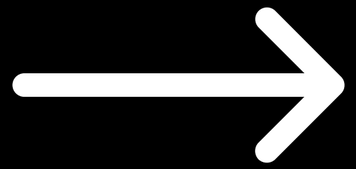
PYTHON



INTRODUCTION

PLAN DE COURS

SECTION 1 : PYTHON DE BASE

- Révision des types de données en Python (variables, listes, tuples, dictionnaires, etc.)
- Révision des structures de contrôle en Python (if, for, while, etc.)
- Introduction aux fonctions en Python
- Introduction aux modules en Python (import, etc.)

PLAN DE COURS

SECTION 2 : FLASK

- Introduction à Flask .
- Développement d'applications web avec Flask
- Utilisation de templates et de routes en Flask
- Gestion des requêtes et des réponses en Flask

PLAN DE COURS

SECTION 3 : DJANGO

- Introduction à Django .
- Développement d'applications web avec Flask
- Utilisation de modèles et de vues en Django
- Gestion des requêtes et des réponses en Django

PLAN DE COURS

OBJECTIFS

- Comprendre les bases de Python et ses applications en développement web
- Apprendre à utiliser Flask pour développer des applications web simples
- Apprendre à utiliser Django pour développer des applications web complexes

PLAN DE COURS

MÉTHODE

- Théorie et pratique en alternance
- Exercices et projets pour renforcer les connaissances
- Support et feedback réguliers pour aider à la progression

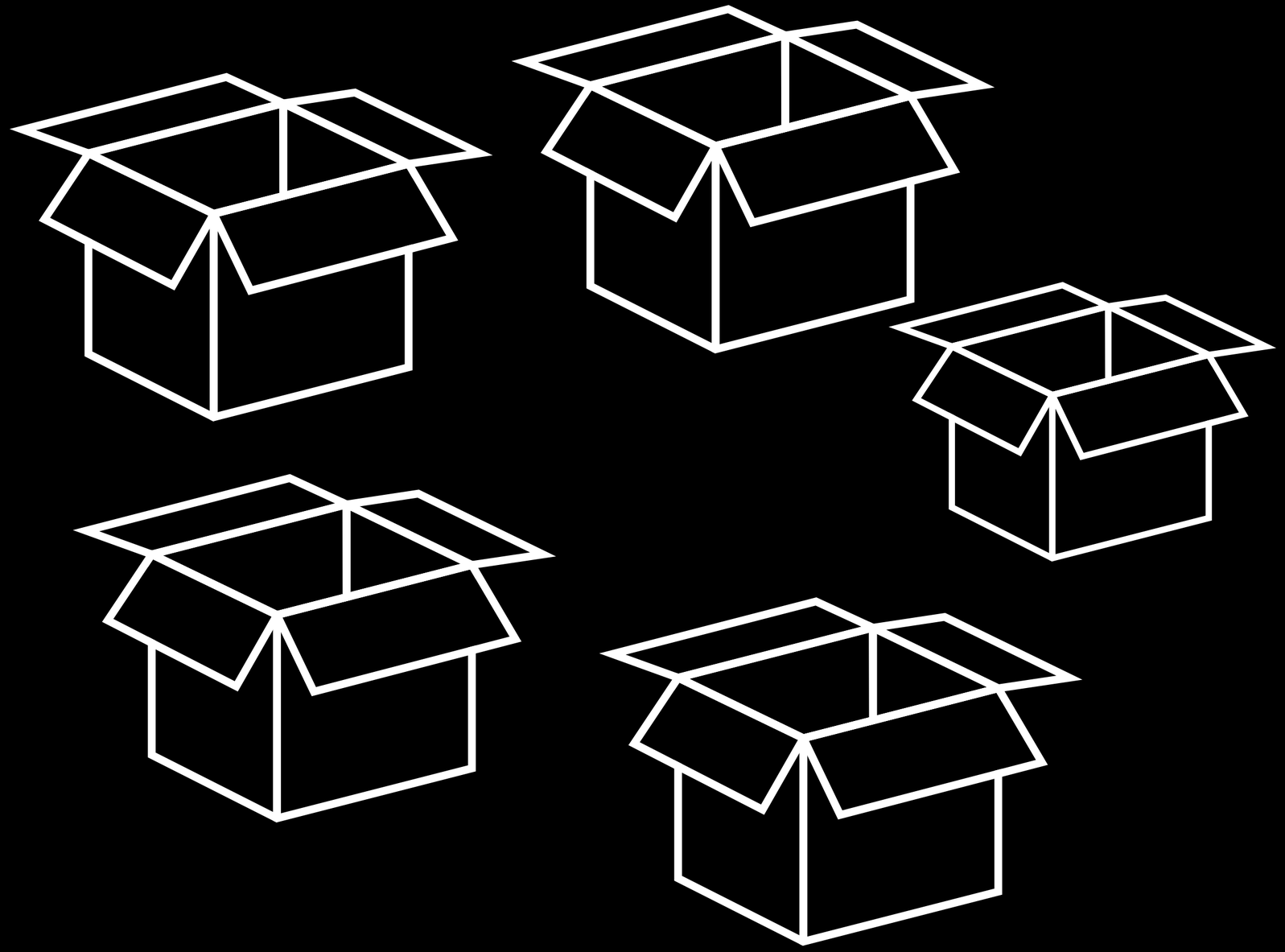
PLAN DE COURS

DURÉE

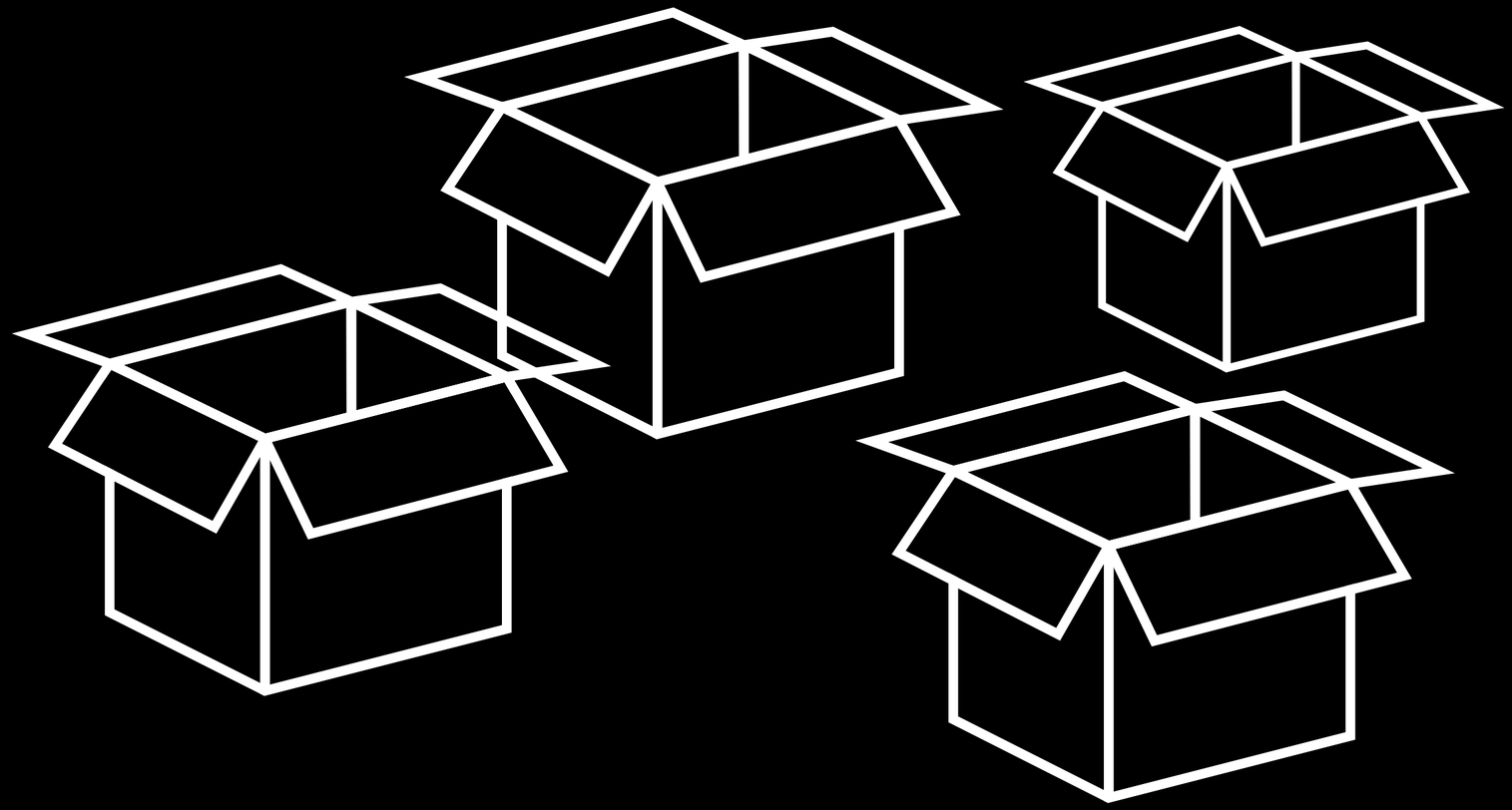
- 5 séances pour la partie Python de base et Flask.
- 5 séances pour la partie Django.
- Chaque sceance 2 H

C'EST PARTI !

Variables



Variables



note

12

Day

“Lundi”

Nom

“David”

Variables

- Entiers (int)
- Flottants (float)
- Chaînes de caractères (str)
- Booléens (bool)

Variables

Entiers (int)
Flottants (float)
Chaînes de caractères (str)
Booléens (bool)

a = 10 b = 12.3 c = ”Hello” d = True

Les Listes

Les Listes

```
ma_liste = [1, 2, 3, 4, 5]
```

```
notesdedavid = [13,12.31,15]
```

Les Listes

```
ma_liste = [1, 2, 3, 4, 5]
```

mutable (modifiable)

Les Listes

ma_liste = [1, 2, 3, 4, 5]

mutable (modifiable)

ma_liste[0]

ma_liste[0] = 10

ma_liste.append(6)

ma_liste.remove(2)

extend(L)

...

Les Tuples

Les Tuples

```
mon_tuple = (1, 2, 3, 4, 5)
```

Les Tuples

```
mon_tuple = (1, 2, 3, 4, 5)
```

immutable (non modifiable)

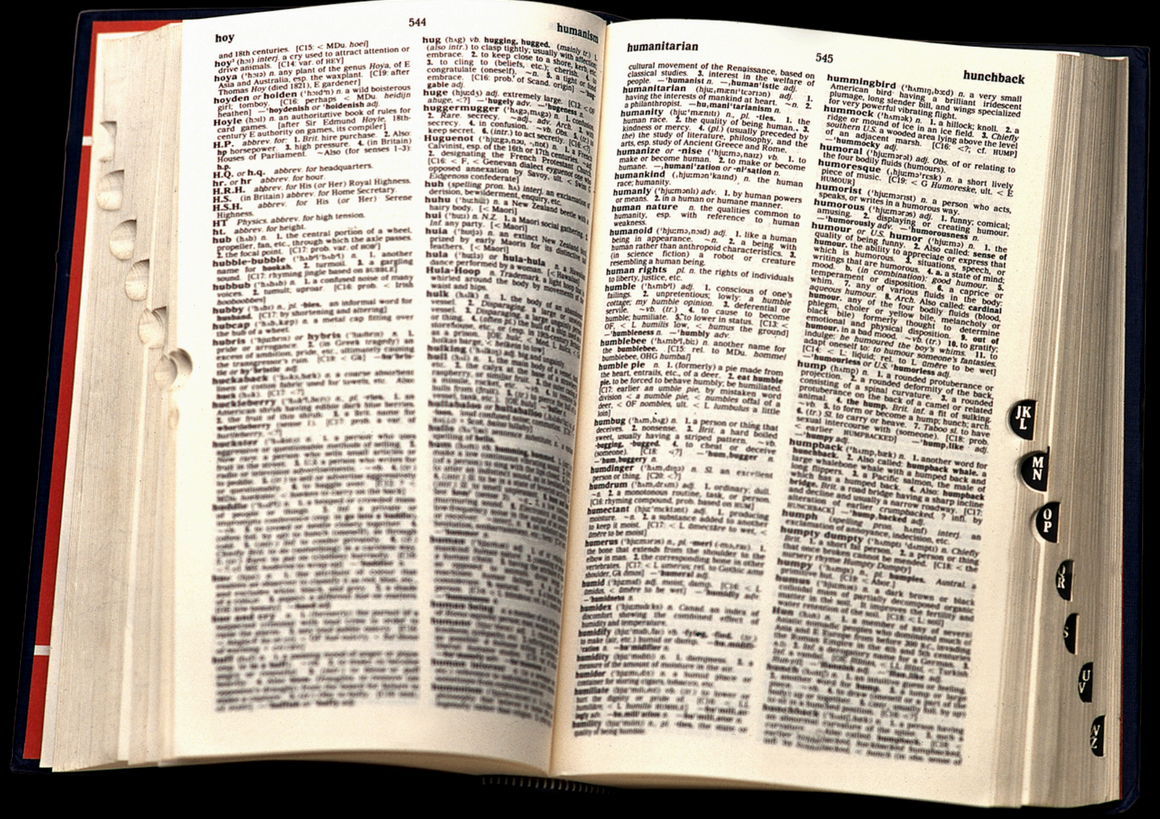
Les Tuples

```
mon_tuple = (1, 2, 3, 4, 5)
```

immutable (non modifiable)

```
mon_tuple[0]
```

Les dictionnaires



Les dictionnaires

```
ingénieur = {"nom": "Phillipe", "age": 30}
```

Les dictionnaires

ingénieur = {"nom": "Phillipe", "age": 30}

- **ingénieur["nom"] renvoie le valeur Phillipe.**

Les dictionnaires

ingénieur = {"nom": "Phillipe", "age": 30}

- **ingénieur["nom"] renvoie le valeur Phillipe.**
- **ingénieur["ville"] = "Paris"**

ingénieur = {"nom": "Jean", "age": 30, “ville”: “Paris”}

Les dictionnaires

ingénieur = {"nom": "Phillipe", "age": 30}

- **ingénieur["nom"] renvoie le valeur Phillipe.**
- **ingénieur["ville"] = "Paris"**

  → **ingénieur = {"nom": "Jean", "age": 30, “ville”: “Paris”}**

  - **del ingénieur ["age"]**

Les dictionnaires

ingénieur = {"nom": "Phillipe", "age": 30}

- ingénieur["nom"] renvoie le valeur Phillipe.

- ingénieur["ville"] = "Paris"

ingénieur = {"nom": "Jean", "age": 30, “ville”: “Paris”}

  - del ingénieur ["age"]
  - "nom" in ingénieur renvoie True.

Les structures de contrôle en Python

Instructions conditionnelles

```
if (condition) :
        action
elif (condition) :
        action
else:
        action
```

Instructions de répétition

for :

```
for i in range(5): print(i)
```

while :

```
i = 0
while i < 5:
    print(i)
i += 1
```

Instructions de gestion d'exceptions

```
try :
    x = 5 / 0
except ZeroDivisionError :
    print("Erreur de division par zéro")
```

Instructions de gestion d'exceptions

Exception

Toutes les exceptions en Python

TypeError

```
x = 5 + "a"
```

NameError

```
x = y
```

ValueError

```
x = int("a")
```

FileNotFoundError

```
f = open("fichier_inexistant.txt", "r")
```

Etc . . .

Introduction aux fonctions en Python

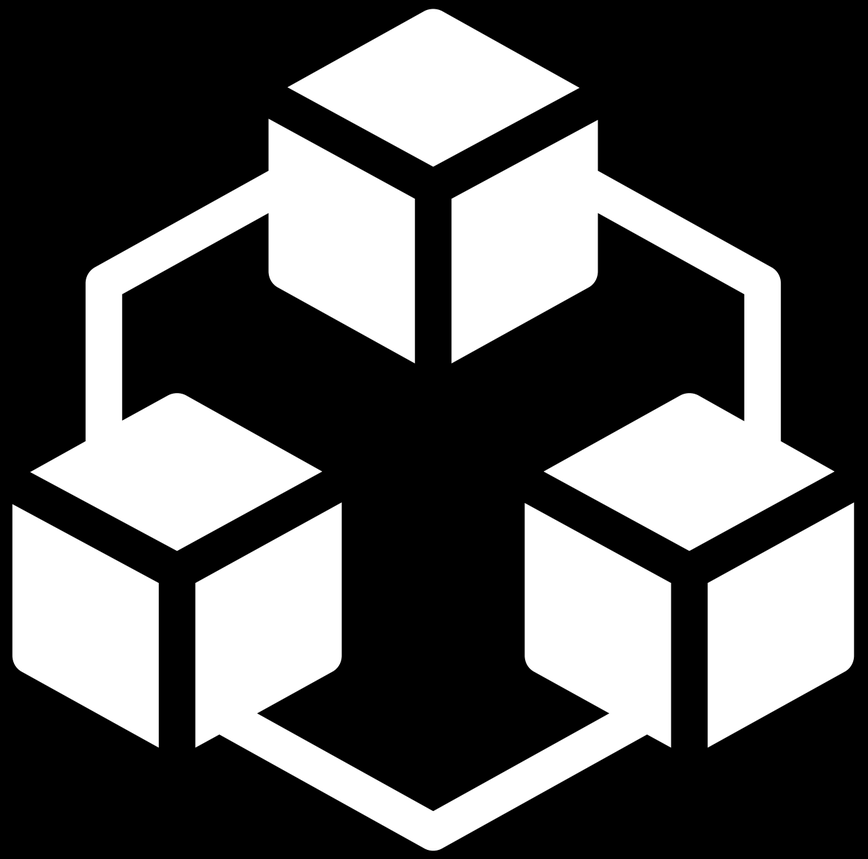
def ma_fonction(arguments):
    (contenue)

```
def bonjour(nom):
    print("Bonjour, " + nom )

bonjour("Alice")
```

La fonction return “Bonjour, Alice”

Importation de modules



Importation de modules

from numpy import np

import math

import requests

from pandas import pd

Etc . . .

QUESTION ?

SOMETHING UNCLEAR

FLASK

FLASK EST UN MICROFRAMEWORK WEB PYTHON LÉGER ET FLEXIBLE, CONÇU POUR FACILITER LA CRÉATION D'APPLICATIONS WEB RAPIDES ET ROBUSTES.

Installation de Flask

```
pip install Flask
```

Installation de Flask

```
pip install Flask
```

```
flask --version
```

Premiers pas avec Flask

```
from flask import Flask;
app = Flask(__name__)
```

Routes

URL spécifiques qui correspondent à des fonctions

www.google.com/search

http://127.0.0.1:5000/

http://127.0.0.1:5000/Hello

Les Views

Les fonctions que gérent les routes:
def index():
def hello():

Template

Un fichier qui contient du code HTML

Index.html
Hello.html

Requête HTTP

demande envoyée par un client à un serveur web

Réponse HTTP

la réponse envoyée par le serveur web en réponse à une requête HTTP.

Les méthodes HTTP

Méthode GET

récupérer des données à partir d'un serveur web

Méthode P0ST

envoyer des données au serveur web pour créer, modifier ou supprimer des ressources