



PowerShell – Pour les développeurs

Vous utilisez PowerShell comme outil de travail principal mais il vous manque quelques connaissances pour être encore plus performant ? Vous souhaitez insuffler de nouvelles façons de travailler au sein de votre équipe ou votre entreprise en utilisant des méthodes DevOps qui ont fait leurs preuves ou vous souhaitez tout simplement mieux partager vos scripts au sein de modules qui prendront place dans une bibliothèque mais vous ne savez pas comment faire ?

La sécurité est pour vous un enjeu préoccupant et vous souhaitez mettre en œuvre les meilleures pratiques et techniques pour déléguer et suivre des actions faites par des collègues ?

Si la réponse à l'une de ces questions est positive, alors cette formation vous saura vous apporter toutes les réponses et plus encore !

Durée : à la carte (3 jours minimum)

Objectif du cours :

- Pousser PowerShell jusqu'à ses limites
- Plonger au cœur du langage
- Interagir avec les API bas niveau de Windows
- Comprendre les concepts et les outils DevOps
- Apprendre à écrire des scripts de qualité professionnelle et à les partager en environnement d'entreprise

Public :

- DevOps
- Scripteurs avancés
- Développeurs

Prérequis :

- Bonnes connaissances de PowerShell
- C# ou autre langage .NET est un plus
- Programmation orientée objet
- Connaissances de l'administration système Windows

Remarque : des travaux pratiques ponctuent chaque partie de cours théorique.



Itinéraire pédagogique

1. Fonctions avancées PowerShell
 - a. Généralités
 - b. L'attribut [CmdletBinding]
 - c. Activer les modes verbose et debug
 - d. Accepter des données via le pipeline (par valeur et par nom de propriété)
 - e. Créer des jeux de paramètres
 - f. Ajout d'une aide intégrée
2. Maîtrise des objets et du concept de classes
 - a. Retourner des objets et non pas du texte
 - b. Création d'objets (et collections d'objets) à partir de zéro
 - c. Ajout de membres à un objet existant
 - d. Programmation avec les classes
 - e. Formatage d'un objet personnalisé
 - f. Incorporation de code C#
3. Gestion avancée des erreurs
 - a. Erreurs critiques et non critiques
 - b. Paramètres communs des cmdlets
 - c. Consignation des erreurs
 - d. Redirection des messages d'erreur
 - e. Interception des erreurs
4. PowerShell Best Practices
 - a. Présentation du PowerShell Script Analyzer
 - b. Recueil des meilleures pratiques
5. Développement de modules
 - a. Pourquoi faire ?
 - b. Ecriture d'un manifeste de module
 - c. Conversion d'un script en module
 - d. Conversion d'une collection de scripts en module
6. Mise en œuvre d'une bibliothèque de modules
 - a. Au sein de l'entreprise (on premises)
 - b. Dans le Cloud
7. APIs REST : consommation et publication
 - a. Explication des concepts
 - b. Consommation d'API REST publiques et d'API REST authentifiées
 - c. Bonnes pratiques de mise en œuvre
 - d. Exposition de données via une API REST (explications avec plusieurs techniques)
8. .NET Framework
 - a. Compréhension des interactions entre PowerShell et le monde .NET
 - b. Accès direct aux classes .NET
 - c. Création d'interfaces graphiques avec WPF et les Windows Forms



9. Appel de DLL externes
 - a. Comprendre les interactions avec les DLL C# externes à PowerShell et à en tirer partie
 - b. Chargement des fichiers DLL
 - c. Fonctionnement interne d'une DLL
10. API P/Invoke
 - a. Accès aux API bas niveaux de Windows
 - b. Mise en œuvre
11. PESTER : améliorez la qualité de vos scripts avec les tests
 - a. Découverte du principe du « testing »
 - b. Tests unitaires
 - c. Tests d'intégration
 - d. Mise en œuvre d'un jeu de tests automatisés
12. Sécurité
 - a. Modèle de sécurité de PowerShell
 - b. Journalisation des scripts
 - c. Mise en œuvre du module Just Enough Administration (JEA) pour la délégation des tâches d'administration
13. Bases de données
 - a. Connexion aux bases de données (SQLServer, MySQL, MariaDb)
 - b. Requêtage de tables
 - c. Insertion et mise à jour de données dans une table
 - d. Administration SQL avec le module dbatools
14. Objets COM
 - a. La technologie COM expliquée
 - b. Identifier les objets COM disponibles sur un ordinateur
 - c. Automatisation des composants Office (Word, Excel, PowerPoint)
15. Dépôts de code (repository)
 - a. Présentation du concept
 - b. Gestion des versions
 - c. Introduction à GIT et à GitHub
 - d. Mise en place d'un dépôt local, d'un dépôt distant et synchronisation des deux