PHP – COURS 02 – Les structures conditionnelles

On a souvent besoin d'afficher des choses différentes en fonction de certaines données. Par exemple, si c'est le matin, vous voudrez dire "bonjour" à votre visiteur, si c'est le soir il vaudrait mieux dire "bonsoir". C'est là qu'interviennent les structures conditionnelles. Elles permettent de donner des ordres différents à PHP selon le cas.

LES SYMBOLES À CONNAÎTRE

$a == $b	égalité	rend vrai si $a est égal à $b
$a != $b	différence	rend vrai si $a est différent de $b
$a <= $b	inférieur ou égal	rend vrai si $a est inférieur ou égal à $b
$a < $b	inférieur	rend vrai si $a est strictement inférieur à $b
$a >= $b	supérieur ou égal	rend vrai si $a est supérieur ou égal à $b
$a > $b	supérieur	rend vrai si $a est strictement supérieur à $b

LA STRUCTURE IF...ELSE

Voici ce qu'on doit mettre dans l'ordre pour utiliser une condition :

- Pour introduire une condition, on utilise le mot "If", qui en anglais signifie "Si".
- On ajoute à la suite entre parenthèses la condition .
- Enfin, on ouvre des accolades à l'intérieur desquelles on mettra les instructions à exécuter si la condition est remplie.

Exemple1 :

```
<?php
  $age = 22;
  if ($age < 18){ // SI l'âge est strictement inférieur  à 18
      echo ("Salut gamin ! c'est un site pour les adultes \n");
      $autorisation = "non";
    }
  else {// SINON
    echo ( "Bonjour, bienvenue sur mon site \n");
    $autorisation = "oui";
  }
  echo ("Avez-vous l'autorisation d'entrer ? La réponse est : $autorisation \n");
?>
```

Exemple 2 :

```
<?php
$a=10;
$b=20;
if ($a > $b)
    echo ("la variable a est plus grande que la variable b \n");
else
    echo ("la variable a est plus petite que la variable b \n");
?>
```

Remarque 1 :

Vous devez mettre des accolades { } lorsque vous avez un bloc d'instructions, c'est à dire plusieurs instructions à exécuter dans le cas contraire (exemple 2) les accolades sont facultatives.

Remarque 2 :

Vous pouvez utiliser l'instruction IF sans le ELSE

```
if ($a > 0)
   echo ("valeur positive \n");
```

LA STRUCTURE IF...ELSEIF....ELSE

```
if($moyenne >= 10)
    echo (" vous avez votre BAC");
elseif($moyenne >= 8)
   echo ("vous allez au rattrapage");
else
    echo (" à l'année prochaine");
```

le mot-clé "elseif" signifie "Sinon si". Dans l'ordre, PHP rencontre les conditions suivantes :

1. Si ....
2. Sinon si... (vous pouvez avoir plusieurs elseif)
3. Sinon,

DES CONDITIONS MULTIPLES

Pour exécuter une instruction on peut avoir plusieurs conditions à la fois. Pour cela, on aura besoin de nouveaux mots-clés. Voici les principaux à connaître :

Mots clés	Signification	Symbole	Exemple
AND	ET	&&	if($age>18 && $ville=="melun" && $sexe=="M")
OR	OU	||	if ($dep==77 || $dep==91)

LE CAS DES BOOLÉENS

Les booléens sont des variables qui valent soit **true** (vrai) soit **false** (faux). Les booléens sont particulièrement utiles avec les conditions ! Voici comment on teste une variable booléenne :

```
$autorisation=true;
if($autorisation)
    echo (" accès autorisé");
else
    echo ("accès refusé");
```

Un des avantages des booléens, c'est que vous n'êtes pas obligés d'ajouter le == true. Quand on écrit – *if ($autorisation)* - , PHP comprend : si la variable $autorisation est égale à **true**.

De la même manière, - *if (! $autorisation)* - , PHP comprend : si la variable $autorisation est égale à **false**.

STRUCTURE D'UN TEST

Il n'y a que deux formes possibles pour un test ; la première est la plus simple, la seconde la plus complexe.

```
Si booléen Alors
    Instructions
Finsi
```

```
Si booléen Alors
  Instructions 1
Sinon
  Instructions 2
Finsi
```

Un booléen est une expression dont la valeur est VRAI ou FAUX. Cela peut donc être :

- une variable (ou une expression) de type booléen
- une condition

La structure d'un test est relativement claire. Dans la forme la plus simple, arrivé à la première ligne (Si… Alors) la machine examine la valeur du booléen. Si ce booléen a pour valeur VRAI, elle exécute la série d'instructions. En revanche, dans le cas où le booléen est faux, l'ordinateur saute directement aux instructions situées après le FinSi.

Dans le cas de la structure complexe, c'est à peine plus compliqué. Dans le cas où le booléen est VRAI, et après avoir exécuté la série d'instructions 1, au moment où elle arrive au mot « Sinon », la machine saute directement à la première instruction située après le « Finsi ». De même, au cas où le booléen a comme valeur « Faux », la machine saute directement à la première ligne située après le « Sinon » et exécute l'ensemble des « instructions 2 ». Dans tous les cas, les instructions situées juste après le FinSi seront exécutées normalement.

Une condition est une comparaison, cela signifie qu'une condition est composée de trois éléments :

- une valeur
- un opérateur de comparaison
- une autre valeur

```
if($moyenne >= 10)
    echo (" vous avez votre BAC");
```

Dans cet exemple, on indique que si la condition $moyenne est supérieure ou égale à 10 est VRAIE, on exécute l'instruction : echo (" vous avez votre BAC"). Dans le cas contraire on passe à l'instruction suivante.

Un test ouvre donc deux voies, correspondant à deux traitements différents. Mais il y a des tas de situations où deux voies ne suffisent pas. Par exemple, un programme devant donner l'état de l'eau selon sa température doit pouvoir choisir entre trois réponses possibles

- glace si température inférieure à 0
- liquide si température supérieure à 0 et inférieure à 100
- gazeuse si température supérieure à 100

Une première solution serait la suivante :

```
if ($temperature < 0)
        echo " c'est de la glace \n";
if ($temperature > 0 AND $temperature < 100)
        echo " c'est du liquide \n";
if ($temperature >= 100)
        echo " c'est gazeux \n" ;
```

Vous constaterez que c'est un peu laborieux. Les conditions se ressemblent plus ou moins, et surtout on oblige la machine à examiner trois tests successifs alors que tous portent sur une même chose, la température de l'eau. Il serait ainsi bien plus rationnel d'imbriquer les tests de cette manière :

```
if ($temperature < 0)
   echo " c'est de la glace \n";

 else {
   if ($temperature > 100)
     echo " c'est gazeux \n";
   else
     echo " c'est du liquide \n";
 }
```

Nous avons fait des économies : au lieu de devoir taper trois conditions, dont une composée, nous n'avons plus que deux conditions simples. Mais aussi, et surtout, nous avons fait des économies sur le temps d'exécution de l'ordinateur. Si la température est inférieure à zéro, celui-ci écrit dorénavant « C'est de la glace » et passe directement à la fin, sans être ralenti par l'examen d'autres possibilités (qui sont forcément fausses).
Cette deuxième version n'est donc pas seulement plus simple à écrire et plus lisible, elle est également plus performante à l'exécution.

Remarque : lorsqu'on écrit un programme informatique, il faut avant tout avoir un minimum de réflexion et essayer de trouver la structure la plus claire à comprendre. Il ne faut pas hésiter à faire des jeux d'essais avec quelques variables afin de tester la validité du code

Le code suivant est-il valide ?

```
if ($moyenne >= 10)
   echo " Vous avez votre examen \n";
elseif ($moyenne >= 12)
     echo " Vous avez votre examen avec une mention Assez Bien \n";
```