Chapitre 2

Introduction au PHP

2.1. Ecrire à l'écran

L'une des premières choses que l'on apprend lorsqu'on débute dans un langage de programmation, c'est d'écrire *bonjour.*
La fonction qui permet d'afficher *bonjour* à l'écran et que vous verrez constamment dans ce livre, est *echo*.
Voici ce que cela nous donne en PHP.

```
1.<?php
2.echo 'Bonjour';
3.?>
```

▪ **Première remarque**
Vous voyez que pour écrire du PHP, nous devons entourer le code de deux éléments :
<?php : cet élément indique que tout ce qui va suivre dans le code sera du PHP.
?> : Cet élément indique la fin du code PHP.

▪ **Deuxième remarque**
A la fin de la ligne 2 qui est la ligne de notre code PHP, vous remarquerez qu'il y a le caractère ";" (point virgule).
Ce caractère indique à PHP que l'instruction est terminée et que par conséquent, on peut soit fermer le code PHP avec l'élément **?>**, soit écrire une nouvelle instruction qui sera elle-même terminée par un ";"
Attention, l'oubli de ce caractère entraînera une erreur ! Sauf dans le cas où l'instruction qui ne possède pas ce caractère est la dernière instruction. Mais d'une manière générale : employez toujours le ";" à la fin d'une instruction. Cela fait partie des bonnes règles. Et en programmation, il faut être rigoureux.

Il existe une autre fonction en PHP qui permet d'écrire à l'écran comme le *echo*. Cette fonction s'appelle *print*.
Le code suivant nous affichera également *bonjour.*

```
1.<?php
2.print 'Bonjour';
3.?>
```

Une petite information complémentaire concernant les fonctions *print* et *echo*, elles peuvent s'écrire avec des parenthèses, comme ceci :

```
1.<?php
2.echo ('Bonjour');
3.?>
```

Ces parenthèses sont rarement utilisées, vous pouvez donc vous en dispenser.
Enfin, voici une dernière façon d'écrire *bonjour* à l'écran :

```
1.<?= 'Bonjour'; ?>
```

Nous n'utiliserons pas ce type d'écriture dans ce livre.

2.2. Les commentaires

D'une manière générale, il est toujours très opportun de commenter son code lors de son écriture. Lorsqu'on revient sur un code après quelques semaines, on est toujours très content d'y trouver quelques commentaires.
Il existe deux façons d'écrire des commentaires en PHP.

▪ **La première façon**
// Commentaire sur une ligne.

▪ **La seconde façon**
/* Ceci est un commentaire
écrit sur plusieurs
lignes.
*/

Voici un exemple d'application.

```
1.<?php
2.echo 'Bonjour'; // Ce code affichera Bonjour
3./*
4.Le code suivant affichera également Bonjour :
5.*/
6.print 'Bonjour';
7.?>
```

// Ce code affichera Bonjour :
est un commentaire sur une ligne.
/*
Le code suivant affichera également Bonjour :
*/
est un commentaire sur plusieurs lignes.

Chapitre 3

Les variables

3.1. Découverte des variables

Tout langage de programmation est basé sur la notion de variable.

▪ **Qu'est-ce qu'une variable ?**
Comme son nom l'indique, une variable est un élément qui varie.
Prenons un exemple simple pour bien comprendre cette notion.

```
Bonjour Chris.
Bonjour Mike.
```

Nous venons d'écrire deux phrases simples comportant chacune deux mots.
Si nous comparons ces deux phrases, nous pouvons nous apercevoir qu'il y a d'un côté un point commun et de l'autre une divergence.
Le point commun de ces deux phrases est le mot *Bonjour.*
La divergence de ces deux phrases sont les mots *Chris* et *Mike*.

▪ **Traduisons maintenant cela en langage informatique**
Le mot *Bonjour* est une chaîne de caractères non variable, donc constante.
Alors que les mots *Chris* et *Mike* eux varient d'une phrase à l'autre. Ces mots sont donc variables.
Nous pourrions alors résumer ces deux phrases en une seule et écrire ceci :
Bonjour Variable.

Ainsi nous pouvons à loisir remplacer le mot *Variable* par le mot *Chris* et ainsi écrire la phrase :
Bonjour Chris.

Ou bien nous pouvons remplacer le mot *Variable* par le mot *Mike* et ainsi écrire la phrase :
Bonjour Mike.

Mais ce qui est vraiment puissant dans cette notion de variable, c'est que nous pouvons remplacer le mot *Variable* par n'importe quel mot ou n'importe quelle chaîne de caractères, comme par exemple la chaîne de caractères *tout le monde*.
Ainsi notre phrase deviendrait :
Bonjour tout le monde.

Après cette petite introduction destinée à vous ouvrir l'esprit au monde passionnant de l'informatique, nous allons entrer dans le vif du sujet.

3.2. Ecrire nos premières variables

3.2.1. Les règles à respecter

En PHP, une variable est toujours précédée du mot-clé $.
Une variable peut s'appeler de n'importe quel nom du moment qu'elle respecte ceci :
- La première lettre de son nom doit obligatoirement commencer soit par un caractère alphabétique, soit par le caractère underscore (c'est-à-dire ceci : _)
- Son nom ne doit contenir aucun espace et aucun caractère autre qu'un caractère alphanumérique non accentué ou le caractère underscore.

Enfin, les variables sont sensibles à la casse. Cela signifie qu'elles font la différence entre des lettres écrites en majuscule et celles écrites en minuscule.
Voyons maintenant tout cela de plus près.
$nom est une variable.
$Nom est également une variable.
$_nom est aussi une variable.

En revanche,
$Mon Nom est invalide car elle contient un espace.
$Mon-Nom est également invalide car elle contient un caractère autre qu'un caractère alphanumérique et autre que le caractère underscore.
$2Nom est elle aussi invalide car la première lettre de son nom ne commence pas par un caractère alphabétique ou le caractère underscore.
$prénom n'est pas valide car elle contient un caractère accentué.

Enfin, les variables **$nom** et **$Nom** sont différentes car comme précisé plus haut, les variables sont sensibles à la casse.

3.2.2. Déclaration de notre première variable

Nous allons appeler notre première variable *MyFirstVariable*.
Voici donc notre première variable PHP :
$MyFirstVariable;

Nous allons maintenant lui attribuer une valeur. Nous allons par exemple lui attribuer la valeur *rouge*.
Voici ce que cela nous donne :
$MyFirstVariable = 'rouge';

Voyons cela d'un peu plus près.
Tout d'abord, le nom que nous avons donné à notre variable respecte bien les critères obligatoires que nous avons vus plus haut.

Ensuite nous lui avons attribué une valeur, la chaîne de caractères *rouge*. Cette chaîne de caractères a été placée entre deux quotes, comme ceci : 'chaîne'.
Une petite précision, ici nous avons utilisé des simples quotes mais nous aurions également pu utiliser des doubles quotes. Le mot quote signifie guillemet.
Enfin, l'instruction se termine par un ";" (point virgule), comme toujours avec PHP.
Egalement, vous remarquerez qu'à aucun moment je n'ai dit que la variable était égale à quelque chose, j'ai plutôt dit qu'on lui affectait quelque chose. Cette notion est très importante.
Le signe égal n'est pas à interpréter comme le signe égal que vous interprétez en mathématiques. On dit que l'on affecte une valeur à une variable, mais la variable n'est en aucun cas égale (au sens mathématique du terme) à la valeur.
Maintenant que nous avons défini notre première variable en lui affectant la valeur *rouge*, nous allons voir comment exploiter cette variable dans un texte afin de pouvoir afficher le résultat sur notre page web.
Voilà ce que nous voulons voir s'afficher sur notre page web :

```
Ma voiture est rouge
```

Voici donc comment nous allons nous y prendre en PHP.

```
1.<?php
2.$MyFirstVariable = 'rouge';
3.?>
4.Ma voiture est <?php echo $MyfirstVariable; ?>
```

▪ **Explications**
Nous avons déclaré notre variable en lui affectant la valeur rouge, cela a été fait entre les balises PHP.
Ensuite nous sommes sortis du PHP pour écrire la phrase *Ma voiture est*.
Puis nous sommes retournés dans le PHP afin d'écrire le contenu de notre variable grâce à la fonction *echo*.

Comme vous l'avez maintenant compris, si nous changions de voiture et que celle-ci était bleue, il nous suffirait alors d'écrire :
$MyFirstVariable = 'bleue';

Evidemment, nous aurions en retour :

```
Ma voiture est bleue
```

Voilà donc ce qu'est une variable, un élément qui varie.

3.3. Les familles de variable

Nous allons ranger les différents types de variables en deux grandes familles, les variables qui contiennent des chaînes de caractères et les variables qui contiennent des chiffres.

3.3.1. Les variables qui contiennent des chaînes de caractères

Nous avons vu ce type de variables précédemment. Ce sont des variables à qui nous affectons un mot ou bien une phrase. Elles n'ont d'autre sens que celui que nous voulons bien leur donner.
Nous allons reprendre le tout premier exemple de ce chapitre et nous allons créer une variable **$nom** à qui nous allons affecter un mot :
$nom = 'Chris';

Ici la variable **$nom** contient la chaîne de caractères *Chris*. Nous allons écrire en PHP, la phrase ***Bonjour*** *chaîne de caractères*.
Voilà ce que cela nous donne.

```
1.<?php
2.$nom = 'Chris';
3.?>
4.Bonjour <?php echo $nom; ?>
```

Ici nous avons affecté la chaîne de caractères *Chris*, à la variable **$nom**.
Ensuite nous avons écrit la phrase *Bonjour valeur de la chaîne de caractères contenue dans la variable $nom*.
Lorsque nous affichons le résultat de ce code sur notre page web, nous obtenons ceci.

```
Bonjour Chris
```

Si nous souhaitions afficher la phrase suivante, en utilisant le même code PHP.

```
Bonjour tout le monde, comment allez-vous ?
```

Il nous suffirait alors simplement de déclarer une nouvelle chaîne de caractères à notre variable **$nom**. Comme ceci :
$nom = 'tout le monde, comment allez-vous ?';

▪ **Récapitulatif des chaînes caractères**
'Chris' est une chaîne de caractères, *'tout le monde, comment allez-vous ?'* en est une autre.

▪ **Quel est le point commun entre ces chaînes de caractères ?**
Si vous êtes observateur, vous avez pu constater qu'à chaque fois que nous avons affecté une chaîne de caractères à la variable **$nom**, cette chaîne de caractères est

entourée par des quotes. C'est précisément cela qui détermine que la valeur affectée à la variable est une chaîne de caractères.
Donc pour déclarer une variable contenant une chaîne de caractères, nous devrons entourer cette chaîne de caractères par des quotes.
Voici l'écriture précise de l'affectation d'une chaîne de caractères à une variable :
$variable = 'chaîne de caractères';

Maintenant que vous connaissez la règle, nous allons préciser quelques petits détails importants à connaître. Tout d'abord, nous pouvons utiliser des simples quotes ou bien des doubles quotes, comme ceci :
$variable = 'chaîne de caractères';
ou bien
$variable = ''chaîne de caractères'';

Ensuite, imaginons que nous souhaitions affecter la chaîne de caractères suivante à une variable : *l'ami*.
Par exemple, sur le même principe que ce que nous avons vu plus haut, nous souhaitons afficher à l'écran la phrase *Bonjour l'ami*.
En écrivant ceci :

```
1.<?php
2.$nom = 'l'ami';
3.?>
4.Bonjour <?php echo $nom; ?>
```

Nous appelons notre page web dans notre navigateur, voici ce que nous allons voir apparaître à l'écran :

```
Parse error: syntax error, unexpected 'ami' (T_STRING)
```

Pourquoi cette erreur ?
Tout à l'heure, nous avons donné la règle d'écriture de l'affectation d'une chaîne de caractères à une variable :
$variable = 'chaîne de caractères';

Et nous avons dit que la chaîne de caractères devait être entourée par des quotes. Cela signifie que la chaîne de caractères doit recevoir une quote d'ouverture et une quote de fermeture. Or en écrivant :
$nom = 'l'ami';

Nous avons trois quotes. Une quote d'ouverture, une quote de fermeture et une quote dans la chaîne de caractères. D'où l'erreur générée.
Afin de palier à cela, PHP nous offre deux possibilités.

- La première est d'entourer dans ce cas, la chaîne de caractères par des doubles quotes, comme ceci :
$nom = ''l'ami'';
La chaîne de caractères se retrouve ainsi avec une double quote d'ouverture et une double quote de fermeture, et elle ne contient qu'une simple quote.
- La seconde possibilité est de placer un anti-slash devant la quote se trouvant à l'intérieur de la chaîne de caractères. Comme ceci :
$nom = 'l\'ami';

3.3.2. Les variables qui contiennent des chiffres

La seule différence entre la famille des variables qui contiennent des chaînes de caractères et la famille des variables qui contiennent des chiffres est l'entourage par des quotes.
Voyons tout de suite quelques exemples.
$variable = 'deux'; // il s'agit d'une variable de la famille des chaînes de caractères, puisqu'elle est entourée par des quotes et en plus ce n'est pas un chiffre.
$variable = '2'; // est également une variable de la famille des chaînes de caractères, puisqu'elle est entourée par des quotes.
$variable = 2; // voilà une variable de la famille des chiffres, puisqu'elle n'est pas entourée par des quotes.

3.3.3. Différence entre ces deux familles de variable

Une variable de la famille des chiffres nous permettra :
- de pouvoir réaliser des calculs;
- d'afficher son contenu dans notre page web grâce à la fonction *echo*.

Alors qu'une variable de type chaîne de caractères nous permettra juste :
- d'afficher des mots ou bien des phrases sur notre page web, grâce à la fonction *echo*.

Voici un petit programme simple, permettant d'illustrer ce que nous venons d'apprendre.

```
1.<?php
2.$variable_1 = 'Je suis de la famille des chaînes de caractères';
3.$variable_2 = 2;
4.?>
5.<?php echo $variable_1; ?>
6.<br />
7.<?php echo $variable_2; ?>
```

Nous obtenons ceci à l'écran.

```
Je suis de la famille des chaînes de caractères
2
```