



# ICT122 – INTRODUCTION À POWERSHELL

---

# Historique

- L'interpréteur "cmd.exe" et les scripts VBscripts sont devenus désuets, pas prévus pour l'administration
- Support en natif pour tous les nouveaux OS MS depuis Vista, server 2008 etc..
- Uniformisation de l'administration des systèmes
- d'exploitation Windows
- Offrir un outil digne de la concurrence Linux
- Syntaxe facilitant l'apprentissage et l'utilisation
- Hérite du Framework.Net => orienté objets

# Les 3 commandes de base

- `Get-Help`: affiche l'aide, rubrique ou commande
- `Get-Command` : permet de rechercher les commandes disponibles de multiples façons
- `Get-Member` : comme PowerShell est orienté objets, permet d'obtenir les informations sur un objet, variable, etc..

Ce sont les 3 commandes indispensables à maîtriser!

# Alias

- Les alias permettent l'assignation de noms raccourcis sur les cmdlets. Ceci permet de personnaliser la syntaxe des cmdlets.
- Exemple: assignation de l'alias `gh` sur la *cmdlet* `get-help`:
  1. S'assurer qu'aucun alias du même nom n'existe: `get-alias g*`
  2. Appeler l'aide de la commande `set-alias: get-help set-alias`
  3. Appeler la cmdlet `set-alias` avec le nom du nouvel alias:  
`set alias gh get-help`
  4. Utiliser `get-alias` pour vérifier que l'alias est assigné:  
`get-alias gh`

# Le Pipe

- Le tube permet de transmettre le résultat d'une commande à une autre commande.
- Exemple: `gci . | sort`
- Liste les éléments du répertoire courant puis les transmet à la commande `sort` qui va les classer par ordre alphabétique.

# Format

- Quatre cmdlets permettent le formatage de l'affichage dans PowerShell:

<code>Format-list</code>	Affichage par liste
<code>Format-wide</code>	Affichage par colonnes
<code>Format-table</code>	Affichage sous forme de tableau
<code>Format-custom</code>	Affichage personnalisé

# Naviguer dans le système de fichier

PowerShell	Cmdlet	
DIR	Get-ChildItem	Lister le contenu d'un répertoire
CD	Set-Location	Changer de répertoire courant
MD	New-Item	Créer un fichier/répertoire
RD	Remove-Item	Supprimer un fichier/répertoire
MOVE	Move-Item	Déplacer un fichier/répertoire
REN	Rename-Item	Renommer un fichier/répertoire
COPY	Copy-Item	Copier un fichier/répertoire

# Les chaînes de caractère

- Comme dans la plupart des langages scripts, les chaînes de caractères sont délimitées par des guillemets simples ou doubles.
  - `$a = 'Hello'`
  - `$b = 'world'`
- Entre guillemets simples, les variables ne sont pas substituées:
  - `Write-Host '$a $b' => $a $b`
- Entre guillemets doubles, les variables sont substituées :
  - `Write-Host "$a $b" => Hello world`



# Substitution des variables

- Syntaxe d'accès à une propriété d'objet:

- `$(objet.propriété)`

- PowerShell substitue la variable.

Exemple:

- `$a = Get-ChildItem c:\config.sys`

- `Write-Host "Taille fichier = $a.Length octets"`

- `=> Taille du fichier = c:\config.sys.Length octets`

- Il faut donc écrire:

- `Write-Host "Taille fichier = $($a.Length) octets"`

- `=> Taille du fichier = 10 octets`

# Démarrage de la console

- La console démarre avec les droits de simple utilisateur
- Les droits sont donc limités
- Pour ouvrir la console classique ou graphique (ISE) avec les privilèges Administrateur ☐
  - Clic droit – **Exécuter en tant qu'administrateur**

