

PRÉSENTATION DU LANGAGE *PHP*

1. Qu'est-ce que le *PHP* ?

- Le *PHP* (pour *PHP: Hypertext Preprocessor*) est un langage de script qui, comme *ASP* et *JSP*, s'exécute du côté serveur.

Même si cela n'est pas obligatoire, le code *PHP* est habituellement inclus dans une page *XHTML* classique.

Pour être exécutée, le suffixe de l'application doit être **.php** (mais il est possible de changer le suffixe à notre guise).

- Comme tout code exécuté côté serveur, le produit de l'exécution d'une application *PHP* apparaîtra du côté client sous forme de code *XHTML/CSS/JavaScript*.
- Il n'existe aucune façon pour l'utilisateur de voir le code source *PHP* à partir du navigateur.
- Le code *PHP* doit être inclus entre les balises **<?php et ?>** pour être traité comme tel.

Tout code qui n'est pas entre les balises ci-dessus mentionnées est ignoré par l'interpréteur *PHP* et est envoyé directement au navigateur.

Comme vous pourrez le constater rapidement, la syntaxe du langage est très similaire aux langages *C#*, *Java* et *JavaScript*.

Le langage *PHP* est faiblement typé ce qui exige du programmeur d'être très rigoureux dans la rédaction de son code. Le préfixage des identificateurs sera donc très important (Ex. : **\$intQteProduit**, **\$dblTotalVentes**, **\$binTrouve**, etc.)

2. Documentation disponible sur le *PHP*

- Un site incontournable pour tout programmeur *PHP* : **www.php.net**.
- Un manuel de référence très complet (en français) est disponible en ligne à l'adresse : www.php.net/manual/fr/.

3. Avant de commencer

a. Un petit mot sur la structure de langage **echo**

- Structure de langage *PHP* permettant d'afficher une chaîne de caractères.
Syntaxe : void **echo** (string arg1 [, string ...])
 void **echo** string arg1 [, string ...]
- Il n'y a pas de saut de ligne automatique dans le résultat de l'exécution du code *PHP*. Lorsque l'on désire provoquer un saut de ligne dans une page Web, nous devons utiliser la balise **
**.

On utilisera la séquence « `\n` » pour améliorer la présentation du code *XHTML* généré. Notez que l'ajout de « `\n` » n'a aucun effet dans le résultat affiché à l'écran.

b. Utilisation d'un séparateur d'instructions

- En *PHP*, toute instruction doit se terminer par un point-virgule.
- Même s'il est possible d'écrire plus d'une instruction par ligne (séparées de points-virgules), il n'est pas suggéré de le faire, car cela peut nuire à la lisibilité du code.

c. Insertion de commentaires

- Il existe trois façons d'insérer un commentaire en *PHP* soit...

```
// Commentaire
# Commentaire
/* Commentaire */
```

- Chaque fonction *PHP* devra être précédée d'une boîte de commentaires qui devra prendre la forme ci-dessus. Détails à venir.

```
/*
|-----|
| nomFonction (aaaa-mm-jj)
| Brève description
|-----|
*/
```

4. Deux grands types d'application *PHP*

Pur *PHP*

- Nous emploierons l'expression « *PHP* pur » ou « pur *PHP* » pour désigner une application dont le code source est uniquement en *PHP*.

Le parfait exemple d'un tel type d'application serait celui qui permettrait d'alimenter en données une application *AJAX*. Un générateur de code source pourrait être également un bon exemple.

Mixte *XHTML-PHP*

- Contrairement à une application de type *pure PHP*, ce type d'application est composé d'un mélange de code *XHTML*, *CSS*, *JavaScript* et *PHP*.

Même si le code source est composé de *XHTML*, de *CSS* et de *JavaScript*, il est impératif que le suffixe d'une telle application soit **.php**.

Nous reviendrons rapidement sur le mixage du *XHTML* et du *PHP*.