

# CR431 Introduction à PowerShell:

## Devoir 1

### Date de remise:

- 26 novembre 2025, à **19:00**
- **Aucun** retard ne sera toléré
  - **10%** Dès la première minute, 10% par tranche de 24h supplémentaires
- Assurez-vous d'appuyer sur le bouton **TURN-IN**

### Méthode de remise:

- Communauté Teams CR431 via l'onglet « devoir » ou « Assignment ».
- Assurez-vous que le travail remis par votre co-équipier CORRESPOND à vos attentes AVANT la date limite.

### Document à soumettre:

- Un rapport PDF contenant les réponses sous la forme de capture d'écran **ET** une description de la réponse. **Une page titre de Polytechnique doit être utilisée.** Le PDF doit être soumis dans Teams directement. Je m'attends à une description du cheminement. Les travaux ne contenant que des captures d'écran recevront la note zéro.
- **Votre fichier « .psm1 » de module PowerShell et le « psd1 » si vous avez créé un manifest**
- Une démonstration vidéo (courte) est un atout (Optionnel).
- Vous pouvez aussi utiliser une archive .zip pour les fichiers sources en excluant le rapport PDF
- Votre fichier d'archive ZIP doit comporter le prénom de tous les participants

### Description:

Un Module PowerShell est, en terme simple, un regroupement de fonctions associées à un sujet particulier (ex : gestion active directory). Les commandes PowerShell sont en général ajoutées via des modules.

Le consommateur ou utilisateur d'un de ces modules peut ajouter les commandes contenues dans le module dans sa session PowerShell pour les utiliser comme des commandes natives ou à l'intérieur de ses propres scripts. Pour associer toutes les fonctions d'un module autour du même concept, on doit suivre quelques règles comme les noms du module qui indique son utilisation (i.e. : Les fonctions du module Active Directory doivent interagir avec ce dernier). Les noms des fonctions du module doivent aussi respecter le standard PowerShell en **respectant les verbes et indiquer le sujet des fonctions** (i.e. les fonctions Active Directory ont un nom qui débutent toutes par « AD »). Ceci facilite la recherche de commande et permet de mieux organiser les commandes dans PowerShell.

Le but de ce devoir est de **construire** votre propre module à propos d'un cas d'utilisation de votre quotidien. Par exemple, votre module pourrait servir à :

- Installer, modifier, gérer une infrastructure (i.e. : machines virtuelles, disques durs d'entreprise, activation de services à distance, capture d'espace disques sur un réseau)
- Gérer des comptes inactifs, identifier comptes avec le même mot de passe, etc.
- Gérer une bibliothèque de Photos basée sur des caractéristiques du Meta data de chaque photo.
- Gérer des vulnérabilités découvertes et les évaluer en fonction du KEV (known exploited vulnerabilities)

Votre module doit démontrer les connaissances acquises pendant le cours telles que les boucles, les conditions, les paramètres et la gestion d'exceptions. Ces tâches doivent être accomplies à l'aide de fonctions qui seront incluses dans votre module PowerShell. Vous pouvez inclure d'autres modules pour agréger vos fonctions.

**ATTENTION : l'utilisation de menu à l'intérieur de votre module est à proscrire sauf s'il est appelé via une fonction! L'importation du module ne doit pas générer d'interaction avec l'utilisateur.**

Requis du devoir et rapport:

1. Au minimum 3 (trois) fonctions principales à propos d'un sujet précis et décrire votre cas d'utilisation, ex:
  - a. Création et destruction de VM dans Azure ou Hyper-V avec tous les paramètres requis
  - b. Monitoring de mémoire des serveurs dans votre environnement
2. Décrire l'installation de votre module lorsque je le téléchargerai
3. Décrire les dépendances de votre module (ce qui doit être présent pour que ça fonctionne)
4. Écrire votre texte pour une personne débutante en T.I.
5. Décrire les commandes utilisables (par moi) et expliquer comment je dois utiliser chacune de vos fonctions en prenant soin d'inclure la description des paramètres dans le rapport. Pour chaque fonction :
  - a. Comment l'utiliser
  - b. Liste des paramètres
  - c. À quoi sert chaque paramètre
  - d. Format attendu des paramètres
  - e. Décrire le résultat attendu
  - f. Deux Exemples d'utilisation

## Pointage

Les notes sont attribuées en fonction du contenu du rapport et du code dans le module (2 fichiers doivent être soumis au minimum). La difficulté et complexité du code est prise en compte (combinaison des notions apprises depuis le début du cours). Je m'attends à un rapport **professionnel** bien structuré, bien séparé par section pour que le lecteur puisse rapidement trouver l'information.

Documentation et description de votre module et ses fonctions dans le rapport PDF: 25%

- Respect des 5 points mentionnés dans la description ci-haut
- Est-ce qu'une personne ne vous connaissant pas peut réussir à installer le module?

Complexité du code: 25%

La complexité du code est évaluée en fonction de :

- Paramètre des fonctions, retour des fonctions
- Validation et/ou prototypage des variables et objets
- Traitement des exceptions
- Nombre de décisions prises par fonction
- Nombre de boucles par fonction

Commentaires dans le code (chaque fonction, chaque boucle, etc.) : 25%

- Quelle est l'intention de la fonction, test, validation, comparaison
- Logique du test ou de la boucle (ce qui devrait se passer)
- Format ou résultat attendu

Présentation (Page titre, visuel, syntaxe, grammaire) : 25%

- Page titre de Polytechnique conforme avec nom du cours, à qui votre rapport est présenté, vos noms et matricule et date de soumission
- Séparation des sections du projet
- Police de caractère 12 (texte) et grosseur semblable dans les captures d'écran
- Simple interligne
- 15 pages (maximum 20)
- Paragraphe en mode « justifié » (alignement à droite ET à gauche)
- Rognage des captures pour éliminer l'information non essentielles (i.e. : les zones sans informations)
- Identification des captures d'écrans
- Introduction, conclusion