**Azure PowerShell Essentials**

Welcome to this course on Azure PowerShell where we will become confident and competent with the following:

* The syntax, structure, and usage of PowerShell and Azure PowerShell.
* Managing Azure resources through Azure PowerShell.
* Building scripts in PowerShell ISE and Visual Studio Code that use Azure PowerShell.

# Introduction

## What is PowerShell

PowerShell is a scripting language and an interactive command environment, it was developed by Microsfot for task automation and configuration management

Shell CLI = Command-Line Interface

PowerShell can manage Azure Active Directory, Azure subscription, Azure SQL DataBase and Azure Resource Group.

There are four mains tools for working with Microsoft Azure

Azure Power Shell

Azure CLI

Azure portal

Software development kit ( SDK)

## Installing PowerShell

PowerShell is installed by default and can be found inside the search field from the taskbar by typing "PowerShell".

You can also install PowerShell Core from <https://github.com/PowerShell/PowerShell/releases/tag/v6.2.2>.

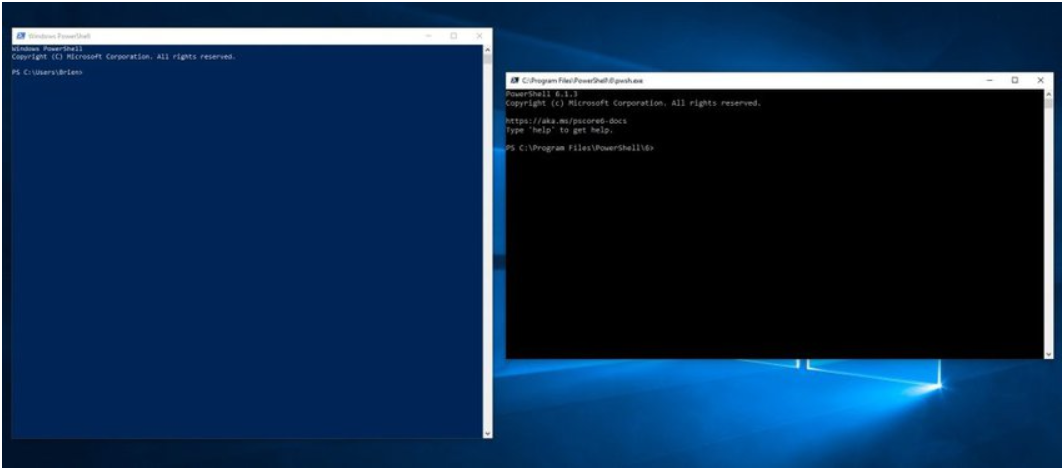
Installing PowerShell Core is a simple process. As you can see in the figure below, PowerShell core uses a simple installation wizard that is similar to that of most other Windows applications.

**What is the difference between PowerShell and Powershell core?**

The first and foremost **difference** is as we know that **PowerShell Core** is multiplatform and runs on Windows, Linux, and MacOS while the old one runs on Windows only. If you are curious, you can run **PowerShell Core** on unsupported ARM versions as well, but that is buggy and is expected to throw some errors.

The most obvious difference between the two windows is the background color

On left PowerShell and on right PowerShell Core.



PowerShell Versus PowerShell Core

2511 commandes et 79 modules dans PowerShell

2096 commandes et 53 modules dans PowerShell Core

PowerShell est la version 5

PowerShell Core est la version 6

## Benefits of learning PowerShell

PowerShell is a powerful scripting tool that can expedite our admin tasks and increase our daily productivity.

## Lecture: Basic Commands in PowerShell

This lesson introduces the basic commands everyone should be familiar with when using PowerShell and PowerShell Core

**Se déplacer:**

Set-Location ou cd

**Lancer et arrêter un programme:**

Start-Process NameOfProgramm

Stop-Process NameOfProgramm

Ex: Start-Process notepad

**Clear the screen**

cls

**Obtenir de l’aide**

Get-Help NameOfCommand

**Obtenir des informations sur les commandes :**

Get-Command

Ex: Get-Command -Module Microsoft.PowerShell.Management

La cmdlet Get-Command de PowerShell affiche les commandes disponibles dans la session active.

**Get the current directory:**

Get-Item .

Il ne faut pas oublier le point à la fin.

**Get all the items in the current directory.**

Get-Item \*

**Afficher le contenu d’un fichier**

Get-Content -Path .\demo.txt -TotalCount 8

 L'applet de commande Get-Content obtient le contenu de l'élément à l'emplacement spécifié par le chemin d'accès, par exemple le texte d'un fichier. Elle lit une ligne de contenu à la fois et retourne un objet pour chaque ligne.

Le –Path n’est pas obligatoire, le TotalCount indique le nombre de lignes souhaitées, si on ne le met pas il retournera toutes les lignes.

**Lancer et arrêter un service :**

Start-Service -Name "eventlog"

Stop-Service -Name AudioSrv

**Pour verifier le statut d’un service :**

Get-Service -Name AudioSrv

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

**Copier un fichier:**

Copy-Item "C:\Users\DirName\demo.txt" -Destination "C:\Demo"

**Supprimer des fichiers:**

Remove-Item C:\Users\DirName\\*.\*

**L’équivalent de pwd:**

Get-Location

## PowerShell Modules

A module is a package that contains PowerShell commands, such as cmdlets, providers, functions, workflows, variables, and aliases. PowerShell users who write commands can use modules to organize their commands and share them with others. PowerShell users who receive modules can add commands in the modules to their PowerShell sessions and use them just like the built-in commands. Module components include code files, assemblies, manifest files, and directories.

The four types of PowerShell modules include script modules, binary modules, manifest modules, and dynamic modules.

How to find modules available for installation:

Get-Module -ListAvailable

How to find installed modules:

Get-Module

How to find all commands contained within a module:

Get-Command -Module (module name)

How to get help on commands contained within a module:

Get-Help (command name)

Update and install help files for commands:

Update-Help

# PowerShell Concepts

## Command and Syntax Basics

The syntax is verb-noun and parameters, ex: Get-Command

Common parameters include -help, -confirm, and -verbose.

Connaître toutes les commandes et fonctions disponibles: **Get-Command**

Connaître tous les services disponibles: **Get-Service**

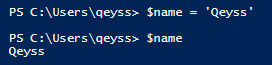
## The Three most important Commands in PowerShell

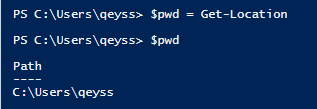
The three commands covered in this lesson are the building blocks for learning to use PowerShell as a Power User. These commands include:

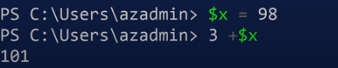
* Get-Help
* Get-Command
* Get-Service

## Using Variables

Les variables commencent par un $ comme en PHP







## Understanding PowerShell Pipelines

Pipelines act like a series of connected segments of pipe. Items moving along the pipeline pass through each segment. To create a pipeline in PowerShell, we connect commands together with the pipe operator: |. The output of each command is used as input to the next command.

This command pipeline starts the WMI service on the computer:

get-service wmi | start-service

Mastering pipelines requires practice, but combining simple commands into pipelines saves time and typing as well as makes our scripting more efficient.

# PowerShell in Azure

Installing the Azure PowerShell Module

To install the Azure PowerShell module for the current user:

Install-Module -Name Az -AllowClobber -Scope CurrentUser

To install the Azure PowerShell module for all users:

Install-Module -Name Az -AllowClobber -Scope AllUsers

## Signing into Azure

There are three authentication methods:

1. Logging in from the Cloud Shell (automatic).
2. Logging in from a local PowerShell session using Connect-AzAccount and web authentication.
3. Logging in with a service principal.

## The Most Important Azure PowerShell Commands

Avoir des informations sur l’inscription du compte:

Get-AzSubscription

Get-AzSubscription | fl 🡪Pour afficher cela en format liste

List all Azure Virtual Machine

Get-AzVm

Start or Stop Azure Virtual Machine

Start-AzVm, Stop-AzVm, and Remove-AzVm

Create a New Azure Virtual Machine

New-AzVm

Savoir toutes les ressources disponibles :

Get-AzResourcce

Create a new resourceGroup

New-AzResourceGroup

List All Storage Account

Get-AzStorageAccount

Create a new StorageAccount

New-AzStorageAccount

List All Virtual network

Get-AZVirtualNetwork

Create a Virtual Network

New-AZVirtualNetwork

List All Az Load Balancer

Get-AZLoadBalancer

To log or disconnect

Login-AzAccount and Logout-AzAccount

Create a new az public Address

New-AzPublicIPAddress

## Working with Outputs

Each Azure PowerShell cmdlet formats output to be easy to read. PowerShell allows you to convert or format cmdlet output by piping to one of the following cmdlets:

**Format-Table**

Default output. Doesn’t display all the information

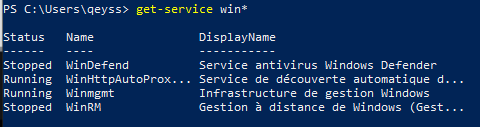
**Format-List**

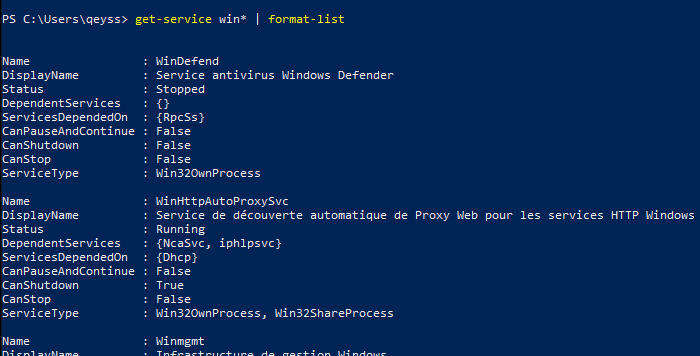
Displays two columns: one for property names and one for property values.

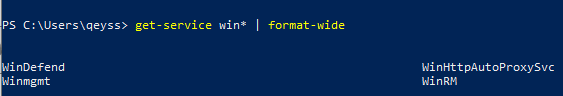
**Format-Wide**

Displays one property name per query. Displayed property controlled by argument.

Quelques exemples:







## Using Nested Properties

L'applet de commande Get-Member permet d’obtenir les propriétés et méthodes des objets, c’est ce qu’on appelle les membres.

Ex : get-service | get-member

Cette commande affiche les propriétés et méthodes des objets processus (System.ServiceProcess.ServiceController) générés par l'applet de commande Get-Service.

Elle utilise l'opérateur de pipeline (|) pour envoyer la sortie d'une commande Get-Service à Get-Member.

Get-Member is an essential command for discovering more about a PowerShell objects.  Because you are at the command line, you cannot right-click an object and check its properties; instead, what you can do is type: Get-ObjectXYZ | **Get-Member.**

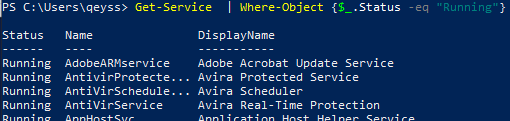
## Filtering Results

This lesson focuses on filtering and selecting PowerShell objects using the Where-Object and Select-Object  and the $\_ operator and comparison operators.

In the  $\_ operator, the  $\_  represent the current pipeline object and comparison operators.

Si je veux lister tous les services en cours d’exécution :

Get-Service | Where-Object {$\_.Status -eq "Running"}

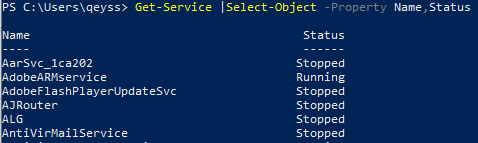


Get-Service | Where-Object {$\_.Status -eq "Stopped"}

Get-Service | Where-Object {$\_.Status -eq "Stopped"} | Select-Object -First 5

Get-Service | Where-Object {$\_.Status -eq "Stopped"} | Select-Object -Last 5

The Select-Object cmdlet selects specified properties of an object or set of objects. It can also select unique objects, a specified number of objects, or objects in a specified position in an array.



# Managing Azure with PowerShell

## Creating Virtual Machines Using PowerShell in Azure

## Creating Storage Account in Azure

Creating Azure SQL Databases

Creating Standard Load Balancers in Azure

l

f