

Chapitre 4

Les fonctions et structures de contrôle

1. Les conditions

1.1 if

Les instructions de type condition permettent d'exécuter du code si une condition est vraie.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue" :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bienvenue';
}
?>
```

Ci-dessus, la valeur Robert est affectée à la variable \$prenom puis cette variable est testée à l'aide de l'instruction **if**.

La syntaxe est donc :

```
if (condition) { instruction }
```

Notez que :

- "est égal à" se note `==`.
- "est différent de" se note `!=`.
- "est inférieur à" se note `<`.
- "est supérieur à" se note `>`.
- "est inférieur ou égal à" se note `<=`.
- "est supérieur ou égal à" se note `>=`.

L'instruction **sinon** se note **else**.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue", sinon "A bientôt" :

```
<?php
$prenom = 'Toto'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bienvenue';
}
else
{
    echo 'A bientôt';
}
?>
```

Affiche :

A bientôt

En effet, le code teste si la variable `$prenom` est égale à Robert, puis comme ce n'est pas le cas, le code passe dans le **else** (sinon) et exécute `echo "A bientôt";`.

Enfin la dernière instruction pour les conditions est **else if** appelée **sinon si**.

Cela permet de tester d'autres conditions non testées par le **if**.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue", sinon s'il est égal à Toto, afficher "Bonjour", sinon "A bientôt" :

```
<?php
$prenom = 'Toto'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bienvenue';
}
else if ($prenom == 'Toto') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

Affiche :

Bonjour

Le code teste si \$prenom est égal à Robert puis si \$prenom est égal à Toto. Et comme \$prenom est effectivement égal à Toto, le code exécute echo "Bonjour";.

Il est possible d'ajouter autant d'instructions **else if** que vous voulez.

Si la condition (`$prenom == "Robert"`) est vérifiée, les autres conditions dans les **else if** ne sont même pas vérifiées, donc si vous passez dans le **if**, vous êtes sûr de ne jamais passer dans le ou les **else if** ainsi que dans le **else**.

À l'inverse, si vous écrivez :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
```

```
{
    echo 'Bienvenue';
}

if ($prenom == 'Toto') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

Le code teste si \$prenom est égal à Robert puis si \$prenom est égal à Toto et comme \$prenom est initialisé avec la valeur Robert, ce code affiche "Bienvenue" et "A bientôt".

Vous pouvez aussi imbriquer les **if** les uns à l'intérieur des autres autant de fois que vous le souhaitez.

Par exemple :

```
<?php

$age = 30; //déclaration de la variable $age

if ($age > 20) //test de la variable $age
{
    if ($age == 30) //test de la variable $age
    {
        echo 'Bienvenue';
    }
    else {
        echo 'A bientôt';
    }
}
?>
```

Affiche :

Bienvenue

Le programme teste si la variable `$age` est supérieure à 20 puis comme c'est le cas, le programme teste ensuite si `$age` est égal à 30 et exécute `echo "Bienvenue";`.

Enfin, vous pouvez intercaler du code HTML entre les conditions en PHP.

Par exemple :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    ?>
    Bonjour <!--Code HTML-->
<?php
}
else
{
    ?>
    A bientôt <!--Code HTML-->
<?php
}
?>
```

est équivalent à :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

1.2 switch

L'instruction **switch** est équivalente au **if** mais elle est utilisée par le développeur pour plus de clarté dans le code. Elle est équivalente au **if** si vous utilisez un **break** pour sortir du **switch**. Sinon toutes les instructions qui suivent le **case** dans lequel vous êtes entré seront exécutées.

La syntaxe est :

```
switch (condition) {  
    case expression: instruction  
    case expression: instruction  
    ...  
}
```

```
<?php  
  
$prenom = 'Robert'; //déclaration de la variable $prenom  
  
switch ($prenom) //test de la variable $prenom  
{  
    case 'Robert':  
        echo 'Bonjour';  
        break;  
    case 'Jean':  
        echo 'A bientôt';  
        break;  
}  
?>
```

L'instruction **break** provoque la sortie du **switch** et donc si **\$prenom** est égal à "Robert", le code exécutera `echo "Bonjour";` puis **break** et sortira du **switch** sans même tester "Jean".

Cela équivaut à utiliser l'instruction **else if** à la place de **if** :

```
<?php  
  
$prenom = 'Robert'; //déclaration de la variable $prenom  
  
if ($prenom == 'Robert') //test de la variable $prenom  
{  
    echo 'Bonjour';  
}
```