTML. Ces bouts de code PHP seront les parties dynamiques de la page, c'est-à-dire les parties qui peuvent changer toutes seules.

Le code suivant illustre cela :

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>Ma page web</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Ma page web</h1>
```

```
    <p>
```

```
      Bonjour <!-- Insérer le pseudo du visiteur ici --> !
```

```
    </p>
```

```
</body>
```

```
</html>
```

Comme vous pouvez le voir, on retrouve le code HTML que l'on connaît bien... et on insère en plus des données dynamiques au milieu. Ici, par exemple, c'est le pseudonyme : il change en fonction du visiteur.

Reconnaissez la forme d'une balise PHP

Si je vous parle de cela, ce n'est pas par hasard. Pour utiliser du PHP, on va devoir introduire une nouvelle balise.

1. Elle commence par `<?php`
2. Et se termine par `?>`

Voici une balise PHP vide :

```
<?php ?>
```

À l'intérieur, on écrira donc du code source PHP :

```
<?php /* Le code PHP se met ici */ ?>
```

On peut sans problème écrire la balise PHP sur plusieurs lignes.

En fait, c'est même indispensable car la plupart du temps, le code PHP fera plusieurs lignes.

Cela donnera quelque chose comme :

```
<?php
```

```
/* Le code PHP se met ici
```

```
Et ici
```

```
Et encore ici */
```

```
?>
```

Insérez une balise PHP au milieu du code HTML

La balise PHP que nous venons de découvrir s'insère au milieu du code HTML, comme je vous l'ai dit plus tôt. Pour reprendre l'exemple que l'on a vu au chapitre précédent :

```
<!DOCTYPE html>
```

```
<html>

<head>

  <title>Ceci est une page de test avec des balises PHP</title>

  <meta charset="utf-8" />

</head>

<body>

  <h2>Page de test</h2>


  <p>

    Cette page contient du code HTML avec des balises PHP.<br />

    <?php /* Insérer du code PHP ici */ ?>

    Voici quelques petits tests :

  </p>


  <ul>

    <li style="color: blue;">Texte en bleu</li>

    <li style="color: red;">Texte en rouge</li>

    <li style="color: green;">Texte en vert</li>

  </ul>


  <?php

  /* Encore du PHP

  Toujours du PHP */

  ?>

</body>

</html>
```

Peut-on placer une balise PHP n'importe où dans le code ?

Oui ! Vraiment n'importe où. Pas seulement dans le corps de la page, d'ailleurs : vous pouvez placer une balise PHP dans l'en-tête de la page (regardez la ligne 4 de l'exemple ci-dessous).

```
<!DOCTYPE html>

<html>

  <head>

    <title>Ceci est une page de test <?php /* Code PHP */ ?></title>

    <meta charset="utf-8" />

  </head>
```

Plus fort encore, vous pouvez même insérer une balise PHP au milieu d'une balise HTML, comme le montre la ligne 5 de l'exemple ci-dessous (bon, ce n'est pas très joli, je vous l'accorde) :

```
<!DOCTYPE html>

<html>

  <head>

    <title>Ceci est une page de test</title>

    <meta <?php /* Code PHP */ ?> charset="utf-8" />

  </head>
```

Comment ça fonctionne ? À quoi ça peut servir ?

Il faut se rappeler que PHP génère du code HTML. Nous allons mieux comprendre le fonctionnement en apprenant à afficher du texte en PHP.

Affichez du texte

Bon, tout ça c'est bien beau, mais il serait temps de commencer à écrire du code PHP, non ?

Vous allez mieux comprendre comment le PHP fonctionne, c'est-à-dire comment il génère du code HTML. Il est indispensable de bien comprendre cela ; soyez donc attentif !

Utilisez l'instruction echo

Le PHP est un langage de programmation, ce qui n'était pas le cas du HTML. Dans ce cours, nous partons de zéro.

Tout langage de programmation contient ce qu'on appelle des **instructions**. On en écrit une par ligne en général, et en PHP elles se terminent toutes par un point-virgule. Une instruction

commande à l'ordinateur d'effectuer une action précise.

Ici, la première instruction que nous allons découvrir permet d'insérer du texte dans la page web. Il s'agit de l'instruction echo, la plus simple et la plus basique de toutes les instructions que vous devez connaître.

Pour avoir plus d'informations sur l'instruction echo, vous pouvez consulter le site officiel de PHP : **[l'instruction echo](#)**. Le site officiel de PHP est la principale ressource pour la documentation, les téléchargements et les informations essentielles sur le langage de programmation PHP.

Voici un exemple d'utilisation de cette instruction :

```
<?php echo "Ceci est du texte"; ?>
```

<!-- Ou bien, avec des parenthèses -->

```
<?php echo("Ceci est du texte"); ?>
```

Comme vous le voyez, à l'intérieur de la balise PHP on écrit l'instruction echo suivie du texte à afficher entre guillemets. Les guillemets permettent de délimiter le début et la fin du texte, ce qui aide l'ordinateur à se repérer.

L'instruction se termine par un point-virgule comme je vous l'avais annoncé, ce qui signifie "**Fin de l'instruction**".

C'est beaucoup plus rare, mais l'instruction echo peut être entourée de parenthèses ouvrantes et fermantes.

Notez qu'il existe une instruction identique à echo appelée print , qui fait la même chose.

Cependant, echo est plus couramment utilisée.

Il faut savoir qu'on a aussi le droit de demander d'afficher des balises. Par exemple, le code suivant fonctionne :

```
<?php echo "Ceci est du <strong>texte</strong>"; ?>
```

Le mot « texte » sera affiché en gras grâce à la présence des balises et .

Comment faire pour afficher un guillemet ?

Bonne question.

Si vous mettez un guillemet, ça veut dire pour l'ordinateur que le texte à afficher s'arrête là. Vous risquez au mieux de faire planter votre beau code et d'avoir une terrible « Parse error ».

La solution consiste à faire précéder le guillemet d'un antislash \ :

```
<?php echo "Cette ligne a été écrite \"uniquement\" en PHP."; ?>
```

Vous savez que le code PHP s'insère au milieu du code HTML. Alors allons-y, prenons une page basique en HTML et plaçons-y du code PHP (ligne 12).

```
<!DOCTYPE html>

<html>

    <head>

        <title>Notre première instruction : echo</title>

        <meta charset="utf-8" />

    </head>

    <body>

        <h2>Affichage de texte avec PHP</h2>


        <p>

            Cette ligne a été écrite entièrement en HTML.<br />

            <?php echo("Celle-ci a été écrite entièrement en PHP."); ?>

        </p>

    </body>

</html>
```

Je vous propose de copier-coller ce code source dans votre éditeur de texte (Visual Studio Code ou autre...) et d'enregistrer la page. Nous allons l'essayer et voir ce qu'elle produit comme résultat.

Mais au fait, comment on fait déjà pour enregistrer une page PHP ?

Enregistrez une page PHP

Enregistrez la page avec l'extension .php . Par exemple : affichertexte.php , dans le dossier tests que je vous ai fait créer.

Il doit se trouver :

- dans C:\XAMPP\htdocs\tests sous Windows,

Quel que soit votre système d'exploitation, l'essentiel est que le fichier soit enregistré dans le dossier www si vous utilisez MAMP ou WAMP (ou un de ses sous-dossiers) ; sinon le fichier PHP ne pourra pas s'exécuter !

Une fois la page enregistrée, il faut maintenant la tester.

Testez une page PHP

Pour tester votre page PHP, tout dépend de votre système d'exploitation, mais la manœuvre est la même dans les grandes lignes.

1. Démarrez XAMPP si ce n'est pas déjà fait.
2. Allez à l'adresse <http://localhost/tests> . Une page web s'ouvre, indiquant tous les fichiers qui se trouvent dans le dossier tests . Vous devriez avoir le fichier `affichertexte.php` .
3. Cliquez dessus : votre ordinateur génère alors le code PHP puis ouvre la page. Vous avez le résultat devant vos yeux ébahis. 🤖🤖

Le même résultat peut être obtenu dans votre navigateur, en allant directement à l'adresse suivante : <http://localhost/tests/affichertexte.php> .

La méthode devrait être quasiment la même, que vous soyez sous Windows, macOS X ou Linux.

Si vous utilisez le serveur local de PHP, vous pouvez sauvegarder le fichier où vous le souhaitez, mais en contrepartie, vous devez toujours démarrer le serveur au niveau d'un dossier contenant le fichier à exécuter. Pour rappel, voici la commande : `php -S localhost:8080` .

Ne pas oublier : le serveur PHP recherche automatiquement un fichier nommé `index.php` comme point d'entrée par défaut !

Alors, que voyez-vous ?

Je pense que vous êtes étonné et surpris de ce que je vous ai fait faire : ça a l'air d'être inutile, et ce n'est pas tout à fait faux...

Le code PHP a « écrit » une ligne à l'écran, tout simplement.

Mais... ce n'est pas plus simple de l'écrire en HTML, finalement ?

En attendant, pour vous amuser et comprendre la force de PHP, essayez juste le code suivant :

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>Ma page web</title>
```

```
</head>

<body>

    <h1>Ma page web</h1>

    <p>Aujourd'hui nous sommes le <?php echo date('d/m/Y h:i:s'); ?>.</p>

</body>

</html>
```

Testez ce code et admirez : la date et l'heure s'affichent automatiquement sur la page web !

Attendez quelques minutes, puis actualisez la page : l'heure s'est mise à jour toute seule !

Regardez le code source de la page générée dans le navigateur : vous verrez qu'il n'y a pas de code PHP et que l'heure a directement été envoyée dans le code HTML après exécution du code PHP par le serveur.

Commentez votre code

Un **commentaire** est un texte que l'on écrit pour soi (ou pour d'autres développeurs qui vont travailler dessus) dans le code PHP.

Ce texte est ignoré, c'est-à-dire qu'il disparaît complètement lors de la génération de la page. Il n'y a que vous qui voyez ce texte.

Mais alors, à quoi sert un commentaire ?

Cela permet de vous y retrouver dans votre code PHP, parce que si vous n'y touchez pas pendant des semaines et que vous y revenez, vous risquez d'être un peu perdu.

Vous pouvez écrire tout et n'importe quoi, le tout est de s'en servir à bon escient.

Il existe deux types de commentaire :

1. Les commentaires **monolignes**.
2. Les commentaires **multilignes**.

Tout dépend de la longueur de votre commentaire. Je vais vous présenter les deux.

Faites des commentaires monolignes

Pour indiquer que vous écrivez un commentaire sur une seule ligne, vous devez taper deux slashes : « // ». Tapez ensuite votre commentaire.

Un exemple ?

```
<?php
```



```
echo "J'habite en Chine."; // Cette ligne indique où j'habite
```

```
// La ligne suivante indique mon âge
```

```
echo "J'ai 92 ans.";
```

```
?>
```

Je vous ai mis deux commentaires à des endroits différents :

- le premier est à la fin d'une ligne ;
- le second est sur toute une ligne.

À vous de voir où vous placez vos commentaires : si vous commentez une ligne précise, mieux vaut mettre le commentaire à la fin de cette ligne.

Faites des commentaires multilignes

Ce sont les plus pratiques si vous pensez écrire un commentaire sur plusieurs lignes, mais on peut aussi s'en servir pour écrire des commentaires d'une seule ligne. Il faut commencer par écrire `/*` puis refermer par `*/` :

```
<?php
```

```
/* La ligne suivante indique mon âge
```

```
Si vous ne me croyez pas...
```

```
... vous avez raison ;o) */
```

```
echo "J'ai 92 ans.";
```

```
?>
```

Ici, les commentaires n'ont pas grande utilité, mais vous verrez de quelle façon je les utilise dans les prochains chapitres pour vous décrire le code PHP.

En résumé

- Les pages web contenant du PHP ont l'extension `.php`
- Une page PHP est une simple page HTML qui contient des instructions en langage PHP.
- Les instructions PHP sont placées dans une balise ouvrante et fermante : `<?php ?>`
- Pour afficher du texte en PHP, on utilise l'instruction `echo`
- Il est possible d'ajouter des commentaires en PHP pour décrire le fonctionnement du code. On utilise pour cela les symboles `//` ou `/* */`