



Plan de Formation Python



Description de la formation



Quels sont les avantages de suivre une formation Python ?

Le langage Python est particulièrement polyvalent même s'il est particulièrement utilisé dans la Data Science et le Machine Learning grâce à la richesse de ses bibliothèques scientifiques d'analyse de données. Tous les profils de développeurs pourront exploiter ses possibilités que ce soit pour automatiser des tâches systèmes (scripts), approvisionner les systèmes (Ansible), programmer toute sorte d'application (Web, Graphique, ...), créer des Web Services, développer des applications embarquées, programmer des objets connectés, etc.



Quelles sont les opportunités professionnelles après une formation Python ?

Les professionnels formés en Python peuvent accéder à des postes de développeur web, analyste de données, ingénieur en intelligence artificielle, scientifique des données, etc.

Cette formation Python vous permet de disposer des solides bases en programmation objet, présentées et déclinées dans le contexte de Python. Les bibliothèques fondamentales de Python vous seront présentées et nous détaillerons l'accès aux données de différentes sources, selon différents formats, ainsi que les interfaces graphiques afin d'être complet dans la mise en œuvre d'une première application Python réaliste.



Python



Objectifs

Opérationnel

Savoir programmer objet en Python en utilisant les bonnes structures de données et les bibliothèques standards

Pédagogiques

À l'issue de cette formation Python, vous aurez acquis les connaissances et compétences nécessaires pour :

- ✓ Connaître les fondamentaux de la programmation objet en Python
- ✓ Maîtriser la syntaxe du langage Python
- ✓ Connaître et mettre en œuvre les modules Python
- ✓ Manipuler les données en Python
- ✓ Concevoir des interfaces graphiques en Python
- ✓ Disposer d'une première expérience dans l'utilisation des outils de test et d'évaluation d'un programme Python

Python

A qui s'adresse cette formation ?

Public

Ce cours Python s'adresse à tout développeur souhaitant acquérir les bases de la programmation en Python, que ce soit pour concevoir et développer une nouvelle application ou maintenir une application existante.

Elle intéressera également les ingénieurs, statisticiens ou data scientist ayant besoin de connaître les fondamentaux de Python dans le cadre d'un traitement ultérieur de données ou de manipulation de données.

Enfin, compte tenu du caractère informatique généraliste du langage Python, cette formation pourra intéresser toute personne désirant consolider sa culture en développement logiciel (objet, bibliothèques, data, IHM, tests) pour participer aux projets d'entreprises avec plus de recul technique.



Python

Prérequis

Les participants à cette formation Python doivent connaître un langage de programmation, car la formation ne prévoit pas de revenir sur les bases de la programmation (variables, test, boucle, fonction...). L'assimilation de ces concepts en même temps que la syntaxe de Python reste cependant possible mais constitue un risque pédagogique important.

En revanche, il n'est pas nécessaire de connaître la programmation objet car la formation prend le temps de présenter les concepts objets fondamentaux (classe, objet, instance, propriétés, méthodes, héritage) puis dans un second temps leur mise en œuvre en Python.



Python

Contenu du cours

Bien commencer avec Python

- Pourquoi utiliser Python plutôt qu'un autre langage ?
- Exemples de projet en Python
- Différences entre les versions de Python
- Installation de Python (pré-requis, fichiers, arborescence détaillée)
- Fonctionnement de l'interpréteur Python
- Les apports de l'interpréteur interactif iPython
- Un premier programme...

Travaux pratiques

Objectifs : Prendre en main rapidement son environnement de développement

Description : Premier programme simple interactif de saisie et d'affichage de texte afin de prendre en main l'interpréteur Python. Un second programme non interactif permet de voir l'aspect automatisé de Python.

Python

Contenu du cours

Les bases de Python (la syntaxe, les types)

- Organisation du code (modules, package, importations)
- Recommandations de nommage
- Les types de base (entier, chaîne, réel, ...)
- Erreurs classiques sur les types
- Boucles (for), opérateurs, tests (if/else)
- Définition et appel de fonction

Travaux pratiques

Objectifs : Mettre en œuvre la syntaxe fondamentale de Python

Description : Saisie de notes dans un tableau numpy, affichage de la moyenne, de la plus petite et de la plus grande note. Chargement des notes depuis un fichier CSV (avec en tête), recherche du meilleur étudiant.J2

Python

Contenu du cours

Programmer objet en Python

- Limites de la programmation « simplement » structurée
 - Philosophie, concepts, et vocabulaire de la programmation objet
- Représentation des données en programmation objet (classes, propriétés)
- Représentation des traitements (méthodes)
 - Contrôler l'accès aux données (encapsulation)
 - Exemples de classes prédéfinies en Python permettant de profiter pleinement de l'approche objet
 - Cycle de vie d'un objet (constructeur, destructeur)
 - Accéder aux attributs et fonctionnalités de l'objet cible (self)
 - Comment les objets communiquent entre eux ?
 - Classe mère et classe fille (héritage)

Python

Contenu du cours

- Héritage simple et héritage multiple
- La fonction super()
- Protection des traitements (public, protected, private)
- Le mécanisme de traitement des exceptions
- Exemple de formalisme UML (diagramme de classe)
- Tour d'horizon des classes fondamentales existantes en Python
- Notion de Design Patterns

Travaux pratiques

Objectifs : Savoir programmer objet en Python. Comprendre les liens entre les classes.

Description : Mise en œuvre progressive d'un modèle objet en Python CLASSE, ELEVE, EXAMEN, NOTES regroupant l'ensemble des concepts vus dans ce chapitre et favorisant la réflexion sur les bonnes pratiques (encapsulation, héritage ? composition ?) tout en prenant en main la syntaxe de façon concrète.

Python

Contenu du cours

Types de données évolués

Savoir choisir les bonnes structures de données

Tuples, séquences et listes (append, extend, insert, ...)

Fonctions utiles avec les listes (filter, map, reduce)

Gestion des piles ou des files

La puissance des dictionnaires (tableaux associatifs)

Construire une liste sur la base d'une expression
(compréhension)

Travaux pratiques

Objectifs : Savoir utiliser les listes en Python

Description : Ajout à la classe ELEVE de propriétés (utilisation des list et set) sous la forme de label inclinés (utilisation de la librairie Matplotlib)J3

Python

Contenu du cours

Compléments sur le langage Python

Générateurs et itérateurs

Scripts exécutables

Opérateurs associés aux listes

Enchaînement de tests par rapport aux listes

Comparer les séquences

Travaux pratiques

Objectifs : Savoir utiliser les itérateurs

Description : Modification de la classe Résultat Examen afin qu'elle implémente un itérateur. Afficher dans un log les données qu'elle contient via l'utilisateur de l'itérateur (bibliothèque logging)

Python

Contenu du cours

Les modules de Python

- Philosophie de Python avec les modules (standard, tiers, Tour d'horizon des modules standards)
- Module re (expressions rationnelles)
- Modules os et sys (services du système d'exploitation)
- Module csv (fichiers structurés par séparateur)
- Comment trouver les modules ?
- Installer des modules tiers (setuptools, EasyInstall)
- Le dépôt central : Python Package index (Pypi)
- Importer des modules
- Mieux gérer son environnement avec Virtualenv

Travaux pratiques

Objectifs : Maîtriser l'utilisation des modules en Python

Description : Création d'un nouveau module en restructurant le projet précédent afin que les 2 classes Resultat et ResultatExamen soient dans un package note. On écrit ces données dans un fichier texte (librairie csv). On modularise notre projet afin de simplifier sa maintenance et sa livraison en respectant les bonnes pratiques.J4

Python

Contenu du cours

Gestion des données (Fichiers, SGBD, XML, JSON, API)

- Accéder aux fichiers (objet File)
- Conformité des modules Python pour accéder à une base de données (DB-API)
- Python et requêtes SQL
- Filtrer les données grâce aux expressions régulières
- Présentation de la manipulation de flux XML en Python (ElementTree)
- Manipuler des fichiers CSV
- Le format JSON
- Principe des API et des Web Services
- Récupérer des données via des API

Python

Contenu du cours

Travaux pratiques

Objectifs : Savoir dialoguer avec une base de données de type SQL en encapsulant les traitements de lecture/écriture pour respecter les bonnes pratiques d'architecture

Description : Construction d'un accès à une base MySQL ou PostgreSQL (librairies panda, psyco)

Dans un module persistance, on construit une classe ResultatDao, qui permet toutes les actions standards sur la table résultats (insertion, mise à jour, suppression, lecture ciblée ou complète de la table). Consommation de Web Services sous forme d'API.J5

Python

Contenu du cours

Création d'interface graphique

- Spécificités du développement d'interface graphique (programmation événementielle)
 - Présentation de la bibliothèque TkInter.
 - Les principaux conteneurs.
 - Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
 - Le gestionnaire de fenêtres.
 - Le placement des composants, les différents layouts.
- La gestion des événements, l'objet event.
- Les applications multifenêtres.
 - Un petit mot sur la bibliothèque Qt

Travaux pratiques

Objectifs : Mettre en œuvre une IHM en Python avec TkInter

Description : Développement d'une maquette pour compléter l'atelier précédent sous forme graphique (menu, boîtes de dialogue, liste, boutons, ...) afin de proposer une interface pour la saisie des actions de manipulations des données et l'affichage des résultats

Python

Contenu du cours

Qualité et outils

- En quoi consiste l'assurance qualité d'un programme Python (QA) ?
- Vue d'ensemble de la « trousse à outils » proposée au développeur
- Utiliser le plugin Eclipse pour Python (Pydev)
- Rechercher des bugs avec PyChecker
- Vérifier le respect des standards avec PyLint
- Utiliser le guide de style PEP 8 (Python Enhancement Proposals)
- Générer de la documentation
- Disposer de statistiques sur l'exécution de son programme (profiling)
- Comprendre l'intérêt des outils de Tests (Doctests, Unit tests)

Python

Contenu du cours

Travaux pratiques

Objectifs : Bien comprendre le champ fonctionnel de chaque outil présenté dans ce chapitre et savoir les utiliser

Description : Audit de code Python ne respectant pas les standards professionnels en termes de lisibilité. Bonnes pratiques dans le Débogage d'un programme. Mise en œuvre de tests unitaires simples. Génération de documentation. Repérage des parties consommatrices d'un programme Python d'analyse de données

Python

Contenu du cours

Travaux Pratiques

Les travaux pratiques sont conçus dans une approche pédagogique qui permet de valider les acquis et pas seulement de disposer d'un exemple concret de mise en œuvre. Les concepts importants sont donc toujours au cœur du TP et la trame de ces derniers fait office à la fois de synthèse réutilisable pour vos propres développements futurs mais aussi de révision lorsque vous vous replongerez dedans.

Dans les travaux pratiques vous manipulerez les nombreuses bibliothèques proposées par Python comme Numpy, Matplotlib, Logging, csv, panda, psycho, unittest, sympy, etc. dans une approche progressive et dynamique sans que l'on cherche à être exhaustif sur leurs possibilités mais afin que vous soyez autonome dans vos choix techniques et leur mise en œuvre.



Knowledge forges Empires

Nos Contacts



empire-training.tn



+216 55 826 628
+216 74 201 616



16 Rue D'Athènes, Sfax



Contact@empire-training.tn