

BOUCLES ET CONDITIONS

LES BOUCLES

- Une boucle est une structure répétitive qui permet d'exécuter plusieurs fois les instructions se trouvant à l'intérieur du bloc d'instructions
- Le concept des boucles est fondamental à maîtriser car on le retrouve dans absolument **TOUS** les langages de programmations

LA BOUCLE WHILE

- Les instructions de bloc d'instructions de cette boucle sont répétées tant que la condition de la boucle est **TRUE**. Si la condition est **FALSE**, on sort de la boucle

```
while (<condition>
{
    #bloc d'instructions
}
```

- Exemple :

```
$nombre = 0

while ($nombre -lt 10)
{
    Write-Host $nombre
    $nombre++
}
```

LA BOUCLE DO-WHILE

- La boucle Do-While ressemble à la boucle While, sauf que la condition est évaluée à la fin.

```
Do
{
    #bloc d'instructions
}
while (<condition>)
```

- Exemple :

```
Do
{
    [int]$var = Read-Host "Entrez une valeur entre 0 et 10"
}
while( ($var -lt 0) -or ($var -gt 10) )
```

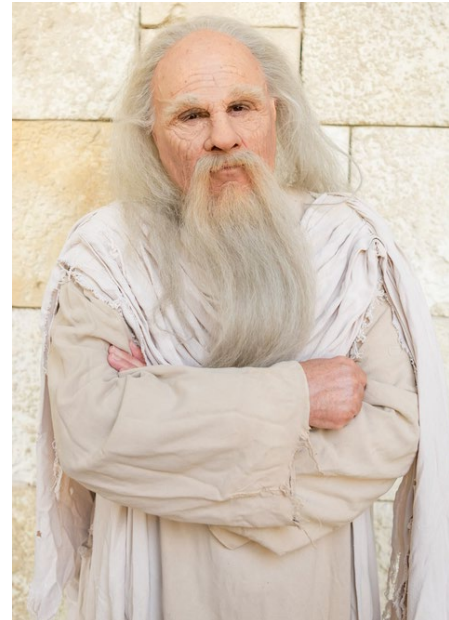
LA BOUCLE FOR (À NE PAS CONFONDRE AVEC FORT BOYARD)

- La boucle For permet d'exécuter un certain nombre de fois un bloc d'instructions. Il faut indiquer la valeur initiale, la condition de répétition et la valeur de l'incrément.

```
For (<initial>;<condition>;<increment>)  
{  
    #bloc d'instructions  
}
```

- Exemple :

```
For ($var = 0;$var -lt 10;$var++)  
{  
    write-host $var  
}
```



LES CONDITIONS

- Les structures conditionnelles permettent, grâce à l'évaluation de conditions, de diriger l'exécution de votre script vers un bloc d'instructions ou un autre.
- Le concept des conditions est également fondamental à maîtriser car on le retrouve dans absolument TOUS les langages de programmations

LA CONDITION IF

- La structure de la condition If est la suivante:

```
If (<condition>)  
{  
    #bloc d'instructions  
}
```

- Exemple :

```
$var = Read-Host "Entrez un caractère"  
If ($var -eq 'A')  
{  
    Write-Host "Le caractère saisi par l'utilisateur est un 'A'"  
}
```

LA CONDITION IF, ELSE

```
[int]$var1 = Read-Host "Saisissez un nombre"  
[int]$var2 = Read-Host "Saisissez un nombre"
```

```
If ($var1 -ge $var2)  
{  
    Write-Host $var1 "est plus grand ou égal à" $var2  
}  
Else  
{  
    Write-Host $var1 "est plus petit que" $var2  
}
```


LA CONDITION IF, ELSE, ELSEIF

```
[int]$var = Read-Host "Entrez une valeur : 1, 2 ou 3"

If ($var -eq 1)
{
    Write-Host "La valeur saisie est le 1"
}
Elseif ($var -eq 2)
{
    Write-Host "La valeur saisie est le 2"
}
Elseif ($var -eq 3)
{
    Write-Host "La valeur saisie est le 3"
}
```

LA CONDITION SWITCH

- La condition Switch permet de remplacer toute une série de Elseif et de n'avoir **qu'une seule condition**, qui va diriger vers une multitude de bloc d'instructions.

```
Switch (<expression>)  
{  
    <valeur_1> { <#bloc d'instructions 1#> ; Break }  
    <valeur_2> { <#bloc d'instructions 2#> ; Break }  
    <valeur_3> { <#bloc d'instructions 3#> ; Break }  
    <valeur_4> { <#bloc d'instructions 4#> ; Break }  
    <valeur_5> { <#bloc d'instructions 5#> ; Break }  
    Default { <#bloc d'instructions 6#> }  
}
```

LA CONDITION SWITCH

```
$var = Read-Host "Saisissez une valeur entre 1 et 5"
```

```
Switch ($var)
```

```
{  
    1 { Write-Host "Vous avez saisi la valeur 1" ; Break }  
    2 { Write-Host "Vous avez saisi la valeur 2" ; Break }  
    3 { Write-Host "Vous avez saisi la valeur 3" ; Break }  
    4 { Write-Host "Vous avez saisi la valeur 4" ; Break }  
    5 { Write-Host "Vous avez saisi la valeur 5" ; Break }  
    Default { Write-Host "Valeur saisie incorrect" }  
}
```