1. **Qu’est-ce qu’un langage de programmation ?**

*Un langage de programmation est un système conventionnel d’expression d’instructions d’un humain à un ordinateur afin qu’il puisse le réaliser.*

1. Définir les principaux éléments constituants un langage de programmation.

*Relativement à un langage naturel, un langage de programmation est constitué : d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine.*

*Mais ces composants élémentaires sont :*

*-les identificateurs,*

*-les mots-clefs,*

*-les constantes,*

*-les chaînes de caractères,*

*-les opérateurs,*

*-les signes de ponctuation*

1. Quand dit-on qu’un langage de programmation est interprété ?

*C’est un langage qui est lu ligne par ligne, ensuite traduit en langage machine afin d’être exécuter par l’ordinateur.*

1. Si un langage n’est pas interprété, alors qu’est-ce qu’il pourrait bien être ?

*Il pourrait être compilé*

1. Expliquer les avantages et inconvénients d’un langage interprété.

*L’avantage du langage interprété c’est sa portabilité, c’est à dire sa capacité à fonctionner sur différents types d’OS. En effet, il suffit d’avoir l’interpréteur sur l’ordinateur sur lequel on veut faire tourner le logiciel et on peut en théorie le faire fonctionner.*

*Par contre, l’inconvénient majeur du langage interprété c’est sa rapidité. En effet, comme vous l’avez vu, l’interpréteur traduit le code source ligne par ligne ce qui prend un peu de temps. Un langage interprété est toujours un peu plus lent qu’un langage compilé.*

*Et enfin, en envoyant le code source sur l’ordinateur sur lequel il va être exécuté, le logiciel est moins facilement protégé contre la copie et le code source peut-être modifié.*

1. Souvent nous entendons parler de paradigme de programmation, de quoi s’agit-il ?

*Un paradigme de programmation est une approche logique qu'un développeur va adopter pour résoudre son problème.*

*Programmation impérative, procédurale, structurée, fonctionnelle, OO, concurrente.*

1. Y-a-t-il une différence significative entre la Programmation Orientée Objet et la programmation procédurale ? Argumenter votre réponse.

*Oui, il existe une différence entre les 2 :*

*La principale différence entre la POO et la programmation procédurale est que l'objectif de la programmation procédurale est de décomposer la tâche de programmation en un ensemble de variables et de sous-programmes, tandis que l'objectif de la POO est de décomposer la tâche de programmation en objets, qui encapsulent les données et méthodes.*

1. Quelles sont les caractéristiques du langage Python ?

* *Historique. ...*
* *Langage open source. ...*
* *Travail interactif. ...*
* *Langage interprété rapide. ...*
* *Simplicité du langage (cf. ...*
* *Orientation objet. ...*
* *Ouverture au monde. ...*
* *Disponibilité de bibliothèques*

1. En quoi la version 3 de Python se distingue-t-elle de sa version 2, y a-t-il une compatibilité entre les deux ?

* *La syntaxe Python 3 est plus simple et facilement compréhensible alors que la syntaxe Python 2 est relativement difficile à comprendre.*
* *Le stockage par défaut des chaînes Python 3 est Unicode, tandis que les stockages Python 2 doivent définir la valeur de chaîne Unicode avec "u".*
* *La valeur Python 3 des variables ne change jamais alors que dans Python 2, la valeur de la variable globale sera modifiée lors de son utilisation dans la boucle for.*
* *Les exceptions Python 3 doivent être placées entre parenthèses tandis que les exceptions Python 2 doivent être placées entre des notations.*
* *Les règles Python 3 de classement des comparaisons sont simplifiées alors que les règles Python 2 de classement des comparaisons sont complexes.*
* *Python 3 propose la fonction Range () pour effectuer des itérations alors que, dans Python 2, le xrange () est utilisé pour les itérations.*

1. Quelle est la dernière version stable de Python, comment l’installer ?

*La dernière version de python est le 3.10.8.*

*Comment l’installer ?*

*Télécharger le setup sur* [*https://www.python.org/downloads/*](https://www.python.org/downloads/)*, l’installer comme tout autre logiciel.*

1. Qu’est-ce qu’un interpréteur Python ? Montrer comment l’utiliser à travers une démo.

*L'interpréteur Python est une machine virtuelle, ce qui signifie qu'il s'agit d'un logiciel qui émule un ordinateur physique.*

1. Présenter et exécuter un script Python.
2. Quel est le rapport entre Python et Anaconda ?
3. Installation et prise en mains de Anaconda.

*-Téléchargez l’installeur du système correspondant, et double-cliquez pour lancer l’installation.*

*-Répondez aux différentes questions lors de l’installation. Une fois l’installation terminée, vous pouvez vérifier que celle-ci s’est bien passée en lançant l’application Jupyter Notebook*

***Prise en main :***

*Sur le côté gauche, vous voyez la navigation, d'où vous pouvez accéder à vos environnements, à la documentation (apprentissage) ou aux liens de la communauté. Plus particulièrement, la page d'accueil vous donne accès à :*

*JupyterLab : interface Web pour le projet Jupyter qui vous permet de gérer et de travailler facilement avec les blocs-notes Jupyter.*

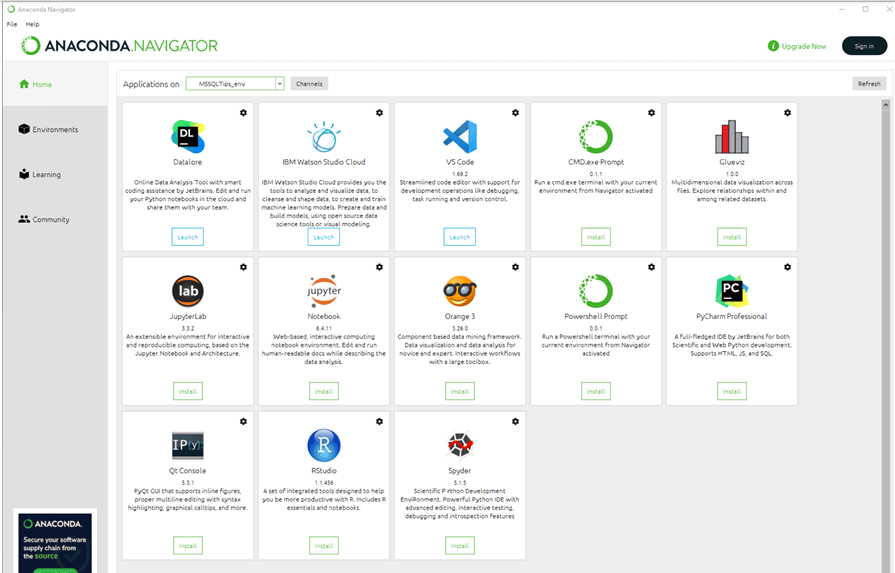
*Notebooks Jupyter : (le plus populaire du projet Jupyter) vous permettant de créer, exécuter, visualiser et présenter rapidement des expériences scientifiques.*

*Spyder : environnement scientifique open source spécialement conçue pour l'informatique Python.*

*VS Code : le célèbre éditeur de code source open source. Après avoir activé le raccourci dans les paramètres, l'icône du navigateur pointera vers une installation existante.*

*PyCharm : IDE conçu pour le développement général Python, Web ou science des données.*

*Orange : un outil d'exploration de données visuelles vous permettant de créer des workflows d'analyse et d'apprentissage automatique.*



1. Présenter Jupyter Notebook et JupyterLab.
2. **Les bases du langage Python :**

**● Installation de packages**

**● Gestion de module**

**● Les variables : nommage, types, déclaration, affectation, portée etc.**

**● Les opérateurs et expressions**

**● Structures de données**

**● Boucles**

**● Conditions**

**● Fonctions usuelles**

**● Fonction : déclaration et appel**

**● Fonction Lambda**

**● Gestion d’exceptions**

**● Gestion de fichiers**

**● POO : concepts**

**● POO avec Python**

1. **● Module**