

# Formation en Langage Python - niveau débutant

Cours + TD + TP + suivie des membres en lignes

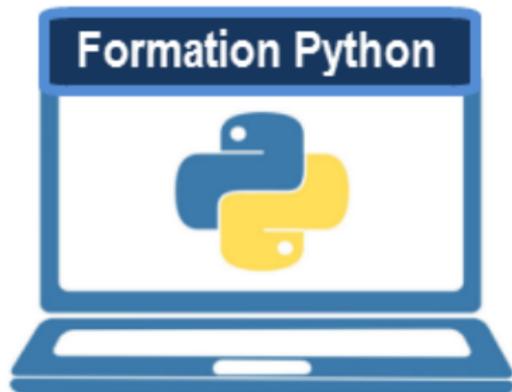
Younes Derfoufi. Docteur Agrégé

yderf64@gmail.com +212618959526

Janvier 2020

# Formation en Langage Python

Assurée par Younes Derfoufi



## Horaire-Rythme

La formation que nous proposons aux membres débutants sera traitée sous forme de doses très progressives. Comportant des cours adaptés aux rythmes des membres :

- 1 – Une séance par semaine
  - 2 – Deux séances par semaine
  - 3 – Trois séances par semaine
- Selon la disponibilité des membres

## Méthode adoptée

La formation sera traitée sous forme de **cours + TD + TP** accompagnés de solutions avec **suivi des membres** (aide à la réalisation des exercices et devoirs accompagné d'aide et soutien en ligne sur **Facebook** ou **Whatsup**)

- 1 Maîtriser la syntaxe du langage Python

# Référentiel des compétences

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet
- 4 Maitriser la programmation Python avec les bases de données SQLite3, MySql...

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet
- 4 Maitriser la programmation Python avec les bases de données SQLite3, MySql...
- 5 Connaître et mettre en œuvre les différents modules Python

# Référentiel des compétences

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet
- 4 Maitriser la programmation Python avec les bases de données SQLite3, MySql...
- 5 Connaître et mettre en œuvre les différents modules Python
- 6 Concevoir des interfaces graphiques avec Tkinter, PyQt, wxPython...

# Référentiel des compétences

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet
- 4 Maitriser la programmation Python avec les bases de données SQLite3, MySql...
- 5 Connaître et mettre en œuvre les différents modules Python
- 6 Concevoir des interfaces graphiques avec Tkinter, PyQt, wxPython...
- 7 Réalisation de projets réels: système de gestion de stock, système de gestion d'hotel, système de gestion des ressource humaines...

# Référentiel des compétences

- 1 Maîtriser la syntaxe du langage Python
- 2 Maitriser l'algorithmique & structure de données en Python
- 3 Acquérir les notions essentielles de la programmation objet
- 4 Maitriser la programmation Python avec les bases de données SQLite3, MySql...
- 5 Connaître et mettre en œuvre les différents modules Python
- 6 Concevoir des interfaces graphiques avec Tkinter, PyQt, wxPython...
- 7 Réalisation de projets réels: système de gestion de stock, système de gestion d'hotel, système de gestion des ressource humaines...
- 8 Préparation aux certificats internationaux: OpenEDG Python Institute, Microsoft Python Certification Exam...

## 1 Variables & type de variable

- 1 Variables & type de variable
- 2 Fonctions en Python

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères
- 5 Manipulation des types numériques, manipulation de chaînes de caractères.

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères
- 5 Manipulation des types numériques, manipulation de chaînes de caractères.
- 6 Manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple)

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères
- 5 Manipulation des types numériques, manipulation de chaînes de caractères.
- 6 Manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple)
- 7 Les dictionnaires en Python

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères
- 5 Manipulation des types numériques, manipulation de chaînes de caractères.
- 6 Manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple)
- 7 Les dictionnaires en Python
- 8 Les ensembles en Python

- 1 Variables & type de variable
- 2 Fonctions en Python
- 3 Les blocs, les commentaires.
- 4 Formatage des chaînes de caractères
- 5 Manipulation des types numériques, manipulation de chaînes de caractères.
- 6 Manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple)
- 7 Les dictionnaires en Python
- 8 Les ensembles en Python
- 9 **Travaux pratiques:** Installation d'IDE & langage Python & configuration + **aide et soutien en ligne**

- 1 La structure conditionnelle **if/elif/else**.

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.
- 4 La fonction **range**.

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.
- 4 La fonction **range**.
- 5 Les **lambda** expression.

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.
- 4 La fonction **range**.
- 5 Les **lambda** expression.
- 6 Les **générateurs** , **itérateurs**, **décorateurs** en Python

# Structure de contrôle en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.
- 4 La fonction **range**.
- 5 Les **lambda** expression.
- 6 Les **générateurs** , **itérateurs**, **décorateurs** en Python
- 7 Gestion des erreurs et **exception** en Python

- 1 La structure conditionnelle **if/elif/else**.
- 2 Les opérateurs logiques et les opérateurs de comparaison.
- 3 Les boucles d'itérations **while** et **for**. Interruption d'itérations **break/continue**.
- 4 La fonction **range**.
- 5 Les **lambda** expression.
- 6 Les **générateurs** , **itérateurs**, **décorateurs** en Python
- 7 Gestion des erreurs et **exception** en Python
- 8 **TD : Exercices avec solutions + aide et soutien en ligne**

- 1 La notion abstraite de programmation orienté objet

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python
- 7 Encapsulation des données.

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python
- 7 Encapsulation des données.
- 8 Héritage

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python
- 7 Encapsulation des données.
- 8 Héritage
- 9 Polymorphisme

# Programmation orienté objet en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python
- 7 Encapsulation des données.
- 8 Héritage
- 9 Polymorphisme
- 10 Modélisation UML en Python

- 1 La notion abstraite de programmation orienté objet
- 2 La programmation orienté objet en Python
- 3 Notion de classe, d'attributs et de méthodes.
- 4 Constructeur et destructeur
- 5 Le paramètre self
- 6 Les méthodes statiques en Python
- 7 Encapsulation des données.
- 8 Héritage
- 9 Polymorphisme
- 10 Modélisation UML en Python
- 11 **Travaux pratiques:** étude des cas réels d'objets & héritage en Python + **aide et soutien en ligne**

- 1 Les principes de programmation des interfaces graphiques.

# IHM & Interface graphique Tkinter

- ① Les principes de programmation des interfaces graphiques.
- ② Présentation de la bibliothèque TkInter.

# IHM & Interface graphique Tkinter

- ① Les principes de programmation des interfaces graphiques.
- ② Présentation de la bibliothèque TkInter.
- ③ Les principaux conteneurs.

- ① Les principes de programmation des interfaces graphiques.
- ② Présentation de la bibliothèque TkInter.
- ③ Les principaux conteneurs.
- ④ Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).

- 1 Les principes de programmation des interfaces graphiques.
- 2 Présentation de la bibliothèque TkInter.
- 3 Les principaux conteneurs.
- 4 Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- 5 Gestionnaire de fenêtres.

- 1 Les principes de programmation des interfaces graphiques.
- 2 Présentation de la bibliothèque TkInter.
- 3 Les principaux conteneurs.
- 4 Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- 5 Gestionnaire de fenêtres.
- 6 Le placement des composants, les différents layouts: pack(), place(), grid()...

- 1 Les principes de programmation des interfaces graphiques.
- 2 Présentation de la bibliothèque TkInter.
- 3 Les principaux conteneurs.
- 4 Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- 5 Gestionnaire de fenêtres.
- 6 Le placement des composants, les différents layouts: pack(), place(), grid()...
- 7 Gestion des événements, l'objet event.

- 1 Les principes de programmation des interfaces graphiques.
- 2 Présentation de la bibliothèque TkInter.
- 3 Les principaux conteneurs.
- 4 Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- 5 Gestionnaire de fenêtres.
- 6 Le placement des composants, les différents layouts: pack(), place(), grid()...
- 7 Gestion des événements, l'objet event.
- 8 Le module de design TTK

- 1 Les principes de programmation des interfaces graphiques.
- 2 Présentation de la bibliothèque TkInter.
- 3 Les principaux conteneurs.
- 4 Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- 5 Gestionnaire de fenêtres.
- 6 Le placement des composants, les différents layouts: pack(), place(), grid()...
- 7 Gestion des événements, l'objet event.
- 8 Le module de design TTK
- 9 Tp : Conception d'interface graphique avec la bibliothèque Tkinter.  
Elaboration de *projets réels*: **création d'un éditeur de texte, gestion de stock, gestion des ressources humaines, gestion de bibliothèques, gestion d'hotels...+ aide et soutien en ligne**

## 1 Introduction

# Les modules en langage Python

- 1 Introduction
- 2 Créer votre propre module

# Les modules en langage Python

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib
  - Le module statistics

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib
  - Le module statistics
  - Le module collections

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib
  - Le module statistics
  - Le module collection
  - Le module configparser

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib
  - Le module statistics
  - Le module collection
  - Le module configparser
  - Le module BeautifulSoup & web scrapping.

- 1 Introduction
- 2 Créer votre propre module
- 3 Les modules standards en Python
  - Le module os
  - Le module Sys
  - Le module Image Pillow
  - Le module matplotlib
  - Le module statistics
  - Le module collection
  - Le module configparser
  - Le module BeautifulSoup & web scrapping.
- 4 **TD : Exercices avec solutions + aide et soutien en ligne**

## ① Mode d'ouverture d'un fichier

- 1 Mode d'ouverture d'un fichier
  - Ouverture et lecture d'un fichier en mode read 'r'

## ① Mode d'ouverture d'un fichier

- Ouverture et lecture d'un fichier en mode read 'r'
- Ouverture en mode écriture (write) 'w'

## ① Mode d'ouverture d'un fichier

- Ouverture et lecture d'un fichier en mode read 'r'
- Ouverture en mode écriture (write) 'w'
- Ouverture en mode append 'a'

## ① Mode d'ouverture d'un fichier

- Ouverture et lecture d'un fichier en mode read 'r'
- Ouverture en mode écriture (write) 'w'
- Ouverture en mode append 'a'
- Ouverture en mode lecture & écriture 'r+'

- 1 Mode d'ouverture d'un fichier
  - Ouverture et lecture d'un fichier en mode read 'r'
  - Ouverture en mode écriture (write) 'w'
  - Ouverture en mode append 'a'
  - Ouverture en mode lecture & écriture 'r+'
- 2 Manipulation des fichiers de configuration .ini en Python

- 1 Mode d'ouverture d'un fichier
  - Ouverture et lecture d'un fichier en mode read 'r'
  - Ouverture en mode écriture (write) 'w'
  - Ouverture en mode append 'a'
  - Ouverture en mode lecture & écriture 'r+'
- 2 Manipulation des fichiers de configuration .ini en Python
- 3 Fichiers JSON en Python

- 1 Mode d'ouverture d'un fichier
  - Ouverture et lecture d'un fichier en mode read 'r'
  - Ouverture en mode écriture (write) 'w'
  - Ouverture en mode append 'a'
  - Ouverture en mode lecture & écriture 'r+'
- 2 Manipulation des fichiers de configuration .ini en Python
- 3 Fichiers JSON en Python
- 4 Fichier CSV en Python

- 1 Mode d'ouverture d'un fichier
  - Ouverture et lecture d'un fichier en mode read 'r'
  - Ouverture en mode écriture (write) 'w'
  - Ouverture en mode append 'a'
  - Ouverture en mode lecture & écriture 'r+'
- 2 Manipulation des fichiers de configuration .ini en Python
- 3 Fichiers JSON en Python
- 4 Fichier CSV en Python
- 5 **TD : Exercices avec solutions + aide et soutien en ligne**

## ① Python et les bases de données SQLite3

- ① Python et les bases de données SQLite3
- ② Python et les bases de données MySql

- ① Python et les bases de données SQLite3
- ② Python et les bases de données MySql
- ③ Python et les bases de données Mogodb

- ① Python et les bases de données SQLite3
- ② Python et les bases de données MySql
- ③ Python et les bases de données Mogodb
- ④ **TD : Exercices avec solutions + aide et soutien en ligne**

Formules	Nombres d'heures/semaine	Tarif / mois
<b>Formule1</b>	1 séance par semaine de 2h	150€
<b>Formule 2</b>	2 séances par semaines 2h + 2 h = 4h	300€
<b>Formule 3</b>	3 séances par semaines 2+2+2= 6h	400€
<b>Formule 4</b>	Rythme accéléré - à discuter	à discuter

## Informations de contact:

**Nom :** Younes Derfoufi

**Email :** yderf64@gmail.com

**Tél :** +212618959526

**Facebook :** <https://www.facebook.com/almoubarayate/>

**Youtube :** <https://www.youtube.com/user/InformatiquesFacile>

Merci