

EXPOSÉ : SYNTHÈSE DE NOTRE AVANCÉ DANS LE PROJET

Introduction

Notre groupe constitué de Chloé, Eliott, Sophie et Alba s'occupe au sein d'un projet sur la carte RFIID du travail sur les interfaces homme-machine.

Une interface correspond à l'ensemble des règles et des conventions qui permettent l'échange d'informations entre 2 systèmes donnés. Dans le cadre de la carte RFID pour les interfaces homme-machine il ya différents cas de figure : **des applications** tel que HeidiSQL, des systèmes en ligne comme PHPmyadmin ou encore **des pages ou des logiciels conçu pour des applications**. Ces interfaces permettent en particulier la gestion des données. Ayant traité globalement au début de notre projet de heidiSQL, de PHPmyadmin et des langages de programmation, nous avons choisi de consacrer plusieurs semaines à approfondir nos connaissances sur la programmation.

Ainsi, Sophie et Alba ont appris les rudiments du langage PHP, et Chloé et Eliott ont approfondi l'étude de l'application heidiSQL, cela, afin d'apprendre à créer et gérer des tables et des données. Ceci est pour permettre l'interfaçage entre le client et le système ; autrement dit entre l'homme et la machine : tel est l'objet de notre recherche.

Le langage PHP

Qu'est ce que c'est

PHP est un langage de programmation, c'est-à-dire, un code de communication (qui permet de définir l'ensemble des instructions à effectuer par l'ordinateur lors de l'exécution d'un programme.) Un langage de programmation est comme un autre langage qui possède ses propres mots et ses propres règles. Il permet de donner des ordres à l'ordinateur. PHP est un **langage interprété (un langage de script) exécuté du côté serveur**. La syntaxe du langage provient de celles du langage C, de Perl et de Java. Ses principaux atouts sont :

- La gratuité et la disponibilité du code source
- La simplicité d'écriture des scripts ;
- La possibilité d'inclure le script PHP au sein d'une page
- La simplicité d'interfaçage avec des bases de données (le plus utilisé avec ce langage est *MySQL*;
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).

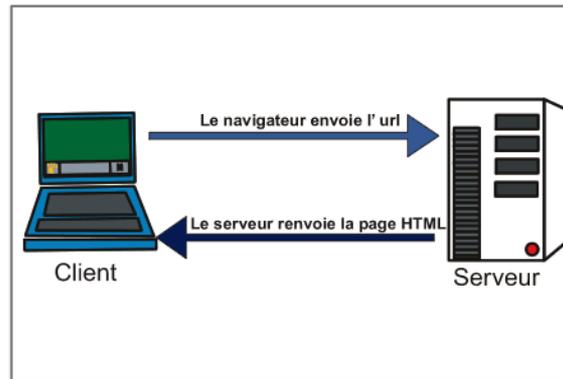
PHP permet un interfaçage simple avec de nombreux **systèmes de gestion de bases de données**.

Son origine

Le langage PHP a été mis au point en 1994 par Rasmus Lerdorf. Ce langage de script lui permettait de conserver la trace des utilisateurs venant consulter son CV sur son site, ce grâce à l'accès à une base de données par l'intermédiaire de requêtes SQL. Rasmus Lerdorf mit en ligne en 1995 la première version de ce programme qu'il baptisa *Personal Sommaire Page Tools*, puis *Personal Home Page v1.0* (traduit par *page personnelle version 1.0*). Puis PHP2, 3, 4 vont se développer et PHP5 en 2008.

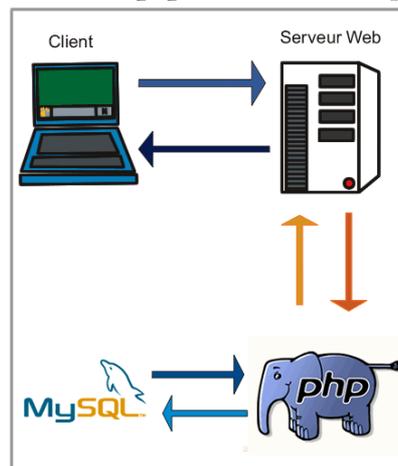
Comment ça marche

PHP est un langage de programmation qui s'intègre dans vos pages HTML.
Voici, en simplifiant, ce qu'il se passe lorsque vous consultez une page HTML dite statique.



- Votre navigateur envoie l'adresse URL (Uniform Resource Locator) que vous avez tapée
- Le serveur web est un "ordinateur" présent sur l'Internet et qui héberge la page que vous demandez. Sur ce serveur on trouve Apache, logiciel apte à traiter les requêtes HTTP que vous envoyez lorsque vous demandez une page web. Apache va donc chercher le fichier demandé et renvoyer à votre navigateur la page HTML.
- Votre navigateur interprète les différents langages se trouvant dans ce fichier et affiche la page.

Lorsqu'une page HTML contient du langage PHP voici ce qui se passe :



- Votre navigateur envoie l'adresse que vous avez tapée.
- Le serveur web cherche si le fichier existe, et si celui-ci porte une extension reconnue comme une application PHP. Si c'est le cas, le serveur web transmet ce fichier à PHP.
- PHP analyse et exécute le code PHP. La base de données renvoie les informations voulues au script qui peut les exploiter (pour les afficher par exemple).
- PHP continue d'analyser la page, puis retourne le fichier dépourvu du code PHP au serveur web.
- Le serveur web renvoie donc un fichier ne contenant plus de PHP, donc seulement du HTML au navigateur qui l'interprète et l'affiche.

Comment écrire en langage PHP

Un code PHP est un ensemble d'instructions se terminant par un point virgule et compris entre les délimiteurs ou balises suivants : `<?php et ?>`. On peut passer du langage HTML à celui PHP en incluant un script entre balises PHP au sein même de balises HTML. Voici ci-dessous un exemple :

```
<html
<head
<title
/ title>
/ head>
<body
< ?php
?>
/ body>
/ html>
```

Nous avons traité de fonctions et de variables.

❖ La fonction écho

Cette fonction sert à écrire un texte. On peut utiliser la fonction echo de la manière suivante (comme montré ci dessous) :

```
< ?php
echo "Bonjour" ;
?>
```

Le texte qui suit le mot echo est entre guillemet et comme toute chaîne d'informations se finit par un point virgule. On voit apparaître sur notre écran *Bonjour*.

❖ Les variables

On peut inclure des variables dans un script PHP. Une variable est désignée par le signe \$ suivi par le nom qu'on lui donne. On peut combiner fonctions et variables. Voici un exemple.

```
<?php
$magasin_ou_je_fais_mes_courses = "Leclerc";
$nombre_de_pommes_achete = 5;
$nombre_de_litre_de_lait = 2;
$nombre_de_pizza = 1;
echo "      Aujourd'hui      chez      $magasin_ou_je_fais_mes_courses      j'ai
acheté $nombre_de_pommes_achete pommes, $nombre_de_litre_de_lait litres de lait et
$nombre_de_pizza pizza.";
?>
```

On verra apparaître sur notre écran *Aujourd'hui chez Leclerc j'ai acheté 5 pommes, 2 litres de lait et 1 pizza*.

On peut par le biais des variables effectuer des calculs. Par exemple :

```

< ?php
$chiffre = 5 ;
$chiffre1 = 6 ;
$chiffre2 = ($chiffre+$chiffre1) ;
echo "$chiffre2" ;
?>

```

On verra apparaître sur notre écran 11 qui est la somme de 5 et 6.

❖ La fonction if

Voici un exemple:

```

< ?php
    $loggedIn = true ;
    if ($loggedIn == true) {
        echo "Vous êtes connecté" ;
    } else {
        echo "Il faut vous connecter" ;
    }
?>

```

Dans ce cas, l'égalité « \$loggedIn = true » résulte correcte. On voit sur l'écran *Vous êtes connecté*. Si on mettait au début une différente variable, l'égalité serait fausse et donc le serveur afficherait *Il faut vous connecter*.

```

< ?php
    $loggedIn = false ;
    if ($loggedIn == true) {
        echo "Vous êtes connecté" ;
    } else {
        echo "Il faut vous connecter" ;
    }
?>

```

❖ La boucle while

While signifie « tant que » et prend généralement la forme suivante :
« Tant que la condition est "vraie", la boucle continue à suivre les instructions. »
Voici un exemple :

```

<?php
$compte = 0;
while ($compte <= 20)
{
echo "Ligne $compte<br />";
$compte++;
}

```

?>

Dans cet exemple, on attribue une variable \$compte qui aura pour valeur 0. Notre condition est que tant que la variable \$compte qui a pour valeur 0 est plus petit ou égal à notre instruction qui est de 20, l'instruction doit continuer.

❖ Les variables arrays

Les variables arrays sont un autre type de variable.

Ex : on crée le site process.php, c'est à ce dernier qu'on va envoyer des données. Pour accéder à l'information qu'on a sur le site on utilise les arrays.

```
<?php
$name=$_POST[« name »];
Echo"hello".$name;
?>
```

On peut aussi créer une variable array qui "combine" plusieurs variables de la manière suivante:

```
< ?php
$person1=Alice ;
$person2=Bob ;
$person 3 =Catherine ;
$people = array("Alice", "Bob", "Catherine");
?>
```

❖ Pour mettre des dates, des heures dans un script en PHP

Pour manipuler les **dates et l'heure en PHP** on utilise principalement la **fonction date** auquel nous ajoutons le ou les «Format» du tableau ci-dessous dans l'ordre qui nous convient.

Par exemple, pour obtenir juste l'année en cours sur 4 chiffres, on écrit simplement

```
<?php
echo date('Y');
?>
```

ce qui nous donne comme résultat 2014.

Pour afficher l'heure, c'est la même chose. Néanmoins, il faut savoir que l'heure qui est renvoyé par le serveur est l'heure locale du lieu géographique où se trouve le serveur...

Pour les jours, on a différents cas de figures :

- « j » correspond au jour du mois sans les zéros initiaux.
- « d » correspond au jour du mois sur deux chiffres (avec un zéro initial)
- « l » correspond au jour de la semaine en anglais.
- « w » correspond au jour de la semaine au format numérique.

Pour les semaines, on a un choix :

« W » correspond au numéro de la semaine dans l'année.

Pour les années, il y a deux possibilités :

« Y » correspond à l'année sur 4 chiffres

« y » correspond à l'année sur 2 chiffres.

Pour les heures, deux cas de figures :

« G » correspond à l'heure (forma 24h) sans les zéros initiaux

HeidiSQL

HeidiSQL est un outil d'administration de base de données. Il a été développé pour être utilisé avec le SGBD (système de gestion de base de données) relationnel MySQL disponible commercialement ou gratuitement. Il est un système d'interface simple avec MySQL.

Ansgar Becker a commencé le développement de MySQL en 1999 en nommant le projet " MySQL -Front ", et a connecté les serveurs MySQL et les bases de données. Le développement privé a continué jusqu'à la version 2.5 paru en Avril 2006. Becker a ensuite ouvert l'application sur SourceForge, renommant le projet " HeidiSQL ".

Depuis la version 8.0, HeidiSQL propose son interface en 22 langues autres que l'anglais. Des traductions sont apportées par les utilisateurs de différents pays.

HeidiSQL vous permet de parcourir et éditer les données, créer et modifier tables, vues, procédures, déclencheurs. Même si les outils de gestion interactive de bases de données ne permettent pas en général de créer un programme d'exploitation, HeidiSQL peut suffire dans beaucoup de cas, et servir de moyen d'apprentissage.

Les tableaux de gestion de base de données peuvent être créés par un raccourci ou grâce à l'outil « requête », qui est une suite de ligne de code à écrire. Les principaux mots utilisés sont les mots

- « create » (pour créer l'objet qui suit),
- « table » (un tableau),
- « show » (pour montrer l'objet qui suit).

Pour utiliser HeidiSQL on fait des requêtes, et les informations demandées nous sont renvoyées.

Heidi SQL est une application intéressante dans le cadre de l'interface homme-machine. Toutefois, les requêtes n'étant pas les plus simples à effectuer, HeidiSQL ne sera pas la meilleure application pour les élèves... Très performante, elle pourra cependant servir aux administrateurs.

Conclusion

Pendant ce premier trimestre, nous avons acquis des bases de langage PHP et MySQL (pour PHP, fonctions et variables.)

Néanmoins, nous réalisons qu'il y a encore beaucoup à travailler. Il faut d'abord amplifier nos connaissances dans ces deux langages puis travailler sur le logiciel lui même. Nous avons établi une liste des objectifs à atteindre dans les prochains mois :

- le contenu du site (fonctions intégrées dans le logiciel, informations incluses, organisation des différentes pages, etc.)
- le système de l'achat des boissons (ajout de l'argent par espèce à la caisse ou par carte bancaire ? Dans ce cas, comment protéger le site ? Comment établir les liens entre banque et lycée ?)
- esthétique du site (ceci est un problème mineur mais quelles police et couleur choisir, quelle sera l'organisation spatiale du site ?)

