



**BTS SIO**

# Séquence 3

## La boucle while

Franck LAMY – BTS SIO1



## LE COURS



## Définition >>>



**Une structure de contrôle de boucle permet d'exécuter de manière itérative (en boucle) certaines parties du code (bloc de code) tant qu'une condition est vérifiée**



**Nécessité d'exécuter plusieurs fois à la suite un même code**



**Une boucle va permettre de n'écrire ce code, à exécuter plusieurs fois, qu'une seule fois**



## Principales boucles »»



**La boucle while – "tant que"**



**La boucle for – "pour"**



**La boucle do...while – "faire...tant que"**



**La boucle foreach – "pour chaque"**

# Exemple

Exercice ➞

compte.php

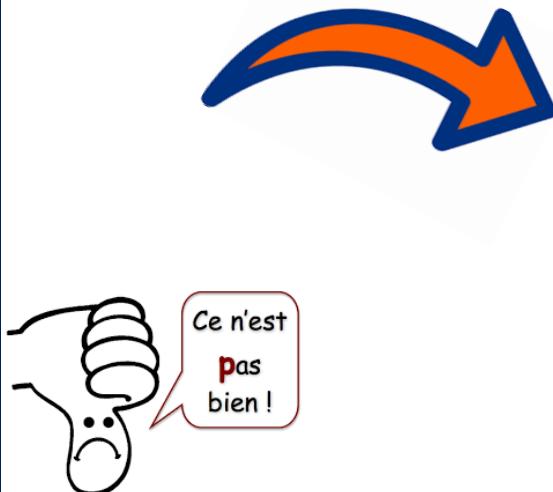


1 2 3 4 5 6 7 8

EXAMPLE



```
echo 1 . '';  
echo 2 . '';  
echo 3 . '';  
echo 4 . '';  
echo 5 . '';  
echo 6 . '';  
echo 7 . '';  
echo 8 . '';
```



On répète plusieurs fois la même instruction

*echo un\_nombre . '';*

Imaginez que l'on veuille compter jusqu'à 100 !



# La boucle

**WHILE**



## Définition &gt;&gt;&gt;



La boucle **while** va permettre d'exécuter un bloc d'instructions **TANT QU'UNE CONDITION EST VRAI**

**VRAI**

```
while (condition) {  
    // instruction(s)  
}
```

**TANT QUE** la condition est **VRAI**  
les instructions sont exécutées

# La boucle while

Exercice ➞

compte.php



```
echo 1 . ' ' ;  
echo 2 . ' ' ;  
echo 3 . ' ' ;  
echo 4 . ' ' ;  
echo 5 . ' ' ;  
echo 6 . ' ' ;  
echo 7 . ' ' ;  
echo 8 . ' ' ;
```

## Avec une boucle **while**

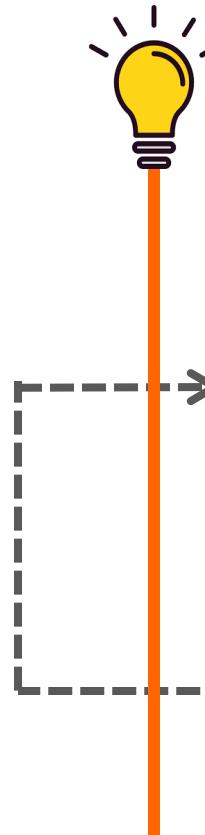
```
$nombre = 1;  
while ($nombre <= 8) {  
    echo $nombre . ' ' ;  
    $nombre = $nombre+1;  
}
```



```
$nombre = 1;  
while ($nombre <= 8) {  
    echo $nombre . ' ';  
    $nombre = $nombre + 1;  
}
```

On répète ce code  
**TANT QUE**  
la condition est

VRAI



On initialise \$nombre à 1

On évalue la condition

Si la condition est VRAI

- On affiche \$nombre
- On modifie \$nombre  
(ici on passe au nombre suivant)
- On évalue à nouveau la condition



## Itération »»

// **initialisation avant évaluation condition**  
**while (condition) {**

// **instruction(s)**  
// **modification avant évaluation**

}

Corps de la boucle



Itération

# La boucle while

Itération »»

```
$nombre = 1 ;  
while ($nombre <= 8 ) {  
    echo $nombre . ' ' ;  
    $nombre = $nombre+1;  
}  
Corps de la boucle
```



Itération  
(ici 8 itérations)



```
$nombre = 1 ;  
while ($nombre <= 8 ) {  
    echo $nombre . ' ' ;  
}
```



## Boucle infinie



**La variable \$nombre est toujours égale à 1 !**



**Il faut penser à modifier la variable \$nombre afin que la condition soit vérifiée à nouveau avec une valeur différente**





## Incrémation »»



**L'incrémation est l'opération qui consiste à ajouter une valeur à une variable.**

**\$variable = \$variable + N**



**\$variable += N**

```
$nombre = 1 ;  
while ($nombre <= 8) {  
    echo $nombre . ' ' ;  
    $nombre = $nombre+1;  
}
```



```
$nombre = 1;  
while ($nombre <= 8) {  
    echo $nombre . ' ' ;  
    $nombre += 1 ;  
}
```



```
// initialisation  
while (condition) {  
    // instruction(s)  
    // modification (incrémantation)  
}  
// instruction(s)
```



Exécutées **tant que** la condition est

**VRAI**



Exécutées lorsque la condition est **FAUX**



On **sor**t de la boucle



## Break »»

```
// initialisation  
while (condition) {  
    // instruction(s)  
    if (condition-if) {  
        // instructions-if  
        break;  
    }  
    // modification  
}  
// instruction(s)
```



L'instruction **break** permet de **sortir** de la **boucle while** de manière **prématurée**



## Plusieurs conditions ➞

// *initialisation*

**while (condition1 && condition2) {**

// *instruction(s)*

// *modification*

}

// *instruction(s)*

ET

// *initialisation*

**while (condition1 || condition2) {**

// *instruction(s)*

// *modification*

}

// *instruction(s)*

OU



**Application du théorème de De Morgan afin d'évaluer la sortie de la boucle**

