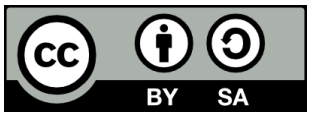
Standards de codage



Contenu

- **Indentation et espacement**
- **Position des accolades**
- **Noms de variables**
- **Guide de style de PowerShell**

Indentation et espacement

Indentation

- **Lorsqu'on écrit du code, il est important de respecter certaines règles pour que celui-ci soit lisible**
- **La règle la plus importante est de bien indenter son code**

Indentation

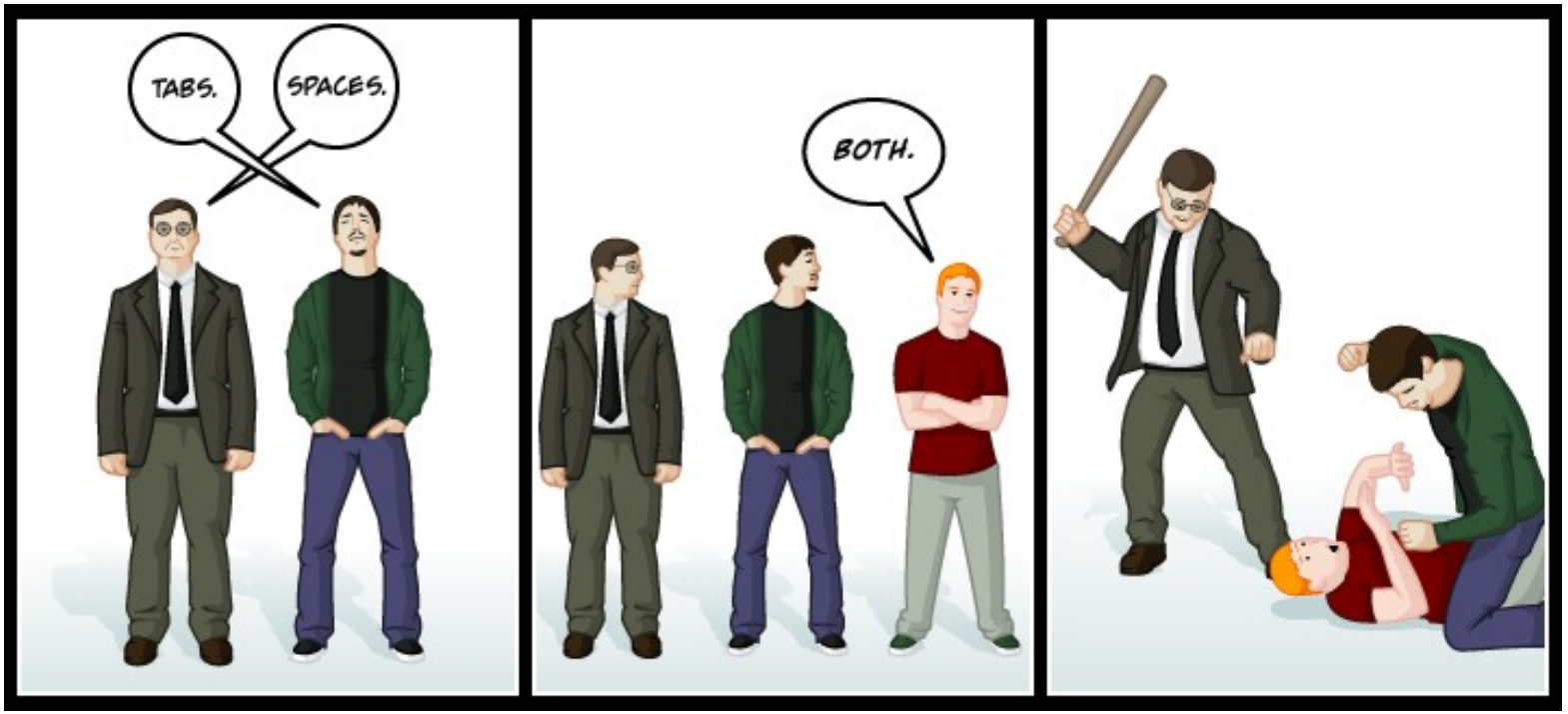
```
if (expression) {
    # code à exécuter si l'expression est vraie
} else {
    # code à exécuter si l'expression est fausse
}
```

Indentation

- **On peut créer une indentation en appuyant sur la touche « Tab » du clavier**
- **Dépendamment de la configuration de l'éditeur de texte utilisé, cela aura pour effet d'insérer soit...**
  - Le caractère invisible « tabulation »
  - Ou un certain nombre de caractères « espace »
- **On peut aussi sélectionner un bloc de code et appuyer sur « Tab » pour indenter plusieurs lignes à la fois**

Tabulations vs Espaces

- **Quand il est question de style de code, la règle d’or est toujours la cohérence!**



Espacement

- **Il peut y avoir, ou non, des espaces entre les mots et les symboles d'une ligne de code:**
  - if (expression) {
  - if(expression){
  - if(expression) {
- **C'est souvent une question de préférence personnelle**
- **Encore une fois, l'essentiel est d'être cohérent!**

Lignes vides

- **Il faut faire un usage judicieux des lignes vides dans le code**
- **Placer une ligne vide entre chaque ligne de code est généralement considéré une mauvaise pratique**
  - Placer plutôt une ligne vide entre chaque « bloc logique » contenant des instructions pour exécuter une même « tâche »

```
$nombre1 = 42

$nombre2 = 23

$somme = 42 + 23

echo "$nombre1 + $nombre2 = $somme"
```

```
$nombre1 = 42
$nombre2 = 23
$somme = 42 + 23

echo "$nombre1 + $nombre2 = $somme"
```

Position des accolades

Position des accolades

- **Il y a deux façons répandues de placer les accolades**
- **Choisissez celle que vous préférez, mais faites preuve de cohérence!**

```
if (expression) {
    # bloc de code
} else {
    # bloc de code
}
```

```
if (expression)
{
    # bloc de code
}
else
{
    # bloc de code
}
```

Noms de variables

Noms de variables

- **Trois principales méthodes existent en programmation pour nommer des variables:**
  - camelCase
  - UpperCamelCase
  - snake_case
- **La méthode suivante est généralement réservée aux constantes:**
  - SCREAMING_SNAKE_CASE

Noms de variables

- **Même si PowerShell n’est pas sensible à la casse, les majuscules demeurent utiles pour la lisibilité des noms de variables**
  - Lorsqu’on utilise camelCase ou UpperCamelCase

Noms de variables

Encore une fois, le mot à retenir est:

COHÉRENCE

Guide de style de PowerShell

Guide de style de PowerShell

- **Un guide de style officiel existe pour PowerShell**
- **Ce guide décrit les standards de codage qu'il est recommandé de suivre lorsqu'on programme en PowerShell**
- **Dans le cadre du cours, vous n'êtes pas tenus de suivre ce guide**

https://github.com/PoshCode/PowerShellPracticeAndStyle

Fin de la présentation

Des questions?

Image par Alexander Drachmann (CC BY-SA 2.0)