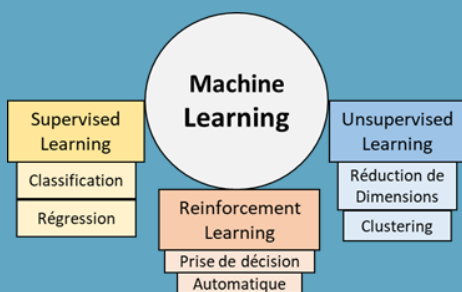
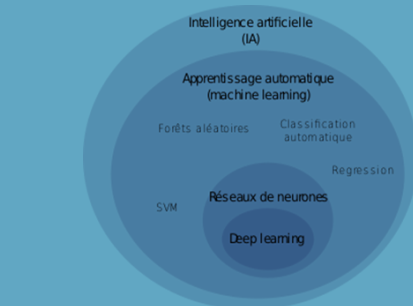


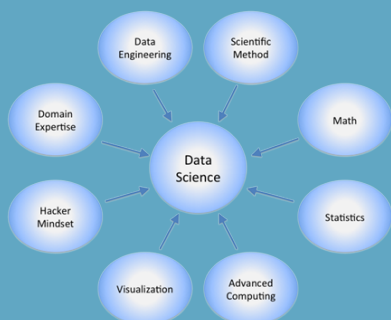
Python présente de nombreuses bibliothèques (collections de portions de code) et donc un vaste domaine d'applications :



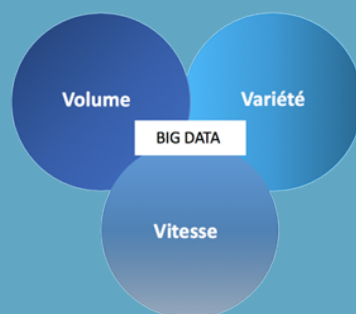
Le Machine Learning



L'Intelligence Artificielle



La Data Science



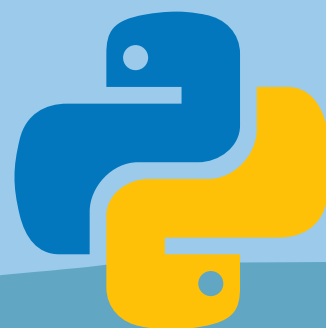
Le Big Data

Le développement Web

La gestion et l'automatisation des Systèmes

PYTHON

Weeneo
CONSULTING



LE DECRYPT'TECH
de Carole

Python a été inventé en 1989 par Guido van Rossum, un mathématicien et développeur néerlandais, avec l'objectif d'un langage simple, intuitif, open source, d'un code source facile de lecture et ce pour diverses tâches quotidiennes.

Origine du nom : Guido Van Rossum a nommé son langage ainsi en référence à la série Monty Python's Flying Circus lorsqu'il l'a écrit cette fameuse semaine de vacances de Noël 1989. Le logo imaginé par Tim Parkin, sur le base de son nom, représente deux serpents inspirés de dessins mayas.

Python est un langage de programmation, c'est-à-dire une notation utilisée pour écrire des programmes informatiques en exprimant des algorithmes permettant d'établir des instructions et opérations qu'un ordinateur exécute pour différentes applications.

Interprété

Il n'a pas besoin d'être compilé, c'est-à-dire converti en code machine dans son ensemble, pour être exécuté. Un interpréteur de commande le traduit et l'exécute élément par élément.

Multi-paradigme

Il permet d'appréhender de multiples façons la programmation informatique et de formuler les solutions appropriées à différentes problématiques

Multiplateforme

Il peut être utilisé sur différents systèmes d'exploitation : Unix, Linux, Windows, Mac OS, Android, IOS, clients légers, .



INTRODUCTION A PYTHON

Créer des données avec Python

Lancer un programme Python :

Installer Python : cf. python.org

Exécuter un programme : ouvrir un terminal, taper python puis print(« hello, world! »)

```
(base) cperrier@LAPTOP-81SG6T2R:/mnt/c/WINDOWS/system32$ python
Python 3.8.5 (default, Sep 4 2020, 07:30:14)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello, world!")
hello, world!
>>>
```

Enregistrer les données avec les variables :

nom_variable=«valeur de la variable» puis print(nom_variable)

```
>>> nom_variable="valeur de la variable"
>>> print(nom_variable)
valeur de la variable
>>> nom_variable="modif var"
>>> print(nom_variable)
modif var
```

Classer les données par type :

Un type de données est une façon de classer différents types de variables. On a par exemple les entiers, les chaînes de caractères et les booléens (true or false).

Enregistrer des groupes de données avec les listes et tuples (comme des listes non modifiables) :

```
>>> nom_liste=["1","2"]
>>> nom_liste
['1', '2']
>>> nom_liste[0]
'1'
```

```
>>> nom_liste.append("3")
>>> nom_liste
['1', '2', '3']
>>> nom_liste.remove("1")
>>> nom_liste
['2', '3']
```

```
>>> nom_tuple=("t1", "t2")
```

Enregistrer des données complexes avec les dictionnaires :

Un dictionnaire est une structure de données qui enregistre des données dans des paires clés-valeurs (ici la clé est « nom_variable » et la valeur « valeur de la variable »).. Pour créer un dictionnaire

```
>>> nom_dico = {
... "nom_var": "val"
... }
```

```
>>> nom_dico
{'nom_var': 'val'}
>>> nom_dico['nom_var']
'val'
```



INTRODUCTION A PYTHON

Gérer la logique du programme dans Python

Contrôlez le déroulement de votre programme avec des conditions

Les instructions « if » et « else » par exemple permettent de contrôler le déroulement du programme. Ici si la condition est « true » elle prendra la valeur « v1 », sinon la valeur « v2 » :

```
~# vim script.py
```

```
condition = true
if condition
print("v1")
else:
print("v2")
```

```
sudo python3 script.py
v1
```

Répétez des tâches facilement à l'aide de boucles

Les boucles sont utilisées lorsque l'on souhaite répéter plusieurs fois un ensemble d'instructions. Une boucle « for » est utilisée pour exécuter le même code pour chaque élément d'une séquence alors qu'une boucle « while » continue de s'exécuter jusqu'à ce qu'une certaine condition soit remplie.

Regrouper des tâches en utilisant les fonctions

« Une fonction est un bloc de code avec un but spécifique auquel vous pouvez donner un nom. Quand vous appelez cette fonction, vous exécutez le code qu'elle contient. Les fonctions vous laissent saisir des paramètres pour exécuter le même code sur différentes valeurs. »



INTRODUCTION A PYTHON

Extraire des données du web avec les packages Python

Importer des packages Python

Exemple de récupération du code source d'une page web :

```
import requests
url = "https://www.google.com/"
page = requests.get(url)
# Voir le code html source
print(page.content)
```

Extraire et transformer les données avec l'extraction web

```
import requests
from bs4 import BeautifulSoup
url = "https://www.gov.uk/search/news-and-communications"
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')
# Récupération des titres de la page (balises HTML <a>)
titres_bs = soup.find_all("a", class_="gem-c-document-list__item-title")
# Création de la liste titres
titres = []
# Ajout des titres dans la liste & affichage du titre ajouté
for titre in titres_bs:
    titres.append(titre.string)
    print(titre.string)
```

Charger des données avec Python

```
# Constitution des listes titres & descriptions
...
en_tete = ["titre", "description"]
# Créer un nouveau fichier pour écrire dans le fichier appelé « data.csv »
with open('data.csv', 'w') as fichier_csv:
    # Créer un objet writer (écriture) avec ce fichier
    writer = csv.writer(fichier_csv, delimiter=',')
    writer.writerow(en_tete)
    # Parcourir les titres et descriptions - zip permet d'itérer sur deux listes ou plus à la fois
    for titre, description in zip(titres, descriptions):
        # Créer une nouvelle ligne avec le titre et la description à ce moment de la boucle
        ligne = [titre, description]
        writer.writerow(ligne)
```



LE TEMOIGNAGE DE NOTRE EXPERT



ROMAIN Ingénieur en développement



Langages de programmation : C, C++, Python sous Linux

Ces langages permettent d'appréhender les fondamentaux des sciences informatiques, et cela est essentiel. Comme vous vous en doutez les projets informatiques prennent souvent du retard et pour cela des processus ont été mis en place afin d'optimiser les temps et coût de développement. Lorsque je suis passé à Python, peu à peu mon métier a changé. J'ai pris plus de plaisir à programmer, tout devenait plus simple. Python offrait des bibliothèques permettant de faire un tas de choses rapidement.

Pour la première fois les projets sur lesquels je travaillais n'avaient plus de retard et prenaient même de l'avance.

Sa définition de Python :

« Python est un langage moderne, et facile qui permet d'être rapidement productif. C'est plaisant d'utiliser Python. C'est un langage interprété qui ne se compile pas (comme du C ou C++).

La place de Python dans son quotidien :

« J'utilise python tous les jours au travail, actuellement je développe une application web (full stack), le backend est en Python (avec sqlite et SQLAlchemy). »

Ses conseils et bonnes pratiques d'utilisation de Python :

Organiser son code en créant des classes (un fichier par classe).

Utiliser l'objet en python.

Les atouts de Python parmi les autres langages :

Sa facilité

La multitude de bibliothèque qu'il propose. « Par exemple, si vous avez besoin d'interagir avec des sites web (sans API), vous pouvez utiliser Selenium, et très simplement, vous pourrez obtenir un module python sur mesure capable d'aller cliquer sur un site (s'authentifier, cliquer sur les pages web). Faire un bot rapidement. »