

# LE LANGAGE POWERSHELL

SISR4 – SIO2

# Langage Powershell

- Un langage de script est un langage composé des commandes. Ces commandes sont interprétées et exécutées directement.
- On peut enchaîner l'exécution des commandes en les inscrivant dans un fichier.
- Cela va permettre d'automatiser des tâches d'administration répétitives.

# Langage Powershell

- ❑ Il est actuellement en version 2.0 nativement avec Windows 2008 R2 et Windows 7 et en version 3.0 pour Windows 8 et Windows Server 2012.
- ❑ On dispose en natif de Windows Powershell (console) et Windows Powershell ISE (Editeur).
- ❑ Nous utiliserons POWERGUI.

# Langage Powershell

- Les commandes de PWS sont appelées des cmdlets (pour command-applets). Elles sont pour la plupart d'entre elles constituées de la manière suivante : un verbe, un tiret, un nom : **verbe-nom**. Par exemple : **get-command**
- Le verbe indique l'action que l'on va appliquer sur le nom. Il y a toute une série de verbes génériques : Get, Add, Remove, Set, etc... Les noms constituant les commandes sont toujours au singulier. C'est aussi vrai pour les options des commandes (on parle de paramètres en PWS et pas d'options).

# Langage Powershell

- ❑ Pour obtenir la liste des commandes : `Get-command`
- ❑ Pour obtenir de l'aide sur une commande : `Get-help`  
`NomDeLaCommande`
- ❑ `Get-help get-childitem`
- ❑ `Get-Help get-childitem -Examples`

# Langage Powershell

- Pour pouvoir exécuter un script, il faut vérifier la stratégie d'exécution :
  - ▣ **Restricted** C'est la stratégie la plus restrictive. L'exécution de scripts est interdite.
  - ▣ **AllSigned** C'est la stratégie la plus sûre. Elle permet de n'exécuter que des scripts signés.
  - ▣ **RemoteSigned** Cette stratégie permet de n'exécuter que les scripts créés localement.
  - ▣ **UnRestricted** C'est la stratégie d'exécution plus ouverte. Tous les scripts peuvent être exécutés quelle que soit leur provenance.

# Langage Powershell

- Pour connaître la stratégie d'exécution en cours
  - ▣ `Get-ExecutionPolicy`
- Pour modifier la stratégie d'exécution en cours
  - ▣ `Set-ExecutionPolicy AllSigned`

# Langage Powershell

- Avec PowerShell, il est possible d'enchaîner les commandes entre elles, de telle sorte que la sortie de l'une devienne l'entrée de l'autre. C'est ce qu'on appelle le pipeline.
- Dans PowerShell, le canal de communication ainsi créé permet de transporter des données sous forme d'objets. Le pipeline est matérialisé par le caractère « | » (AltGr 6). Il transfère le résultat d'une commande comme entrée de la commande qui lui succède.
- Get-command | Set-content c:\temp\fichier.txt



# Langage Powershell

- Le pipeline transporte les données sous forme d'objets. On peut donc ensuite appliquer un filtre sur les objets pour ne conserver que ceux qui nous intéressent.
- L'objet courant est représenté par les caractères « \$\_ ».
- `get-childitem c:\Windows | where {$_.name -like "system*"}`

# Langage Powershell

- ❑ Le pipeline transporte les données sous forme d'objets. On peut donc ensuite appliquer un traitement sur tous les objets récupérés.
- ❑ On utilise pour cela la boucle `forEach` qui permet de parcourir des listes d'objets envoyés par la commande précédente.
- ❑ Exemple : tous les fichiers du répertoire « `c:\temp` » dont le nom commence par « `fichier` » doivent être détruits.

# Langage Powershell

- Lançons la commande Get-Childitem sur le répertoire c:\temp
  - ▣ Get-childitem c:\temp
- Sélectionnons les bons fichiers
  - ▣ Get-childitem c:\temp | where {\$\_.name -like « fichier\*»}
- Détruisons les fichiers concernés
  - ▣ Get-Childitem c:\temp | where {\$\_.name -like « fichier\*»} | foreach {\$\_.delete()}

# Langage Powershell

- Comment savoir que le nom s'appelle « name » et que la méthode pour détruire un fichier s'appelle `delete()` ?
- Pour cela il faut connaître les propriétés et méthodes du type d'objet manipulé.
- On utilise la commande `Get-member`
  - ▣ `Get-childitem c:\temp\fichier.txt | get-member`

# Langage Powershell

- ❑ Pourquoi utiliser PowerGUI ?
- ❑ Parce qu'il offre une aide syntaxique à la Visual Studio que vous avez utilisé l'année dernière.