

Formation GitHub Copilot : l'IA au service du code

Formation : GitHub Copilot pour développeurs

Module 1 – Introduction à GitHub Copilot

- Qu'est-ce que GitHub Copilot ?
- Comment l'IA aide à écrire du code
- Exemples d'utilisation courante
- Démo simple : écrire une petite fonction
- Exercice : laisser Copilot proposer une solution

Module 2 – Installation & configuration

- Installer Visual Studio Code
- Installer GitHub Copilot et Copilot Chat
- Connexion au compte GitHub
- Activer les suggestions
- Découvrir l'interface et les raccourcis utiles
- Exercice : tester l'acceptation et le refus de suggestions

Module 3 – Générer du code avec Copilot

- Complétion automatique simple
- Générer une fonction à partir d'un commentaire
- Améliorer un petit bout de code
- Créer un composant ou une petite fonction dans le langage du groupe
- Traduire un code d'un langage à un autre
- Exercice : générer un mini-script à partir d'une demande simple
- Exercice : améliorer une fonction trop longue

Module 4 – Utiliser Copilot Chat

- Découvrir la fenêtre de chat
- Faire expliquer un code
- Demander de corriger un petit bug
- Faire générer des commentaires ou un mini-README
- Exercice : faire expliquer un petit fichier
- Exercice : demander une amélioration claire

Module 5 – Créer de bons prompts

- Comment poser une question efficace
- Donner du contexte simple
- Demander un style ou une structure

- Exemples de bons prompts
- Exercice : générer une fonction avec un prompt bien formulé
- Exercice : demander une version plus simple d'un code

Module 6 – Bonnes pratiques avec Copilot

- Relire toujours le code généré
- Éviter de tout laisser faire par l'IA
- Ne pas mettre d'informations sensibles dans les prompts
- Comprendre les limites : parfois Copilot se trompe
- Exercice : analyser une proposition de Copilot et la corriger si nécessaire

Les codes informatiques et les infolettres : combiner les deux pour optimiser son marketing?

Dans le cadre d'une infolettre, vous devez savoir accrocher votre lecteur, les internautes qui vous liront devront avoir envie de vous lire. C'est ce qui permet de vous démarquer des concurrents qui distribuent aussi des courriels d'infolettres. Mais pourtant, avoir un beau contenu n'est pas tout! Vous devez aussi disposer judicieusement des images, des styles de votre texte (polices de caractère), puis avoir un bon logiciel d'infolettres pour avoir des styles les plus présentables à vos interlocuteurs.

Il peut arriver un temps où vous aimerez aller plus loin dans la personnalisation de vos courriels. Mais comment faire? Vous aurez certainement des manipulations à faire dans les styles prédéfinis. Avant de pouvoir créer votre propre style, il vaut mieux commencer par calquer un modèle libre de droit pour le modifier et le faire correspondre avec votre propre image de marque. Ceci demande du travail surtout si vous débutez dans la création de modèles d'infolettres.

Quand un créateur d'infolettres désire avoir le plein contrôle sur ce qu'il publie, il lui est recommandé d'apprendre au moins des [bases des codes informatiques](#). Si une terminologie réunit bien plusieurs codes informatiques, il s'agit de la terminologie dite HTML. [Le HTML est un peu la base](#) à la base des codes informatiques plus poussés tels que [les codes PHP](#) (partie intégrante de WordPress) et Javascript. C'est aussi une bonne base pour personnaliser des infolettres si vous avez déjà quelques notions puis que vous aimeriez les mettre en pratique. Avant d'avoir ces notions, un peu

complexes, vous devez évidemment les apprendre, via une quelconque formation ou de façon autodidacte, en tout ou en partie.

Le marketing par infolettres et le HTML : quel lien peut être fait?

Pour vous, relier HTML et infolettre revient à utiliser le HTML non pour créer un site web ou monter des lignes de code hyper complexes dans le but de gérer un réseau de technologies de l'information; plutôt, vous allez maîtriser un compte de messagerie dit « open source » pour manier habilement le style entier d'un courriel et de ses envois aux différents contacts intéressés. Concrètement, vous pourrez créer des courriels plus personnalisés qu'en utilisant seulement un logiciel de messagerie où les modèles sont très peu personnalisables pour les personnes à l'œil plus aiguisé. Il est ensuite important d'avoir une excellente maîtrise du rendu, et de pouvoir le tester AVANT d'envoyer vos courriels. En effet, toute personne qui emploie la terminologie HTML et qui est soucieuse des résultats que verront les autres gens va impérativement vérifier le résultat AVANT la diffusion. Recevoir un courriel avec des caractères spéciaux mal placés et pratiquement illisibles est frustrant, d'autant plus que ce genre de courriels se retrouve souvent à la corbeille. Il y a des façons de créer un courriel et d'en tester le visuel. Ces techniques de travail se trouvent [dans les cours de terminologie informatique de base](#). Apprendre ces notions peut paraître obscur si jamais vous ne connaissez rien à la terminologie. Par contre, c'est en mettant en pratique les notions à votre rythme que vous arriverez à maîtriser la terminologie.

HTML : ce que le HTML demande comme structure

Bien que la terminologie HTML soit bien simple pour un spécialiste des technologies de l'information, il y a bien des chances qu'une simple omission dans la structure d'un document HTML vienne le rendre inutilisable. [À quoi ressemble un document HTML?](#) Il s'agit d'une feuille où vous inscrivez les éléments terminologique requis et qui renvoie un résultat quand vous la rendez disponible à d'autres utilisateurs en ligne.

C'est important, rendre un document de ce type avec une structure précise. Par exemple, si vous placez ce qui devrait être à la première ligne du document à la dixième ligne, votre document, automatiquement, deviendra inutilisable, sauf si vous le corrigez par la suite et le remettez en ligne. Ce qui est tout de même plutôt choquant, surtout pour un projet d'envergure. C'est pourquoi nous conseillons, pour tous projets, de se doter d'une application ou d'un logiciel qui permet de tester le résultat avant de l'envoyer en ligne ou le rendre disponible à un groupe.

Les balises sont ce qui contrôle votre terminologie

Si la terminologie d'un document informatique était aussi malléable que le texte d'un roman ou d'un article de journal, il y aurait trop de risques que n'importe qui puisse publier absolument n'importe quoi et que n'importe qui puisse pirater un document. Le web serait une véritable cacophonie en quelque sorte! C'est pourquoi la structure du document, déjà mentionnée plus haut, est d'une assez haute importance.

Au début et à la fin de chaque ligne du document, vous devez placer des balises. Sans quoi, ce qui est contenu dans la ligne suivante et qui pour vous est quelque chose d'autre, pour le HTML, sera exactement la même ligne terminologique. Ou, pire, en omettant une balise, vous pourriez obtenir une partie de votre document qui sera illisible ou dont la présentation pourra être très étrange.

Chaque balise commence par « < » et se termine par « > ». C'est assez logique, puisque nous ouvrons le crochet pour mentionner quelque chose, et nous le quand notre propos est terminé. De la même façon, ce que nous appelons une « ligne codée » dans un document comprend une première balise en « <> » et se termine par une balise semblable, mais avec le caractère suivant « / ». Le caractère « /> » signifie « terminé » en quelque sorte.

Les balises permettent aussi de définir un code qui s'insère dans le HTML. Par exemple, insérer un code Javascript (pour animer votre site web et le rendre présentable sur tous les appareils, autant les ordinateurs que les iPhones), vous insérerez une balise avant votre code Javascript, avec une autre balise suivant ce code. Autrement, sans la balise de fin, votre animation ne sera pas prise en compte. Si ces quelques notions vous sont très peu familières, apprendre d'une de vos connaissances qui travaille en informatique ou qui connaît la terminologie HTML vous permettra de vous initier à ce monde un peu différent de l'écriture normale ou des mathématiques.

Pour en revenir aux infolettres, quand vous créerez votre premier modèle, grâce au HTML, vous irez chercher une touche de personnalisation que vous n'aviez pas auparavant. En effet, parce que la terminologie HTML permet à elle seule de créer de façon presque illimitée, vous pourrez vous amuser avec vos premiers designs d'infolettres, mais il est recommandé de le faire à l'aide d'un « éditeur de code ». Le plus connu étant probablement [Visual Studio Code](#). Rendez vos infolettres attrayantes et collez-les à votre propre marque grâce à la terminologie HTML.

Bien dessiner un modèle d'infolettre pour bien la schématiser par la suite

Avoir un croquis, dessiner un plan de ce que vous aimeriez faire voir aux lecteurs de vos courriels, représente une méthode de travail couramment

utilisée. Pourquoi les créateurs vont-ils, environ 90% du temps, construire un croquis, ou le faire construire, avant d'en écrire le contenu HTML? Pour la même raison, en fait, qu'un écrivain utilisera un plan avant d'écrire son prochain roman. Vous avez déjà visuellement devant vous les éléments à mettre en place pour votre visuel avant de vous aventurer dans la structure de ceux-ci à l'intérieur de votre éditeur de code. Puisque c'est une tâche difficile et assez abstraite à faire, monter un modèle d'infolettre à partir de HTML, mieux vaut avoir en tête déjà le résultat voulu pour votre premier chef-d'œuvre! En tant qu'artiste de l'infolettre, il est avantageux d'avoir sur papier un plan de votre infolettre, comme une entreprise de construction ne peut improviser la construction d'une maison. Construire une maison est toujours basé sur les plans d'un architecte, comme construire une infolettre est (presque) toujours basé sur le talent et l'habileté d'un graphiste.

De façon concrète, essayez de respecter le plus possible le plan qui vous est fourni. Autrement, vous risquez de dériver vers un contenu qui ne ressemble pas du tout à ce qui est recherché. De cette manière de travailler repose un risque : vous n'aurez pas produit ce que vous deviez produire, mais plutôt ce que vous aimeriez faire. Il y a une grande différence entre faire ce que l'on VEUT faire, et se limiter à ce que l'on PEUT faire, cela s'applique même dans les lettrines d'une infolettre. Ce que l'on veut produire comme résultat n'est pas nécessairement ce que le graphiste nous permet de faire, puisque chacun est le professionnel de son métier, comme le graphiste est le professionnel censé reconduire votre courriel vers une place de choix dans la boîte aux lettres virtuelle Outlook de vos clients!

La police de caractère fait vraiment une différence?

Il existe plus d'une centaine de polices de caractères toutes différentes que vous pouvez placer dans une infolettre. Autant vous aimerez avoir la bonne police de caractère, autant elle sera difficile à trouver, cette police de caractère parfaite. Pourtant, est-ce que cela fera une vraie différence sur le rendu final, avoir une police de caractère à tout casser? Pour mettre tout cela au clair, il est important de se poser la question suivante : « Si j'avais mis une autre police, est-ce que cela aurait changé la perception de mes clients? » Si la réponse est non, c'est que votre police est bonne. Rien ne sert de perdre plusieurs jours à trouver une police de caractère parfaite, mais encore faut-il considérer que plusieurs autres choses peuvent être mises en place pour créer le courriel le meilleur possible.

Disposer vos paragraphes comme dans un tableau, permet de créer des textes à deux ou trois colonnes. Les images que vous allez présenter, rendez-les attrayantes et à propos! Vous gagnerez ainsi beaucoup plus de clients potentiels que si vos images ne sont pas tout-à-fait bien calibrées. Si elles sont trop sombres ou trop éclairées, les images ne permettent pas de faire valoir tout ce qu'elles peuvent délivrer comme message. Pour le web, il est vrai de dire qu'une image vaut mille mots.

Les lecteurs de votre infolettre liront en premier ce qu'ils voient, et non le texte en soi. Les gens peuvent lire un texte environ 15 secondes et s'y désintéresser si la présentation de celui-ci laisse à désirer, aussi attractif ce texte soit-il. Contrairement au roman, où le texte est l'essence même du succès, le texte sur le web repose sur l'image et le fait qu'on puisse saisir le contenu en lisant les deux ou trois premières lignes du texte et en regardant vite fait la mise en forme d'un courriel (par exemple). C'est en personnalisant votre courriel que vous arrivez à joindre votre public cible. Un modèle de courriel devrait se modeler au public qu'il veut aller chercher, aux futurs clients à qui il souhaite donner envie d'en savoir davantage. Malheureusement, aujourd'hui, plusieurs ignorent encore l'impact d'un bon et beau courriel!

CRÉER ET GÉRER UNE BOUTIQUE EN LIGNE AVEC SHOPIFY

Module 1 – Introduction au commerce électronique et à Shopify

- Comprendre le fonctionnement du commerce électronique (B2C, B2B, Dropshipping)
- Présentation de **Shopify** et de son écosystème
- Création d'un compte et configuration initiale
- Tour du tableau de bord (admin Shopify)
- Comparaison rapide avec WordPress + WooCommerce

Module 2 – Configuration générale de la boutique

- Paramètres généraux (langue, devise, fuseau horaire)
- Configuration des paiements (Shopify Payments, PayPal, Stripe)
- Paramétrage des taxes (Canada / Québec)
- Configuration des frais et zones de livraison
- Rédaction des politiques (retours, confidentialité, conditions d'utilisation)

Module 3 – Gestion des produits et des collections

- Création et modification de produits
- Gestion des variantes (taille, couleur, modèle)
- Organisation des collections (manuelles et automatiques)
- Gestion des stocks et inventaires

Module 4 – Design et personnalisation du site

- Choix et installation d'un thème
- Personnalisation des couleurs, polices et sections
- Création de la page d'accueil
- Gestion des menus et navigation
- Optimisation mobile (responsive design)

Module 5 – Applications essentielles

- Présentation du Shopify App Store
- Installation d'une application utile (ex. avis clients ou livraison)
- Gestion et suppression d'applications
- Bonnes pratiques pour éviter les applications inutiles

Module 6 – Notions de visibilité et promotions

- Paramètres SEO de base (titre et description)
- Création d'un code promotionnel simple
- Connexion aux réseaux sociaux
- Consultation des statistiques principales

Module 7 – Gestion des commandes et suivi des performances

- Gestion et traitement des commandes
- Remboursements et annulations
- Suivi des paiements
- Rapports et statistiques de ventes
- Analyse des indicateurs clés (KPI)

Formation – Création d'agents IA et automatisations avec n8n

Module 1 – Introduction à l'automatisation et aux Agents IA

- Définition et principes de l'automatisation
- Différence entre automatisation traditionnelle et automatisation intelligente
- Rôle des agents IA dans les processus métiers
- Cas d'usage concrets : service client, marketing, ressources humaines, informatique, gestion des données
- Vue d'ensemble de l'écosystème no-code / low-code
- *Atelier : analyse collaborative des tâches répétitives automatisables*

Module 2 – Présentation de n8n

- Présentation générale de l'outil n8n
- Architecture : nodes, workflows, triggers
- Positionnement de n8n face à Zapier et Make
- Atouts : open-source, auto-hébergement, flexibilité
- Exemples d'automatisations réalisées avec n8n
- *Atelier : démonstration d'un workflow élémentaire*

Module 3 – Prise en main de n8n

- Exploration de l'interface utilisateur
- Création et structuration d'un workflow
- Types de déclencheurs : Webhook, Cron, Manuel
- Gestion des exécutions et analyse des logs
- Introduction à la gestion des erreurs
- *Atelier : construction du workflow « Hello Automation »*
- *Atelier : déclenchement manuel et action automatisée (message ou fichier)*

Module 4 – Introduction aux Agents IA dans n8n

- Différence entre agent IA et appel API traditionnel
- Présentation des modules IA disponibles (OpenAI, LLM, AI Agent, etc.)
- Fondamentaux du prompting
- Gestion du contexte et des instructions
- *Atelier : création d'un agent IA simple*
- *Analyse et reformulation d'une demande utilisateur*
- *Génération d'une réponse adaptée*

Module 5 – Création d'un Agent IA autonome

- Définition du rôle, des objectifs et des contraintes d'un agent IA
- Structuration logique des actions et des décisions
- Mise en place d'une mémoire contextuelle ou externe
- Connexion à des outils externes : API, bases de données, services tiers
- *Atelier : conception d'un agent IA autonome*
- *Réception et interprétation d'une requête utilisateur*
- *Sélection automatique de l'action appropriée*
- *Exécution de l'automatisation correspondante*

Formation 3CX complète pour

entreprises | Voip

Module 1 : Fondamentaux des réseaux et télécommunications

- Comprendre l'évolution du RTC et des systèmes téléphoniques d'entreprise.
 - Identifier les composants essentiels : PABX, postes analogiques, terminaux IP.
 - Revoir les bases des réseaux TCP/IP : adressage IP, DNS, DHCP, routage.
 - Expliquer le rôle du NAT, des VLAN et de la QoS pour la voix sur IP.
 - Visualiser l'architecture type d'un réseau d'entreprise moderne.
-

Module 2 : Principes et enjeux de la téléphonie sur IP (ToIP)

- Découvrir les concepts clés et le vocabulaire de la téléphonie sur IP.
 - Comprendre la convergence voix/données et les bénéfices pour l'entreprise.
 - Intégrer une solution ToIP dans l'infrastructure informatique existante.
 - Interconnecter la ToIP avec les réseaux classiques (RTC) via trunks SIP et passerelles.
 - Explorer les fonctionnalités avancées offertes aux utilisateurs et les principaux acteurs du marché.
-

Module 3 : Analyse des messages SIP (initiation)

- Présentation du protocole SIP et de ses usages
 - Structure d'un message SIP
 - Principales méthodes : INVITE, ACK, BYE, CANCEL, REGISTER...
 - Codes de réponse (1xx, 2xx, 3xx, 4xx, 5xx)
 - Call flow : déroulement d'un appel SIP
 - Analyse avec Wireshark ou capture 3CX
 - Identifier les problèmes courants : no audio, one-way audio, erreurs SIP
-

Module 4 : Introduction à 3CX et architecture

- Présentation générale de 3CX
- Architecture : serveur, clients, trunks, flux RTP
- Modes de déploiement : cloud, serveur local, mini-serveur
- Gestion des licences et versions
- Ports utilisés, contraintes réseau et firewall checker
- Interface d'administration : vue d'ensemble

Module 5 : Installation et configuration initiale

- Prérequis techniques (Windows, Linux, Cloud)
- Installation pas à pas
- Assistant de configuration (Setup Wizard)
- Configuration du FQDN et du certificat SSL
- Paramètres réseau et NAT
- Sauvegardes et mises à jour

Module 6 : Gestion des extensions, utilisateurs et téléphones

- Création des extensions et gestion des utilisateurs
- Auto-provisioning et Plug & Play
- Configuration des téléphones IP (Yealink, Fanvil, SNOM...)
- Groupes, droits et profils utilisateur
- Messagerie vocale et renvois d'appel

Module 7 : Routage d'appels, SIP Trunks et scénarios opérationnels

- Configuration des trunks SIP
- Règles entrantes et règles sortantes
- Numéros directs (SDA)
- Files d'attente et stratégies de distribution
- Groupes d'appels, renvois et transferts
- IVR : conception et mise en place d'un serveur vocal interactif
- Horaires d'ouverture et modes jour/nuit
- Gestion des enregistrements et conformité

Module 8 : Fonctionnalités avancées et intégrations

- Rapports et statistiques d'appels
- Intégration CRM (Salesforce, HubSpot, Zoho, Pipedrive...)
- Webclient, application desktop et mobile
- Chat interne et Live Chat Website
- Webmeeting et communication unifiée

Module 9 : Sécurité, dépannage et analyse SIP dans 3CX

- Paramètres de sécurité : anti-hacking, blacklist, restrictions IP
 - Analyse des logs et flux SIP dans 3CX
 - Diagnostic des problèmes courants
 - Problèmes d'enregistrement, d'audio, de NAT
 - Méthodologie de support et dépannage VoIP
-

Module 10 : Atelier pratique – Mise en œuvre complète

- Construction d'un mini-système 3CX opérationnel
 - Création d'extensions et tests d'appels
 - Configuration d'un trunk SIP (simulateur ou opérateur)
 - Création d'un IVR simplifié
 - Scénarios d'appels : interne, externe, renvoi, file d'attente
 - Analyse en direct d'un message SIP capturé
-

Formation : Développement d'interfaces WPF avec C# et Visual Studio

Module 1 – Introduction et bases de WPF

- Présentation de WPF et comparaison avec Windows Forms
- Architecture de WPF : XAML, Code-Behind, Data Binding
- Création d'un premier projet WPF dans Visual Studio
- Compréhension du langage XAML et de sa relation avec C#
- Utilisation des contrôles de base (Button, TextBox, Label, ListBox, etc.)
- Organisation de l'interface avec les conteneurs (Grid, StackPanel, DockPanel, WrapPanel)

Module 2 – Mise en forme et styles

- Gestion des ressources (Resources, ResourceDictionaries)
- Introduction aux styles (Styles, Templates)
- Application des thèmes et personnalisation des contrôles
- Utilisation des Layouts pour des interfaces flexibles
- Introduction à l'Event Handling dans WPF
- Exercices pratiques : création d'une mini-interface cohérente

Module 3 – Data Binding et MVVM

- Notions de DataContext et Binding en profondeur
- Types de bindings (One-Way, Two-Way, etc.)
- Gestion des listes et collections (ItemsControl, ListView, DataGrid)
- Validation et conversion de données (ValueConverters)
- Introduction au pattern MVVM (Model-View-ViewModel)
- Cas pratique : mise en place d'une petite architecture MVVM

Module 4 – Projet pratique et fonctionnalités avancées

- Navigation entre vues (UserControl, Pages, Navigation)
- Gestion des commandes (Commands, RelayCommand)
- Interaction avec une base de données ou un service (exemple simple avec Entity Framework ou API simulée)
- Gestion des ressources multimédia (images, icônes)
- Bonnes pratiques pour structurer un projet WPF
- Atelier final : réalisation d'une interface complète en lien avec votre projet en C#

Formation Python: la plus complète

Introduction à Python

- Installer Python et un éditeur (VS Code, Jupyter...)
- Découvrir la console interactive et les scripts
- Créer son premier script Python

Syntaxe et fondamentaux

- Types de données (int, float, str, bool)
- Variables et typage dynamique
- Opérations arithmétiques et logiques
- Conditions (if, elif, else)
- Boucles (for, while) et itérations
- Fonctions (définition, paramètres, retour)
- Introduction au typage statique avec type hints

Manipulation de chaînes de caractères et de fichiers

- Opérations sur les chaînes de caractères
- Lecture et écriture de fichiers avec with open()
- Manipulation de fichiers CSV
- Introduction au format JSON

Structures de données

- Listes, tuples, ensembles, dictionnaires
- Parcours et filtrage avec list comprehension
- Fonctions lambda, map, filter, zip
- Tri et recherche dans les structures

Gestion des erreurs et exceptions

- Comprendre les types d'erreurs (SyntaxError, ValueError...)
- Utiliser try / except / finally
- Lever des exceptions personnalisées
- Bonnes pratiques de gestion d'erreurs

Programmation modulaire

- Créer et utiliser des modules
- Importer avec import et from
- Découverte des modules standards utiles (os, datetime, pathlib, etc.)

Programmation orientée objet (POO)

- Créer une classe, instancier des objets
- Encapsulation, héritage, polymorphisme
- Constructeurs et méthodes spéciales (__init__, __str__...)
- Introduction aux dataclasses

Interaction avec des bases de données

- Connexion à SQLite avec sqlite3
- Lire, insérer, mettre à jour et supprimer des données
- Introduction à SQLAlchemy (ORM moderne)

Travailler avec des API Web

- Requêtes HTTP avec requests
- Consommer une API REST (GET, POST...)
- Analyse des réponses JSON

Développement Web avec Flask

- Principes de base de Flask
- Création d'un serveur REST simple
- Routage et gestion des requêtes
- Tester une API avec Postman ou curl

Tests automatisés avec Pytest

- Écrire des tests unitaires avec pytest
- Asserts, organisation des fichiers de test
- Exécuter les tests et interpréter les résultats

Initiation à la manipulation de données

- Introduction à NumPy et Pandas
- Lire des fichiers CSV avec Pandas
- Filtrer, grouper et résumer des données
- Création de DataFrames à partir de données JSON ou SQL

Démarrer avec Python : des fondamentaux à l'orienté objet

Introduction à la formation Python

- Présentation de Python et de son écosystème
- Origine, philosophie et cas d'usage
- Installation de Python et prise en main de l'IDE
- Découverte de la documentation officielle et des ressources en ligne
- Introduction aux modes d'exécution : script et interactif
- Premier programme : « Hello World »

Module 1 : Syntaxe de Python

- Introduction à la formation [Python](#)
- Types et affectation : int, float, str, bool
- Opérateurs arithmétiques et logiques
- Calculs et opérations sur les variables
- Les commentaires en Python
- Les entrées/sorties (input, print, formatage)
- Instructions de contrôle : pass, if, elif, else, while, for
- La fonction range et les itérations contrôlées
- Chaînes de caractères et slicing
- Fonctions prédéfinies et création de fonctions
- Les fichiers : ouvrir un fichier, modes d'ouverture
- Lire et écrire dans un fichier texte
- Gestion de l'encodage
- Atelier pratique : *Création de scripts complets*

Module 2 : Gestion des erreurs / Exception

- Comprendre les types d'erreurs : SyntaxError, TypeError, etc.
- Lever et déclencher des exceptions personnalisées
- Gestion des erreurs avec try/except
- Utilisation des clauses else et finally
- Affichage des messages d'erreurs détaillés

- Atelier pratique : *Simulation et gestion des erreurs*

Module 3 : Maîtriser les structures de données

- Créer et manipuler des listes
- Utiliser les dictionnaires
- Découvrir les tuples et les ensembles
- Utiliser les slices sur les séquences
- Parcourir les structures avec `enumerate()` et `zip()`
- La méthode `items()` sur les dictionnaires
- Tri et filtrage des collections
- Compréhensions de listes et de dictionnaires
- Atelier pratique : *Liste en pile et file*

Module 4 : Modules et Packages

- Présentation des modules standards : `os`, `sys`, `datetime`, `math`
- Création et organisation d'un module
- Importer des modules : `import`, `from-import`
- Découverte des packages et du fichier `__init__.py`
- Structure d'un projet Python modulaire
- Utiliser `pip` pour installer des bibliothèques externes
- Atelier pratique : *Créer et tester un module*

Module 5 : Programmation orientée objet en Python

- Concepts fondamentaux : objet, classe, instance
- Créer une première classe avec attributs et méthodes
- Encapsulation et conventions de nommage (`_privé`, `__très_privé`)
- Les méthodes spéciales : `__init__`, `__str__`, `__repr__`
- Notion d'héritage et polymorphisme (survol)
- Faire la différence entre composition et agrégation.
- Fonctions à arguments variables (`*args`, `**kwargs`)
- Réutilisation du code et bonnes pratiques orientées objet
- Erreurs et exceptions orientées objet
- Atelier pratique : *Création de classes et mise en relation*

Module 6 : Base de données

- Introduction aux bases de données relationnelles
- Installer un driver (`sqlite3`, `MySQL Connector`, `psycopg2`)
- Connexion à une base de données
- Création d'une table avec SQL
- Insérer, supprimer, modifier les données
- Lire et filtrer des données avec des requêtes
- Utiliser des paramètres et prévenir les injections SQL
- Fermer proprement la connexion
- Atelier pratique : *Créer une base et gérer des données*

Rétro-Ingénierie des Logiciels Malveillants

Introduction

- Rappels sur les bonnes pratiques d'investigation numérique
- Présentation des différentes familles de malwares
- Vecteurs d'infection
- Mécanismes de persistance et de propagation

Laboratoire d'Analyse

- Laboratoire virtuel vs. physique
 - Avantages de la virtualisation
 - Solutions de virtualisation
- Ségrégation des réseaux
 - Réseaux virtuels et réseaux partagés
 - Confinement des machines virtuelles
 - Précautions et bonnes pratiques

Supervision de l'Activité d'une Machine

- Réseau
- Système de fichiers
- Registre
- Services

Initiation à l'Analyse Comportementale

- Variété des systèmes

Mise en Place d'un Écosystème d'Analyse Comportementale

- Configuration de l'écosystème
- Définition des configurations types
- Virtualisation des machines invitées
 - VMware
 - VirtualBox
- Installation de CAPEv2 et VirtualBox

Mise en Pratique

- Soumission d'un malware
- Déroulement de l'analyse

- Analyse des résultats et mise en forme

Amélioration via API

- Possibilités de développement et d'améliorations

Analyse Dynamique de Logiciels Malveillants

- Précautions
 - Intervention en machine virtuelle
 - Configuration réseau
- Outils d'analyse
 - OllyDbg
 - Immunity Debugger
- Analyse sous débogueur
 - Step into / Step over
 - Points d'arrêt logiciels et matériels
 - Fonctions systèmes à surveiller
 - Génération pseudo-aléatoire de noms de domaines (C&C)
 - Bonnes pratiques d'analyse
- Mécanismes d'anti-analyse
 - Détection de débogueur
 - Détection d'outils de rétro-ingénierie
 - Exploitation de failles système

Analyse de Documents Malveillants

- Fichiers PDF
 - Introduction au format PDF
 - Spécificités
 - Intégration de JavaScript et possibilités
 - Exemples de PDF malveillants
 - Outils d'analyse : OLE Tools, éditeur hexadécimal
 - Extraction et analyse de la charge
- Fichiers Office (DOC/DOCX)
 - Introduction aux formats DOC/DOCX
 - Spécificités
 - Macros
 - Objets Linking and Embedding (OLE)
 - Outils d'analyse : OLE Tools, éditeur hexadécimal
 - Extraction de code malveillant et analyse
- Fichiers APK
 - Introduction au format APK
 - Outils d'analyse : jadx, Frida, Genymotion, MobSF
 - Contournement des protections d'émulation
 - Compréhension du fonctionnement des applications

Découvrez aussi nos autres formations en cybersécurité et en informatique :

- [Formation Cybersécurité – Sécurité informatique](#)
- [Formation Ethical Hacking et Tests d'intrusion](#)
- [Toutes nos formations en informatique](#)

Pour en savoir plus sur la rétro-ingénierie, consultez la ressource suivante :

[Qu'est-ce que la rétro-ingénierie ? – Wikipédia](#)

Formation Certification CISSP – Certified Information Systems Security Professional

Module 1 : Gestion de la sécurité et des risques

- Compréhension des concepts fondamentaux de sécurité
- Gouvernance de la sécurité de l'information
- Conformité légale et réglementaire
- Gestion des risques de sécurité
- Politiques de sécurité, normes et procédures
- Éthique professionnelle (Code de conduite ISC²)

Module 2 : Sécurité des actifs

- Identification et classification des actifs
- Propriété de l'information et de la responsabilité
- Conservation, protection et gestion des ressources
- Traitement sécurisé des données sensibles

Module 3 : Architecture et ingénierie de la sécurité

- Concepts de sécurité des systèmes
- Modèles de sécurité et principes de conception
- Sécurité des composants matériels et logiciels
- Mesures de protection contre les vulnérabilités matérielles et logicielles
- Chiffrement et cryptographie appliquée

Module 4 : Sécurité des réseaux et des communications

- Conception et sécurisation de l'architecture réseau
- Protocoles sécurisés et vulnérabilités réseau
- Dispositifs de sécurité (pare-feux, IDS/IPS, VPN)
- Contrôle de la sécurité des canaux de communication

Module 5 : Gestion des identités et des accès (IAM)

- Contrôles d'accès physiques et logiques
- Gestion des identités et du cycle de vie des accès
- Mécanismes d'authentification et d'autorisation
- Gestion des privilèges et séparation des tâches

Module 6 : Évaluation et tests de sécurité

- Conception d'un programme de tests de sécurité
- Tests de vulnérabilités et scans réguliers
- Audits de sécurité internes et externes
- Analyse des résultats et recommandations

Module 7 : Opérations de sécurité

- Gestion des événements de sécurité
- Réponse aux incidents et gestion de crise
- Plan de continuité des activités et reprise après sinistre
- Gestion des ressources humaines en contexte opérationnel

Module 8 : Sécurité du développement logiciel

- Intégration de la sécurité dans le SDLC (cycle de vie du développement logiciel)
- Environnements de développement sécurisé
- Contrôles de sécurité applicative
- Tests d'applications et revue de code sécurisée