La portée des variables en PowerShell

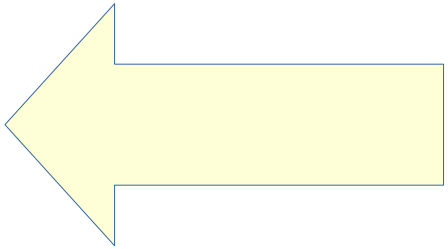
La portée des variables

- **La portée (*scope* en anglais) d'une variable désigne les endroits dans le code où cette variable est visible**

La portée des variables

$x = 42

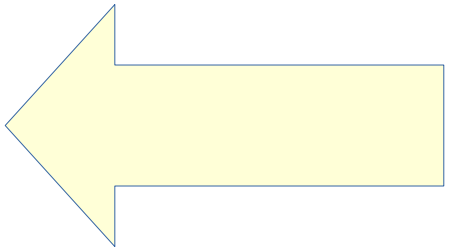
Write-Output $x



La variable $x avec la valeur 42 est visible ici

La portée des variables

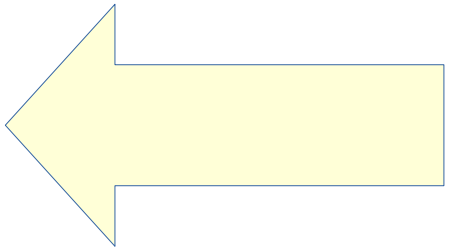
Write-Output $x



N’affiche rien (variable non initialisée)

$x = 42

Write-Output $x

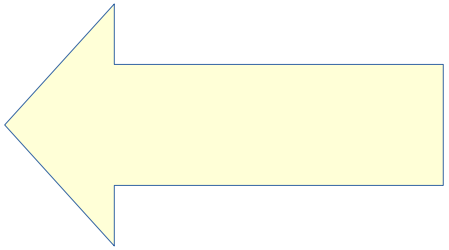


Affiche « 42 »

La portée des variables

$x = 42

if (expression) {



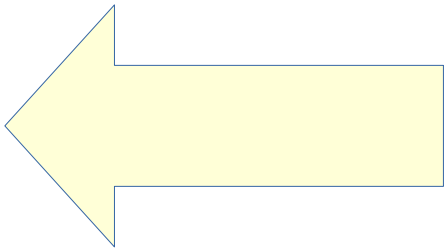
$x est visible ici

}

La portée des variables

```
if (expression) {
    $x = 42
}

# suite du code
```



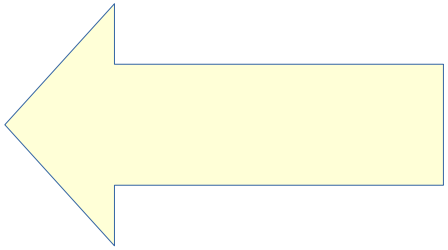
$x est visible ici

- **Un peu contre-intuitif à la lecture du code**
- **Il vaut donc mieux déclarer $x avant le if si on veut s'en servir à l'extérieur de celui-ci**

La portée des variables

```
while (expression) {
    $x = 42
}

# suite du code
```



$x est visible ici

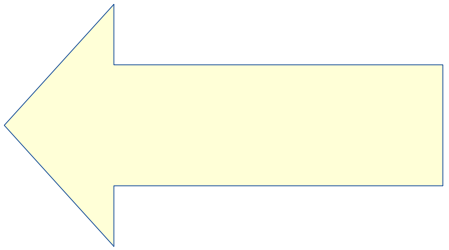
- **Même chose pour les autres types de boucles (Do-While, Do-Until, for, etc)**
- **Un peu contre-intuitif à la lecture du code**
- **Il vaut donc mieux déclarer $x avant le if si on veut s'en servir à l'extérieur de celui-ci**

La portée des variables

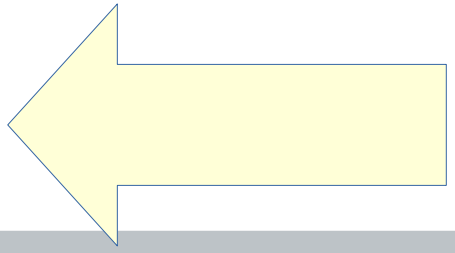
```
$x = 54

function maFonction() {
    Write-Output $x
    $y = 42
}

maFonction
# suite du code
```



$x est visible ici



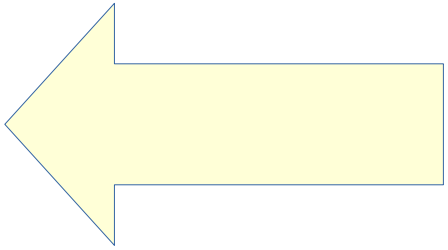
$y n'est **PAS** visible ici!

La portée des variables

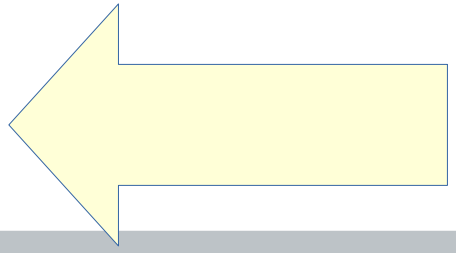
```
$x = 54

function maFonction() {
    $x = 42
    Write-Output $x
}

maFonction
Write-Output $x
```



Affiche 42



Affiche 54!

La portée des fonctions

- **Les fonctions ont aussi une portée**

La portée des fonctions

```
function maFonction1 {
}
```

maFonction2 peut être appelée ici, même si elle est déclarée après maFonction1

```
function maFonction2 {
    function maFonction3 {
    }
    # suite de la fonction
}
# suite du code
```

maFonction1 et maFonction3 peuvent être appelées ici

maFonction1 et maFonction2 peuvent être appelées ici, mais **PAS** maFonction3

Fin de la présentation

Des questions?

Photo par Emily Morter sur Unsplash