§ 1 Premiers pas

1-1 Entrées/sorties

Entrée (input) et formulaires HTML

En PHP, l'entrée standard est le formulaire HTML : Exemple de formulaire HTML. Pour faciliter la création de tels formulaires, le logiciel Créer un formulaire HTML et récupérer les données avec PHP est proposé en téléchargement.

Dans le but de simplifier cette première approche et de mettre l'accent moins sur le langage que sur la construction de programmes, nous reportons l'usage des formulaires HTML à plus tard. En attendant que cette entorse aux bonnes règles de programmation soit corrigée, les données seront directement intoduites au début du programme.

Traitement et sortie (output)

En PHP, le traitement est réalisé par un serveur distant qui, usuellement, retourne les résultats du traitement sous la forme d'une page HTML. La connaissance de quelques balises HTML est donc requise :

```
Texte suivi d'un retour à la ligne<br>
<b>Texte en caractères gras, suivi d'un retour à la ligne</b><br>
<i>Texte en caractères italiques, suivi d'un retour à la ligne</i><br>
<b><i>Texte en caractères gras et italiques, suivi d'un retour à la ligne</i></b><br>
<p>Texte constituant un paragraphe</p>
<p><b>Texte en caractères gras, constituant un paragraphe</b></p>
<p><i>Texte en caractères italiques, constituant un paragraphe</i></p>
<p><b><i>Texte en caractères gras et italiques, constituant un paragraphe</i></b></p>
```

Instruction echo

En langage PHP, l'instruction de sortie se nomme **echo**.

Exemple 1-1-1: bonjour

1-2 Variables, expressions, affectation

Variables

Une variable est un symbole qui désigne une place de la mémoire contenant de l'information (nombre, chaîne de caractères, tableau), par exemple

```
$r, $a, $rayon, $x1, $x2, etc.
```

Les noms des variables commencent par le symbole $ suivi d'une lettre, puis facultativement suivi par des lettres ou des chiffres.

Affectation

L'affectation est l'instruction par laquelle une valeur est attribuée à une variable, par exemple :

```
$a=12;
$t="Commentaire" ;
$x=3*$a+1 ;
```

Après quoi la place de mémoire nommée $a contient la valeur numérique 123.4, la place de mémoire nommée $t contient la chaîne de caractères « Commentaire » et la place de mémoire $x contient le résultat de l'opération 3*$a+1, à savoir 37.

Dans une affectation, le membre de gauche désigne une variable, tandis que le membre de droite désigne une expression qui possède une valeur. Attention, l'instruction

```
3*$x+1=$x
```

est incorrecte, car il n'existe pas de place de mémoire dont le nom est 3*$a+1.

Pour exécuter l'instruction suivante

```
$a=$a+5 ;
```

le membre de droite est d'abord évalué, ce qui donne 17, puis cette valeur est attribuée à la variable nommée dans le membre de gauche. Ainsi, l'ancienne valeur de $a est effacée et remplacée par la nouvelle valeur 17.

Manuel PHP

Dans le but de simplifier l'approche, le présent cours d'initiation se limite à l'essentiel. Par conséquent, il est très incomplet. Pour de plus amples informations sur le langage, prière de se référer au Manuel PHP.

Exemple 1-2-1 : aire du disque

Explications de l'exemple 1-2-1 :

- pi() désigne le nombre 3.14159... ;
- pow($r, 2) désigne le rayon élevé à la puissance 2 ;
- " Rayon du disque = ".$r."<;/p>" désigne la concaténation des trois chaînes de caractères suivantes (le symbole de la concaténation est le point) :
    - " Rayon du disque = "
    - $r
    - " "
- number_format($a, 3) indique que le nombre $a doit être affiché avec 3 décimales.

Exemple 1-2-2 : permutation des contenus de deux variables

Explications de l'exemple 1-2-2 :

- Une chaîne de caractères est délimitée par deux apostrophes 'texte' ou par deux guillemets "Texte". Les guillemets ont l'avantage de pouvoir contenir des chaînes de caractères qui contiennent un apostrophe, par exemple "L'hiver".
  Tandis que '$a' retourne le nom de la variable, "$a" retourne la valeur de la variable.
- L'instruction $b=$a ; a pour effet d'écraser la valeur de $b et de la remplacer par la valeur de $a.

1-3 Structure de contrôle conditionnelle if (condition) {action;}

Exemple 1-3-1 : Ajout des frais de port si le total des marchandises est inférieur à 100 euros

Exemple 1-3-2 : Codage en chiffres romains

1-4 Structures de contrôle alternatives if (condition) {action1;} else {action2;}

if (condition1) {action1;} elseif (condition2) {action2 } else {action3;}

Cas d'une alternative simple, exemple 1-4-1 :

Validation d'une donnée positive

Cas d'alternatives consécutives formant une sélection
Exemple 1-4-2 : Fonction définie par morceaux

f(x) = 0 pour $x \leq 0$
f(x) = x^2 pour $0<x<1$
f(x) = 2*x pour $x \geq 1$.

Cas d'alternatives imbriquées formant une arborescence
Exemple 1-4-3 : Résolution de l'équation du premier degré

Explications de l'exemple 1-4-3 : Il faut distinguer

d'une part, l'instruction d'assignation

$m = 0 qui a pour effet de donner la valeur 0 à la variable $x ;

d'autre part, l'**opérateur de comparaison**

$m ==0 qui est une expression booléenne, c'est-à-dire dont la valeur est *true* ou *false*, et qui représente la condition à laquelle s'applique le *if*.

1-5 Structure de contrôle sélective

Exemple 1-5-1 : Ecris le chiffre avec des lettres

Explications de l'exemple 1-5-1 : Il faut distinguer

la comparaison de nombres

$n = 4 ; if ($n == 0) {... } elseif ($n == 1) {..}
où (4 == 1) vaut *false* et (4 == 4) vaut *true ;*

la comparaison de chaînes de caractères

$n = '4' ; if ($n == '0') {... } elseif ($n == '1') {..}
où ('4' == '1') vaut *false* et ('4' == '4') vaut *true ;*

la comparaison mixte (les chaînes de caractères sont converties en nombres avant la comparaison) :

$n = 4 ; if ($n == '0') {... } elseif ($n == '1') {..}
où (4 == '1') vaut *false* et (4=='4') vaut *true.*

En PHP, les variables et les expressions ont un type qui dépend du contexte :

('x'==0) vaut *true* car, lorsqu'une chaîne de caractères ne représente pas un nombre valide, sa valeur numérique est 0 ;
('x'=='0') vaut *false* car les deux chaînes de caractères sont différentes ;
(''==0) vaut *true* car la valeur numérique de la chaîne vide est 0 ;
(''=='0') vaut *false* car les deux chaînes de caractères sont différentes ;
(x==0) vaut *true* car la valeur numérique du symbole x est 0 *;*
(x=='0') vaut *false* car ce sont deux chaînes de caractères qui sont comparées.