**Cours de Python - Introduction et Programmation**

**Cours de Python - Introduction et Programmation**

**Module 1 : Introduction à Python**

**1.1 Présentation du langage Python**

**Description**

**Python est un langage de programmation interprété, orienté objet et facile à apprendre. Il est utilisé dans plusieurs domaines tels que le développement web, la science des données, l’intelligence artificielle et l’automatisation.**

**Exemple**

**print("Bonjour, bienvenue dans le monde de Python!")**

**Exercices**

1. **Recherchez et listez trois applications concrètes de Python.**
2. **Installez Python sur votre ordinateur et exécutez un script simple.**

**Références**

* [**Documentation officielle Python**](https://docs.python.org/fr/3/)
* [**Tutoriel OpenClassrooms**](https://openclassrooms.com/fr/courses/7168871-apprenez-les-bases-du-langage-python)

**1.2 Installation de Python et environnement de développement**

**Description**

**Installation de Python sous Windows, Mac et Linux. Présentation des environnements de développement : IDLE, Jupyter Notebook, PyCharm, VS Code.**

**Exemple**

1. **Installation de Python sous Windows via** [**python.org**](https://www.python.org/downloads/)
2. **Exécution d’un script simple avec l’IDLE Python.**

**Exercices**

1. **Installez un IDE de votre choix.**
2. **Exécutez un script Python qui affiche votre nom.**

**Références**

* [**Installation Python**](https://realpython.com/installing-python/)

**Module 2 : Les bases de la programmation en Python**

**2.1 Variables et types de données**

**Description**

**Python propose différents types de données : entier, flottant, chaîne de caractères, booléen, liste, tuple, dictionnaire.**

**Exemple**

**nom = "Alice"**

**age = 25**

**est\_etudiant = True**

**print(f"Nom: {nom}, Âge: {age}, Étudiant: {est\_etudiant}")**

**Exercices**

1. **Déclarez trois variables de types différents et affichez-les.**
2. **Convertissez un entier en chaîne de caractères et inversement.**

**Références**

* [**Types de données Python**](https://docs.python.org/fr/3/library/stdtypes.html)

**2.2 Opérateurs et expressions**

**Description**

**Python supporte divers opérateurs : arithmétiques, relationnels, logiques.**

**Exemple**

**a, b = 10, 5**

**somme = a + b**

**produit = a \* b**

**print(f"Somme: {somme}, Produit: {produit}")**

**Exercices**

1. **Effectuez des opérations mathématiques basiques en Python.**
2. **Écrivez un script qui demande deux nombres à l’utilisateur et affiche leur somme et produit.**

**Références**

* [**Opérateurs Python**](https://www.w3schools.com/python/python_operators.asp)

**2.3 Structures de contrôle**

**Description**

**Utilisation des structures conditionnelles (if, else, elif) et des boucles (for, while).**

**Exemple**

**age = int(input("Entrez votre âge : "))**

**if age >= 18:**

 **print("Vous êtes majeur.")**

**else:**

 **print("Vous êtes mineur.")**

**Exercices**

1. **Écrivez un programme qui détermine si un nombre est pair ou impair.**
2. **Créez une boucle qui affiche les nombres de 1 à 10.**

**Références**

* [**Structures de contrôle Python**](https://www.programiz.com/python-programming/if-elif-else)

**Module 3 : Fonctions et Modules**

**3.1 Fonctions en Python**

**Description**

**Les fonctions permettent de structurer le code et de le réutiliser.**

**Exemple**

**def addition(a, b):**

 **return a + b**

**print(addition(10, 5))**

**Exercices**

1. **Créez une fonction qui retourne le carré d’un nombre donné.**
2. **Définissez une fonction qui prend une liste de nombres et retourne leur somme.**

**Références**

* [**Fonctions Python**](https://docs.python.org/fr/3/tutorial/controlflow.html#defining-functions)

**Module 5 : Gestion des exceptions et fichiers**

**5.1 Gestion des erreurs en Python**

**Description**

**Apprendre à gérer les erreurs en Python grâce aux exceptions.**

**Exemple**

**try:**

 **nombre = int(input("Entrez un nombre : "))**

 **print("Le carré de ce nombre est :", nombre \*\* 2)**

**except ValueError:**

 **print("Erreur : Veuillez entrer un nombre valide.")**

**Exercices**

1. **Écrivez un programme qui gère une division par zéro.**
2. **Implémentez un système de gestion d’erreurs pour un programme de calcul de moyenne.**

**Références**

* **Gestion des erreurs en Python**

**Module 6 : Bibliothèques essentielles**

**6.1 Bibliothèques pour la science des données**

**Description**

**Présentation des bibliothèques de science des données en Python.**

**Exemple**

**import numpy as np**

**arr = np.array([1, 2, 3, 4, 5])**

**print("Tableau NumPy:", arr)**

**Exercices**

1. **Créez un programme qui utilise Pandas pour manipuler un fichier CSV.**
2. **Utilisez Matplotlib pour tracer un graphique simple.**

**Références**

* [**NumPy**](https://numpy.org/)
* **Pandas**

**Module 7 : Projets pratiques et exercices**

**7.1 Mini-projets**

1. **Création d’un gestionnaire de tâches**
2. **Développement d’un calculateur de budget**
3. **Analyse de données avec Pandas**

**Exercices**

1. **Développez un projet final intégrant les notions apprises.**
2. **Implémentez un chatbot simple en Python.**

**Références**

* **Exemples de projets Python**