Définition et rôle du PHP

Le terme PHP est l’acronyme de « PHP Hypertext Preprocessor ». Le premier « P » de PHP est en effet lui-même l’abréviation de « PHP », une curiosité qui ne va pas présenter une grande importance pour nous.

Ce langage a été créé en 1994. Sa version stable la plus récente (au 15 juillet 2019) est la version 7.3.7. C’est la version sur laquelle je vais me baser dans ce cours.

Le PHP va nous permettre de créer des pages qui vont être générées dynamiquement. En d’autres mots, grâce au PHP, nous allons pouvoir afficher des contenus différents sur une même page en fonction de certaines variables : l’heure de la journée, le fait que l’utilisateur soit connu et connecté ou pas, etc.

Pour illustrer cela, prenons l’exemple d’un espace client sur un site web e-commerce. Un utilisateur arrive sur un site e-commerce sur lequel il a déjà commandé et crée un espace client. Lors de son arrivée sur le site, il dispose d’un formulaire de connexion à son espace client.

Il va alors devoir fournir des informations (généralement un pseudonyme et un mot de passe) pour pouvoir se connecter et accéder à son espace client. Cet espace client est personnalisé : il va certainement contenir l’historique des commandes de l’utilisateur, son profil avec ses informations de facturation et son adresse de livraison, etc.

Ici, lorsque l’utilisateur rentre ses informations de connexion, celles-ci vont être traitées et analysées en PHP. On va vérifier si les informations sont bonnes et si c’est le cas récupérer des informations spécifiques à cet utilisateur et générer dynamiquement les pages de son espace client avec ces informations.

Lorsqu’un utilisateur fournit des informations comme une adresse, un numéro de téléphone ou passe une commande, les données sont généralement enregistrées dans ce qu’on appelle une base de données. Le PHP va également nous permettre d’aller récupérer des données dans une base de données pour s’en resservir.

De plus, notez que le PHP va s’exécuter côté serveur. Il fait ainsi partie des langages qu’on nomme « server side » en opposition aux langages « client side » qui s’exécutent côté client. Nous expliquerons ces notions en détail dans la prochaine leçon.

Sites statiques et sites dynamiques

Les langages de programmation axés web peuvent être catégorisés selon deux grands types de classement :

* Langages statiques VS langages dynamiques ;
* Langages avec exécution côté client VS langages avec exécution côté serveur.

Les sites dits statiques se caractérisent par le fait qu’ils sont… statiques : ils ne possèdent ni interaction, ni la capacité de s’adapter aux visiteurs. Le code des différentes pages ne va pas changer en fonction d’un utilisateur ou d’une autre variable. Un site de type “CV” par exemple, ou un site servant simplement à présenter ou à donner des informations sur une chose en particulier vont généralement être des sites statiques car il n’y a aucune interaction ni adaptation dynamique avec le visiteur. Un site créé uniquement en HTML et en CSS par exemple sera toujours statique.

Les sites dynamiques, en revanche, vont pouvoir fournir des pages différentes pour chaque visiteur ou selon différentes contraintes et vont nous permettre d’interagir avec l’utilisateur en lui permettant de nous envoyer des données par exemple. De nombreux langages vont nous permettre de créer des sites dynamiques, chacun avec leurs points forts et leurs faiblesses et leur champ d’application. Dans ce cours, nous nous concentrons sur le binôme certainement le plus connu parmi ces langages : le PHP qui va être utile pour tout ce qui est calcul / traitement des données et le MySQL qui va nous servir à gérer nos bases de données.

Nous reparlerons de la distinction client / serveur dans la prochaine leçon. Ici, vous pouvez retenir qu’un site web créé uniquement avec des langages qui s’exécutent côte client sera statique tandis qu’un langage créé avec des langages qui s’exécutent côté client et des langages qui s’exécutent côté serveur sera généralement dynamique.