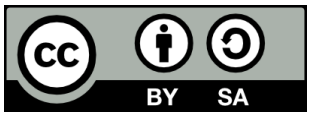
Créer des objets en PowerShell



Contenu

- **Rappels**
- **Le cmdlet New-Object**
- **Créer des objets personnalisés**
- **Créer ses propres classes**

Rappels

Rappel: Les objets

- **Un objet possède des propriétés et des méthodes**
- **Les objets en PowerShell peuvent provenir de différentes technologies, ex :**
  - .NET
  - COM
- **PSObject fournit une représentation commune pour tous les objets, peu importe leur provenance**

Rappel: Les classes

- **En programmation orientée objet, une classe est un type d'objet**
  - Définit la liste des propriétés communes à tous les objets de ce type
  - Définit le code des méthodes de ces objets

Rappel: Les classes

- **On dit d’un objet appartenant à une classe qu’il est une instance de cet objet**

Le cmdlet New-Object

Le cmdlet New-Object

- **Le cmdlet New-Object permet de créer soit :**
  - Une instance d'une classe .NET
  - Un objet COM

Création d’un objet .NET

Syntaxe:

New-Object -TypeName Nom du type -ArgumentList arguments d’initialisation

Exemple

```
PS /home/plbrault/temp/PowerShell> New-Object -TypeName DateTime -ArgumentList 2020, 06, 21, 14, 13, 52


dimanche 21 juin 2020 14 h13 min52 s

PS /home/plbrault/temp/PowerShell>
```

Création d’un objet COM

Syntaxe:

New-Object -ComObject nom de l’objet COM -Property propriétés de l’objet COM

Exemple

Ce script crée un objet COM qui permet d'interagir avec Windows, puis s'en sert pour simuler une saisie au clavier

```
$wshell = New-Object -ComObject wscript.shell
$wshell.SendKeys('Bonjour')
```

Créer des objets personnalisés

Créer des objets personnalisés

- **Pour créer un objet personnalisé, on utilise New-Object pour créer un objet .NET de type PSObject!**
- **On utilise ensuite Add-Member pour lui ajouter des propriétés et des méthodes**

Exemple

Création de l’objet:

```
$personne = New-Object -TypeName PSObject
```

Exemple

Ajout d’une propriété « nom » puis d’une propriété « dateNaissance »:

```
Add-Member -InputObject $personne -MemberType NoteProperty -Name nom -Value "William Byers"
```

```
Add-Member -InputObject $personne -MemberType NoteProperty -Name dateNaissance -Value (Get-Date "1971 3 22")
```

Exemple

Ajout d’une méthode « calculerAge » à l’objet :

```
$calculerAge = {
  $maintenant = Get-Date
  $age = $maintenant - $this.dateNaissance
  return [math]::floor($age.Days / 365)
}

Add-Member -InputObject $personne -MemberType ScriptMethod -Name "calculerAge" -Value $calculerAge
```

Objets vs Tableaux associatifs

- **Pourquoi créer des objets personnalisés plutôt qu'utiliser des tableaux associatifs?**
  - Un tableau d'objets est plus facile à manipuler qu'un *tableau de tableaux associatifs*
  - Comme on vient de le voir, un objet peut avoir des méthodes qui utilisent ses propriétés (il peut aussi les modifier!)

Créer ses propres classes

Créer ses propres classes

```
class Personne {
  # Propriétés
  [string]$nom
  [datetime]$dateNaissance

  # Constructeur
  Personne([string] $nom, [string] $dateNaissance) {
    $this.nom = $nom
    $this.dateNaissance = Get-Date $dateNaissance
  }

  # Méthodes
  [int]calculerAge() {
    $maintenant = Get-Date
    $age = $maintenant - $this.dateNaissance
    return [math]::floor($age.Days / 365)
  }
}

# Instanciation
$willByers = New-Object -TypeName Personne -ArgumentList "William Byers", "1971-03-22"
```

Fin de la présentation

Des questions?

Photo par Jon Tyson sur Unsplash