

# Valider les saisies de l'utilisateur



# Contenu

- **Motivation**
- **Les erreurs de typage**
- **Gestion gracieuse des erreurs**
- **Autres validations**
- **Pour la suite du cours**

# Motivation

## Éviter ce genre de situations:

```
> test.ps1
1 [int]$nombre = Read-Host "Entrer un nombre"
2
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PS /home/plbrault/temp/powershell> /home/plbrault/temp/powershell/test.ps1
Entrer un nombre: allo
MetadataError: /home/plbrault/temp/powershell/test.ps1:1:1
Line |
  1 | [int]$nombre = Read-Host "Entrer un nombre"
     | ~~~~~
     | Cannot convert value "allo" to type "System.Int32". Error: "Input string was not in a correct format."
```

# Les erreurs de typage

- **Dans l'exemple précédent, la valeur saisie par l'utilisateur cause une **erreur****
- **Les erreurs ont généralement pour effet de mettre fin au programme**
  - Ce comportement n'est pas souhaitable
  - On veut plutôt détecter les erreurs et les gérer **gracieusement**

# Gestion gracieuse des erreurs

- **Un bloc Try-Catch permet de détecter qu'une erreur est survenue, et de réagir en conséquence au lieu d'arrêter le programme**

# Gestion gracieuse des erreurs

```
try {  
    [int]$nombre = Read-Host "Entrer un nombre"  
} catch {  
    Write-Output "La valeur que vous avez entrée n'est pas un nombre."  
}
```

**Si une erreur est détectée à l'intérieur du « try », l'exécution saute au « catch ».**

# Gestion gracieuse des erreurs

```
[object]$nombre = $null
```

```
while ($nombre -eq $null) {
```

```
  try {
```

```
    [int]$nombre = Read-Host "Entrer un nombre"
```

```
  } catch {
```

```
    Write-Output "La valeur que vous avez entrée n'est pas un nombre."
```

```
  }
```

```
}
```

# Autres validations

- **Valider que la donnée saisie par l'utilisateur est du bon type n'est parfois pas suffisant**
- **Ex: on demande à l'utilisateur d'entrer un pourcentage**
  - Doit être un nombre
  - Doit être entre 0 et 100
- **On doit ajouter un « if » pour s'assurer que la deuxième condition est rencontrée**

# Pour la suite du cours

- **Pour la suite du cours, vos scripts devront toujours valider les données fournies par l'utilisateur**
  - Validation du type avec une gestion gracieuse des erreurs
  - Validation des autres critères à respecter

# Fin de la présentation

Des questions?



Photo par Emily Morter sur Unsplash