

Introduction au langage Python

Le langage de programmation le
plus populaire au monde

Réalisée par:

Ben Béchir Chaima





- 1 Qu'est-ce que Python ?
- 2 Les bases : "Hello World!"
- 3 Variables et types de données
- 4 Les structures de données fondamentales
- 5 Les structures de contrôle
- 6 Les fonctions
- 7 La programmation orientée objet (POO)
- 8 Les bibliothèques et les modules
- 9 Conclusion et prochaines étapes



Qu'est-ce que Python ?

Un langage polyvalent et lisible

- Un langage de programmation de haut niveau, interprété et polyvalent.
- Crée par Guido van Rossum à la fin des années 1980.
- Accent mis sur la lisibilité du code et la simplicité de la syntaxe.
- Très populaire dans de nombreux domaines : développement web, science des données, IA, automatisation, etc.
- Possède une grande communauté et un vaste écosystème de bibliothèques.



**Les bases : "Hello
World !"**

Premier contact avec le code

Code Python

```
print("Bonjour le monde !")
```

Explications :

- La fonction **print()** est utilisée pour afficher du texte à l'écran.
- Contrairement à d'autres langages, pas de point-virgule à la fin de la ligne.
- L'indentation est très importante en Python pour définir les blocs de code.



Variables et types de données

Le stockage des informations

Déclaration de variables :

Pas besoin de déclarer le type de la variable. L'interpréteur le devine.

Nom: ‘Ali’

Age: 30

Prix: 19.99

Est_connecte: True

Types de données de base :

- ✓ **Str** (chaîne de caractères)
- ✓ **Int** (nombre entier)
- ✓ **Float** (nombre à virgule flottante)
- ✓ **Bool** (booléen : true ou false).



Les structures de données fondamentales

Organiser vos informations

- **Listes (list):** Une collection ordonnée et modifiable d'éléments.
Fruits ['pomme', 'banane', 'orange']
- **Tuples (tuple):** Une collection ordonnée et non modifiable d'éléments.
Coordonnees = (10, 20)
- **Dictionnaires (dict):** Une collection non ordonnée de paires clé-valeur.
Personne = {'nom': 'Ali', 'age': '30'}
- **Exemples (set):** Une collection non ordonnée d'éléments uniques.



Les structures de contrôle

Prendre des décisions et répéter des actions

➤ Conditions :

- ✓ **if...elif...else:** Pour exécuter du code en fonction d'une condition.
- ✓ L'indentation est cruciale pour délimiter les blocs.

➤ Boucles :

- ✓ **For:** Pour itérer sur les éléments d'une collection (liste, chaîne de caractères, etc.).
- ✓ **While:** Pour répéter un bloc de code tant qu'une condition est vraie.



Les fonctions

Organiser et réutiliser votre code

Définition : Un bloc de code réutilisable qui exécute une tâche spécifique.

Syntaxe:

```
def saluer(nom):  
    print(f"Bonjour, {nom} !")  
  
saluer("Ali")
```

- ❖ **Fonctions intégrées :** Python dispose de nombreuses fonctions prêtes à l'emploi (`len()`, `type()`, `range()`, etc).



La programmation orientée objet (POO)

Créer des objets

Concept : Un paradigme de programmation basé sur les "objets", qui combinent des données (attributs) et des fonctionnalités (méthodes).

Syntaxe

```
class Voiture:  
    def __init__(self, couleur, vitesse):  
        self.couleur = couleur  
        self.vitesse = vitesse  
  
    def accelerer(self):  
        self.vitesse += 10
```

- ❖ Création d'un objet : **ma_voiture = Voiture("Rouge", 0)**



Les bibliothèques et les modules

L'écosystème Python

Définition : Des fichiers de code Python réutilisables qui offrent des fonctionnalités supplémentaires.

➤ Comment les utiliser : **import nom_du_module**

Exemples de bibliothèques populaires :

- ✓ **Web** : Django, Flask.
- ✓ **Data Science** : NumPy, Pandas.
- ✓ **IA et Machine Learning** : TensorFlow, PyTorch, Scikit-learn.
- ✓ **Automatisation** : Selenium.



Conclusion et prochaines étapes

Résumé et pistes de développement

Résumé :

- Python est un langage simple, lisible et extrêmement puissant.
- Il possède des structures de données intuitives.
- Son écosystème de bibliothèques le rend adapté à presque tous les domaines.

Prochaines étapes :

- Pratiquer en codant de petits projets.
- Approfondir la POO.
- Choisir un domaine de spécialisation (web, data, etc.) et apprendre les bibliothèques correspondantes.
- Contribuer à des projets open-source.

Merci de votre
attention!

