Résumé

PHP8 / MySQL

03/2023

Théo Yamanoglu - t.yamanoglu@ecole-ipssi.net

Introduction au PHP

- Le terme PHP est l'acronyme de « PHP Hypertext Preprocessor »
- Ce langage a été créé en 1994 par Rasmus Lerdorf pour son site web
- PHP en est à sa version 8 et est le langage de programmation web côté serveur le plus utilisé au monde
- Les langages côté client tels que HTML, CSS ou JavaScript sont interprétés par le navigateur Web que lorsqu'une page internet est ouverte
- Le code PHP est exécuté sur un serveur Web, qui génèrent une sortie HTML, qui est envoyée au navigateur Web. Le navigateur n'obtient pas le code réel (le script PHP) mais seulement le résultat de l'exécution du script

Introduction au PHP

- 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 et la date, le fait que l'utilisateur soit connu et connecté ou non, etc.
- Retenez 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
- Les sites dits statiques se caractérisent par le fait qu'ils sont... statiques. Un site créé en HTML et CSS sera toujours statique, son contenu ne variant pas
- Les sites dynamiques, en revanche, vont pouvoir fournir des résultats de pages différentes pour chaque visiteur selon différentes contraintes et vont nous permettre d'interagir avec l'utilisateur en lui permettant de nous envoyer des données

Introduction à MySQL

- MySQL est un système de gestion de bases de données relationnelle (SGBDR). Une base de données est un ensemble structuré de données
- Les données peuvent être des informations clients (nom, adresse, mot de passe, etc.), les commentaires d'un blog, le texte des articles, etc.
- Les informations ne vont pas être toutes stockées au même endroit mais plutôt dans plusieurs compartiments appelés « tables » qui vont pouvoir communiquer entre elles
- On ne va pas directement pouvoir interagir avec les bases de données car les données sont stockées d'une manière illisible pour un humain

Introduction à MySQL

- Pour manipuler les données stockées dans les bases de données, nous allons devoir utiliser un langage de bases de données, le plus célèbre est le SQL
- SQL est l'acronyme de Structured Query Language (Langage de Requêtes Structurées)
- Les avantages du MySQL sont sa simplicité d'utilisation, sa fiabilité et ses performance
- MySQL est la base de données open source la plus populaire au monde
- Pour résumer : le PHP 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

Fonctionnement d'Internet

- L'Internet est un système créé pour transporter de l'information. C'est un réseau de plusieurs réseaux qui utilisent chacun un protocole différents pour envoyer de l'information
- Le World Wide Web, ou plus simplement « Web », est un de ces réseaux, de machines interconnectées qui stockent des sites. Le Web n'est donc qu'une partie d'Internet
- Lorsqu'une machine est connectée à internet et fournit un accès à un site web, on l'appelle un serveur car elle « sert » les pages d'un site web

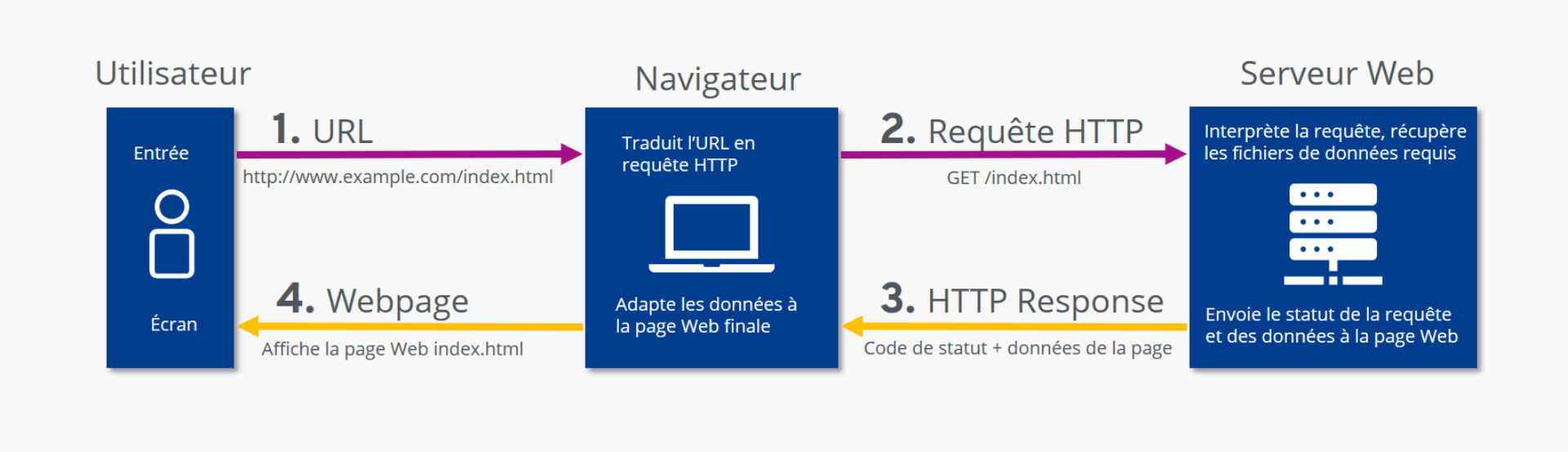
Fonctionnement d'Internet

- Pour pouvoir se comprendre et échanger des données toutes ces machines doivent parler la même langue, c'est-à-dire utiliser le même protocole
- Le Web repose ainsi sur le protocole HTTP, pour HyperText Transfer Protocol (protocole de transfert hypertexte) et sur son frère qui utilise des clefs de cryptage : le HTTPS (Secure HTTP)
- Pour accéder directement à une page web, on passe ainsi par un navigateur en utilisant le protocole HTTP ou HTTPS
- Un navigateur (côté client) et un serveur (côté serveur) ne vont pouvoir chacun effectuer que certaines opérations et lire certains langages

Prérequis

- La majorité des navigateurs ne sont capables de comprendre et de n'exécuter que du code HTML, CSS et JavaScript
- Un navigateur est incapable de comprendre du code PHP. Lorsqu'un navigateur demande à un serveur de lui servir une page, le serveur va donc se charger d'exécuter tout code qui ne serait pas compréhensible par le navigateur
- Une fois ces opérations effectuées, le serveur va renvoyer le résultat (la page demandée après interprétation) sous forme de code compréhensible par le navigateur, c'est-à-dire principalement en HTML. Le navigateur va alors afficher la page au visiteur
- Les opérations réalisées côté serveur vont être transparentes pour le visiteur et que celui-ci ne va jamais avoir accès ni pouvoir voir le code PHP exécuté côté serveur

Prérequis



- Pour coder en HTML et en CSS et afficher le résultat de son code, un simple éditeur de texte et un navigateur suffisent. Pour le PHP et le MySQL, cependant, cela va être un peu plus complexe car le code va s'exécuter côté serveur
- Installez le serveur pour : Windows -> WAMP / Mac Os -> MAMP / Linux -> XAMPP

Premier script PHP

- Nous allons écrire nos scripts PHP soit dans des fichiers dédiés, c'est-à-dire des fichiers qui ne vont contenir que du PHP, soit intégrer le PHP au sein de nos fichiers HTML
- Les fichiers qui contiennent du PHP vont devoir être enregistrés avec l'extension .php. Dans le cas où on intègre du code PHP dans un fichier HTML, il faudra également changer son extension en .php
- Le serveur, pour être en mesure d'exécuter le code PHP, va devoir le reconnaitre. Pour lui indiquer qu'un script ou que telle partie d'un code est écrit en PHP, nous allons entourer ce code avec une balise PHP qui a la forme suivante : <?php ?>

Syntaxe de base du PHP

- Ecrire du code PHP : <?php ?>
- La syntaxe des commentaires : // monoligne et /* */ pour le multiligne
- Les parties echo 'Hello World
'; et echo "Bonjour le Monde"; sont ce qu'on appelle des instructions car on « demande » au code de faire quelque chose
- En PHP « echo » permet d'afficher une chaine de caractères
- Une instruction en PHP doit toujours se terminer par un point-virgule
- Les apostrophes ou guillemets permettent de délimiter les chaines de caractères
- L'antislash \ est en PHP un caractère d'échappement : il sert à indiquer au serveur qu'il ne doit pas tenir compte de la signification spéciale du caractère qui le suit dans le cas où ce caractère est un caractère spécial

Syntaxe de base du PHP

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             //Affiche "Hello World" avec un retour à la ligne
13.             echo 'Hello World <br>'; //Ceci est un commentaire PHP
14.
15.             /*Affiche
16.               "Bonjour le Monde
17.             */
18.             echo "Bonjour le Monde"; /*Ceci est un commentaire PHP*/
19.         ?>
20.         <p>Un paragraphe</p>
21.     </body>
22. </html>
```

Les variables

- Une variable est un conteneur servant à stocker des informations de manière temporaire, comme une chaine de caractères (un texte) ou un nombre par exemple
- En PHP, les variables ne servent à stocker une information que temporairement. Plus précisément, une variable ne va exister que durant le temps de l'exécution du script l'utilisant
- Afin de stocker des informations durablement nous pourrons utiliser les fichiers, cookies ou les bases de données dont nous parlerons plus tard
- On va pouvoir choisir le nom qu'on souhaite donner à chacune de nos variables. Cependant, il y a quelques règles à respecter et à connaitre lors de la déclaration d'une variable

Les variables

- Toute variable en PHP doit commencer par le signe $ qui sera suivi du nom de la variable
- Le nom d'une variable doit obligatoirement commencer par une lettre ou un underscore (_) et ne doit pas commencer par un chiffre
- Le nom d'une variable ne doit contenir que des lettres, des chiffres et des underscores mais pas de caractères spéciaux
- Le nom d'une variable ne doit pas contenir d'espace
- Le nom des variables est sensible à la casse en PHP. Cela signifie que l'usage de majuscules ou de minuscules va créer des variables différentes. Par exemple, les variables $Var, $var et $vAr vont être des variables différentes
- Enfin, il existe des noms « réservés » en PHP. Vous ne pouvez pas utiliser ces noms comme noms pour vos variables, tout simplement car le langage PHP les utilise déjà pour désigner différents objets intégrés au langage

Les opérateurs

- Le signe égal simple = n'est pas un opérateur de comparaison mais un opérateur d'affection (ou d'assignation) : il sert à affecter une valeur à une variable
- L'égalité valeurs est symbolisée en PHP par le double signe égal ==
- L'égalité en termes de valeurs et de types de données, va être représentée en PHP par le triple signe égal : ===
- En PHP, l'opérateur de concaténation est le point .
- Opérateur arithmétique : + - * / %

Les types de données

- Les variables en PHP vont pouvoir stocker différents types de données :
  - Le type « chaine de caractères » ou String en anglais ;
  - Le type « nombre entier » ou Integer en anglais ;
  - Le type « nombre décimal » ou Float en anglais ;
  - Le type « booléen » ou Boolean en anglais ;
  - Le type « tableau » ou Array en anglais ;
  - Le type « objet » ou Object en anglais ;
- La fonction var_dump() en PHP permet d'afficher le contenu et le type d'une variable

Les fonctions interne

- Une fonction correspond à une série cohérente d'instructions qui ont été créées pour effectuer une tâche précise. Pour exécuter le code contenu dans une fonction, il va falloir appeler la fonction
- Les fonctions prêtes à l'emploi sont l'une des plus grandes forces du PHP puisqu'il en existe plus de 1000 qui vont couvrir quasiment tous nos besoins
- Chaînes de caractères : https://www.php.net/manual/fr/book.strings.php
- Tableaux : https://www.php.net/manual/fr/book.array.php
- Traitement des images : https://www.php.net/manual/fr/book.image.php
- Envoi de mail : https://www.php.net/manual/fr/book.mail.php
- Manipulation de date : https://www.php.net/manual/fr/function.date.php
- Manipulation de fichiers : file_get_contents(), fopen(), ...

Les conditions

- Les conditions vont ainsi être un passage incontournable pour rentre un site dynamique puisqu'elles vont nous permettre d'exécuter différents codes et ainsi afficher différents résultats selon le contexte
- La condition if (si)
- La condition if... else (si... sinon)
- La condition if... elseif... else (si... sinon si... sinon)

Les opérateurs de comparaison

Opérateur	Définition
==	Permet de tester l'égalité sur les valeurs
===	Permet de tester l'égalité en termes de valeurs et de types
!=	Permet de tester la différence en valeurs
<>	Permet également de tester la différence en valeurs
!==	Permet de tester la différence en valeurs ou en types
<	Permet de tester si une valeur est strictement inférieure à une autre
>	Permet de tester si une valeur est strictement supérieure à une autre
<=	Permet de tester si une valeur est inférieure ou égale à une autre
>=	Permet de tester si une valeur est supérieure ou égale à une autre

L’instruction switch

```php
<?php
    $x = 2;

    switch($x){
        case 0:
            echo '$x stocke la valeur 0';
            break;
        case 1:
            echo '$x stocke la valeur 1';
            break;
        case 2:
            echo '$x stocke la valeur 2';
            break;
        case 3:
            echo '$x stocke la valeur 3';
            break;
        case 4:
            echo '$x stocke la valeur 4';
            break;
        default:
            echo '$x ne stocke pas de valeur entre 0 et 4';
    }
?>
```

Les boucles while et for

```php
<?php
    $x = 0;

    while($x <= 10){
        echo '$x contient la valeur ' .$x. '<br>';
        $x++;
    }
?>
```

```php
<?php
    for($x = 0; $x <= 5; $x++){
        echo '$x contient la valeur ' .$x. '<br>';
    }
?>
```

include() et require()

- Les instructions PHP include et require vont nous permettre toutes deux d'inclure le contenu des fichiers de code à l'intérieur d'autres fichiers de code
- Exemple : entête, menu et pieds de page identique sur toutes les rubriques
- L'instruction require est plus « stricte » que include, si l'inclusion a été faite avec include, le PHP renverra un simple avertissement et le reste du script s'exécutera quand même tandis que si la même chose se produit avec require, une erreur fatale sera retournée par PHP et l'exécution du script s'arrêtera immédiatement

Les fonctions défini par l'utilisateur

- En plus des fonctions internes, le PHP nous laisse la possibilité de définir nos propres fonctions
- Notez qu'à la différence des variables le nom des fonctions est insensible à la casse. Cela signifie que l'utilisation de majuscules et des minuscules ne servent pas à différencier une fonction d'une autre

```php
<?php
    function bonjour(){
        echo 'Bonjour à tous';
    }
?>
```

- La structure de contrôle return va nous permettre de demander à une fonction de retourner un résultat qu'on va ensuite pouvoir stocker dans une variable ou autre pour le manipuler

Les variables superglobales

- Les variables superglobales sont des variables internes au PHP
- Les variables vont être accessibles n'importe où dans le script et quel que soit le contexte
- Il existe 9 superglobales en PHP :
  - $GLOBALS tableau qui stocke automatiquement toutes les variables globales déclarées
  - $_SERVER contient des informations sur le serveur et le script exécuté
  - $_REQUEST contient toutes les variables envoyées via GET, POST et par les cookies
  - $_FILES contient des informations sur un fichier téléchargé (type, taille, nom, ...)
  - $_ENV contient des informations liées à l'environnement dans lequel s'exécute le script
  - $_SESSION tableau associatif qui contient toutes les variables de session
  - $_COOKIE tableau associatif qui contient toutes les variables passées via des cookies HTTP
  - $_GET manipuler les informations envoyées via un formulaire HTML
  - $_POST manipuler les informations envoyées via un formulaire HTML

Les cookies

- Un cookie est un petit fichier texte qui contient qu'une quantité limitée de données
- Les cookies vont être stockés sur les ordinateurs de vos visiteurs. Ainsi, à tout moment, un utilisateur peut lui même supprimer les cookies de son ordinateur
- Les cookies vont toujours avoir une durée de vie limitée. On pourra définir la date d'expiration d'un cookie
- Généralement, nous allons utiliser les cookies pour faciliter la vie des utilisateurs en préenregistrant des données les concernant comme une préférence de thème par exemple
- On évitera toujours de stocker des informations sensibles dans les cookies car ils sont stockés sur l'ordinateur des visiteurs et nous n'avons donc aucune maitrise ni aucun moyen de les sécuriser après le stockage

Les cookies

- Créer un cookie : https://www.php.net/manual/fr/function.setcookie.php
- Une particularité notable de cette fonction est qu'il va falloir l'appeler avant d'écrire tout code HTML pour qu'elle fonctionne
- Pour récupérer la valeur d'un cookie, nous allons utiliser la variable superglobale $_COOKIE

```php
<?php
    setcookie('user_id', '1234');
?>
<!DOCTYPE html>
<html>
    <head>
        <title>Cours PHP & MySQL</title>
```

```php
<?php
    if(isset($_COOKIE['user_id'])){
        echo 'Votre ID de session est le ' .$_COOKIE['user_id'];
    }
?>
```

Les sessions PHP

- Une session en PHP correspond à une façon de stocker des données différentes pour chaque utilisateur en utilisant un identifiant de session unique
- Un des grands intérêts des sessions est qu'on va pouvoir conserver des informations pour un utilisateur lorsqu'il navigue d'une page à une autre. De plus, les informations de session ne vont cette fois-ci pas être stockées sur les ordinateurs de vos visiteurs à la différence des cookies mais plutôt côté serveur ce qui fait que les sessions vont pouvoir être beaucoup plus sûres que les cookies
- Une session démarre dès que la fonction session_start() est appelée et se termine en général dès que la fenêtre courante du navigateur est fermée
- La superglobale $_SESSION est un tableau associatif qui va contenir toutes les données de session une fois la session démarrée

Les méthodes GET & POST

- GET et POST sont des méthodes d'accès définies dans le protocole HTTP
- Le choix de la méthode dépend de la façon dont les données sont reçues, de la taille et la nature des données
- Lors de la validation d'un formulaire la méthode GET ajoute les données à l'URL

```
<form method="get" action="page.html">
</form>
```

- Cette méthode, fait circuler les informations du formulaire en clair dans la barre d'adresse en suivant le format suivant : Les données sont séparées de l'adresse de la page par le code ? et entre elles par le code &
- Exemple : page.html?nom=Mathieu&prenom=Martin&age=18

Les méthodes GET & POST

- Notez que lorsqu'on utilise le bouton retour ou actualiser du navigateur, les requêtes GET sont exécutées à nouveau
- La méthode POST quant à elle, transmet les informations du formulaire de manière masquée **mais non cryptée**

```
<form method="post" action="page.php">
</form>
```

- La méthode POST est préférée lorsqu'il y'a un nombre important de données à transmettre ou bien lorsqu'il faut envoyer des données sensibles comme des mots de passe. Dans certains cas, seule la méthode POST est admise : un upload de fichier par exemple