

# Les boucles (deuxième partie)



# Contenu

- **Motivation**
- **La boucle For**
- **La boucle ForEach**
- **Retour sur la mise en situation pour les tableaux**

# Motivation

# Motivation

- **Nous avons vu les boucles While, Do While et Do Until**
- **Nous avons vu les tableaux**
- **Nous avons vu comment itérer sur les éléments d'un tableau à l'aide d'une boucle While**

# Motivation

- **Nous allons maintenant voir des types de boucles permettant d'itérer plus facilement sur les éléments d'un tableau**

# La boucle For

# La boucle For

- **Permet d'exécuter un bloc de code X fois**
- **Donne accès au « numéro » (indice) de l'itération en cours**

# Syntaxe

**for (\$i = indice de départ; condition pour continuer; opération d'incrément) {**

**# bloc de code ayant accès à \$i**

**}**

## Exemples:

```
# Afficher les nombres de 1 à 5
```

```
for ($i = 1; $i -le 5; $i++) {
```

```
    Write-Output $i
```

```
}
```

```
# Itérer sur les éléments d'un tableau
```

```
for ($i = 0; $i -lt $monTableau.length; $i++ ) {
```

```
    Write-Output $monTableau[$i]
```

```
}
```

# La boucle ForEach

# La boucle ForEach

- **Permet d'itérer directement sur les éléments d'un tableau, sans même utiliser leurs indices!**

# Syntaxe

```
foreach ($element in $monTableau) {  
    # bloc de code ayant accès à $element  
}
```

## Exemple:

```
foreach ($element in $monTableau) {  
    Write-Output $element  
}
```

# Retour sur la mise en situation pour les tableaux

# Mise en situation

- **Je veux:**

- demander à l'utilisateur d'entrer 100 nombres
- vérifier si tous les nombres entrés sont uniques



Image par Drew Beamer sur Unsplash

# Solution, partie 1

```
1 Set-Variable -Name NOMBRE_DE_NOMBRES -Value 100 -Option Constant
2
3 [int[]]$nombres = @()
4
5 for ($i = 0; $i -lt $NOMBRE_DE_NOMBRES; $i++) {
6     $nombres += Read-Host -Prompt "Entrer le nombre #$i"
7 }
```

# Solution, partie 2

```
9  $sontTousUniques = $true
10 for ($i = 0; $i -lt $NOMBRE_DE_NOMBRES - 1; $i++) {
11     for ($j = $i + 1; $j -lt $NOMBRE_DE_NOMBRES; $j++) {
12         $sontTousUniques = $sontTousUniques -and $nombres[$i] -ne $nombres[$j]
13         if (-not $sontTousUniques) {
14             break
15         }
16     }
17     if (-not $sontTousUniques) {
18         break
19     }
20 }
```

## Solution, partie 3

```
24  if ($sontTousUniques) {  
25  |  Write-Output "Les nombres sont tous uniques"  
26  } else {  
27  |  Write-Output "Les nombres ne sont pas tous uniques"  
28  }
```

# Fin de la présentation

Des questions?

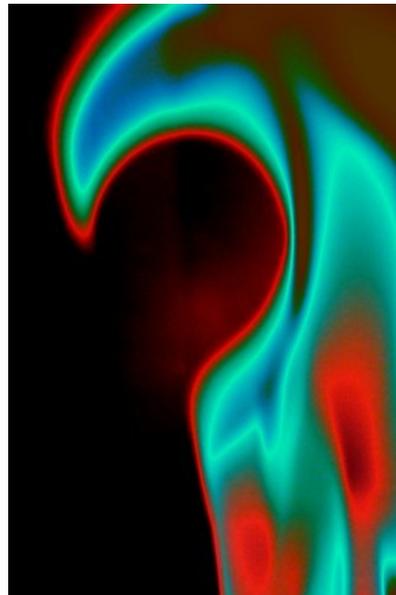


Image par limecools (CC BY-NC-SA 2.0)