



# PYTHON LES BASES



01

## **PRÉ-REQUIS**

De quoi ai-je besoin pour  
suivre cette formation?

02

## **HISTOIRE**

Bref historique du langage  
Python

03

## **AVANTAGES**

Quels sont les avantages  
d'apprendre le langage  
Python?

04

## **CARACTÉRISTIQUES**

Quelles sont les  
caractéristiques du langage  
Python?

05


## **PREMIER SCRIPT**

Comment exécuter mon  
premier script Python?

06

## **LES IDENTIFIANTS**

Qu'est-ce qu'un identifiant  
Python?





07

## **LES MOTS RÉSERVÉS**

Quels sont les mots réservés  
en Python?

09

## **DÉCLARATIONS MULTILIGNES**

Comment réaliser une  
déclaration multiligne?

11

## **LES COMMENTAIRES**

Comment créer des  
commentaires?

## **LIGNES & INDENTATIONS**


Comment est structuré un  
script Python?

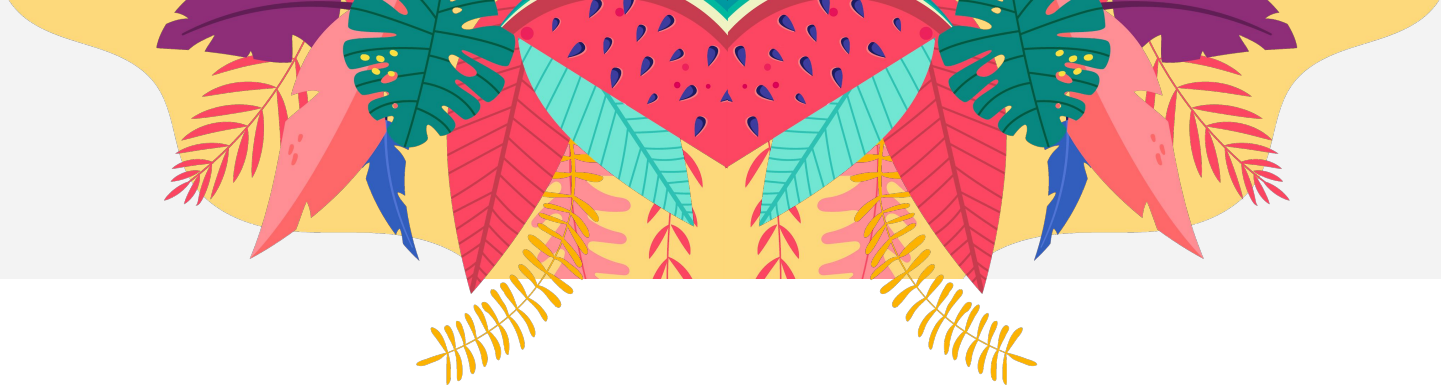
08

10

## **LES CITATIONS**

Comment définir une  
chaîne de caractères?





# PRÉ-REQUIS

## ORDINATEUR

Posséder un ordinateur  
avec un des systèmes  
d'exploitation suivants :  
Windows, MacOS,  
Linux

## PYTHON

Une version Python 3 et  
un IDE comme PyCharm.  
Installer [Python](#) et  
[PyCharm](#)

## ÊTRE MOTIVÉ

Suivez pas à pas la  
formation et devenez un  
véritable Pythoniste ! 🐍

# HISTOIRE

Python est un langage de script de haut niveau, interprété, interactif et orienté objet. Créé par **Guido van Rossum** entre 1985 et 1990, la première version disponible fut distribuée en 1991.

Python est dans le **top des langages de programmation les plus populaires** et se retrouve utilisé dans de **nombreux domaines** tels que :

- La programmation système
- Le domaine scientifique
- Le secteur industriel
- Le développement web
- L'intelligence artificielle
- La data science

...



logo Python





## AVANTAGES

### **Python est interprété**

Python est traité à l'exécution par l'interpréteur.  
Vous n'avez pas besoin de compiler votre programme avant de l'exécuter.

### **Python est orienté objet**

Python est un langage de programmation orientée objet, c'est-à-dire qu'il encapsule votre code dans des objets.

### **Python est interactif**

Vous pouvez interagir directement avec l'interpréteur pour écrire vos programmes.

### **Python est accessible**

Python est un excellent langage pour découvrir la programmation.

# CARACTÉRISTIQUES

## Facile à lire

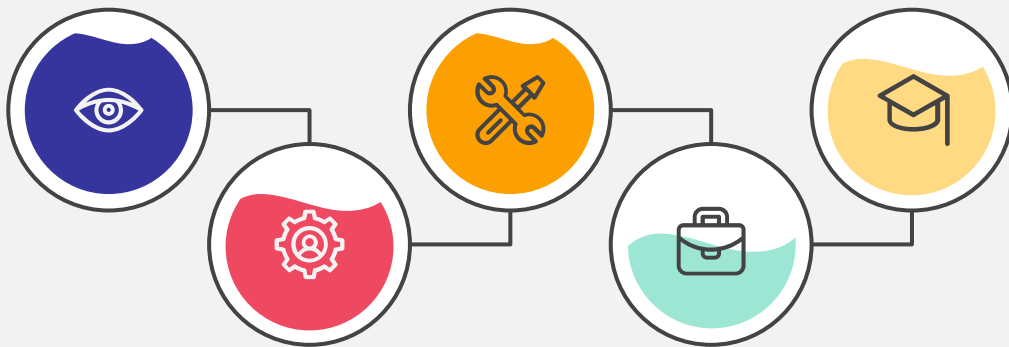
Le code Python est clairement défini et compréhensible.

## Facile à maintenir

Le code source est assez simple à maintenir.

## Facile à apprendre

Python a peu de mots-clés, une structure simple et une syntaxe clairement définie. Cela permet d'apprendre le langage rapidement.



## Mode interactif

Python supporte un mode interactif qui permet de tester et de déboguer de manière interactive des extraits de code.

## Portable

Python peut fonctionner sur une grande variété de plates-formes et possède la même interface sur toutes ces dernières.



# CARACTÉRISTIQUES

## Évolutivité

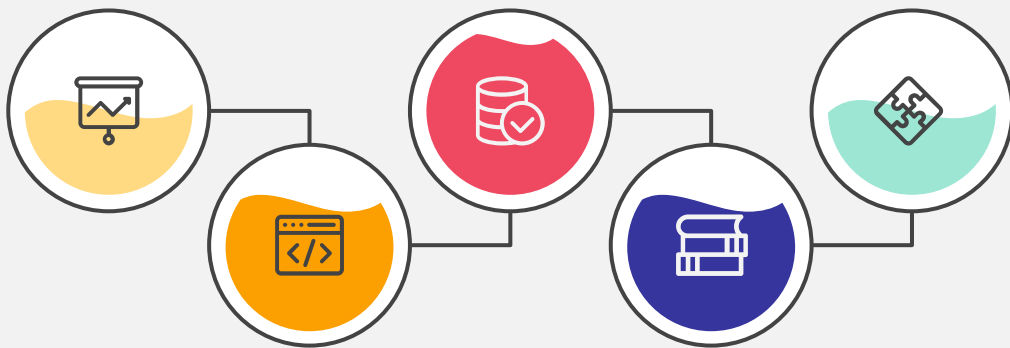
Python offre une meilleure structure et un meilleur support pour les programmes importants.

## Bases de données

Python fournit des interfaces à toutes les principales bases de données commerciales.

## Extensible

Vous pouvez ajouter des modules de bas niveau à l'interpréteur Python. Ces modules permettent aux programmeurs d'ajouter ou de personnaliser leurs outils pour être plus efficaces.



## Programmation d'interfaces graphiques

Python prend en charge les applications graphiques qui peuvent être créées et portées sur de nombreuses plate-formes.

## Une vaste bibliothèque

La majeure partie de la bibliothèque Python est très portable et compatible avec toutes les plates-formes sous UNIX, Windows et Macintosh.





The background of the entire image is a dense, colorful pattern of various tropical leaves and plants. The leaves are in shades of green, yellow, orange, red, and purple. Some leaves have distinct patterns like stripes or spots. In the top left and bottom right corners, there are slices of watermelon with red flesh and black seeds. The central text is set against a white rectangular background.

# **MON PREMIER SCRIPT**



## TERMINAL ou INVITE DE COMMANDES

1. Entrer `python`.
2. Taper le texte suivant et appuyer sur la touche Entrée.

```
print("Hello World!")
```

# MODE INTERACTIF



```
C: \Users\quentin>python  
>>> print("Hello World!")
```

```
Hello World!
```

## PYCHARM

1. Ouvrir PyCharm.
2. Créer un nouveau projet et le sélectionner.
3. Créer un nouveau fichier : **Alt + insert**
4. Choisir **Python file** et entrer le nom de votre fichier.
5. Taper le texte suivant : **print("Hello World!")**
6. Exécuter le script : **Ctrl + Maj + F10**

# MODE INTERACTIF EN LIGNE



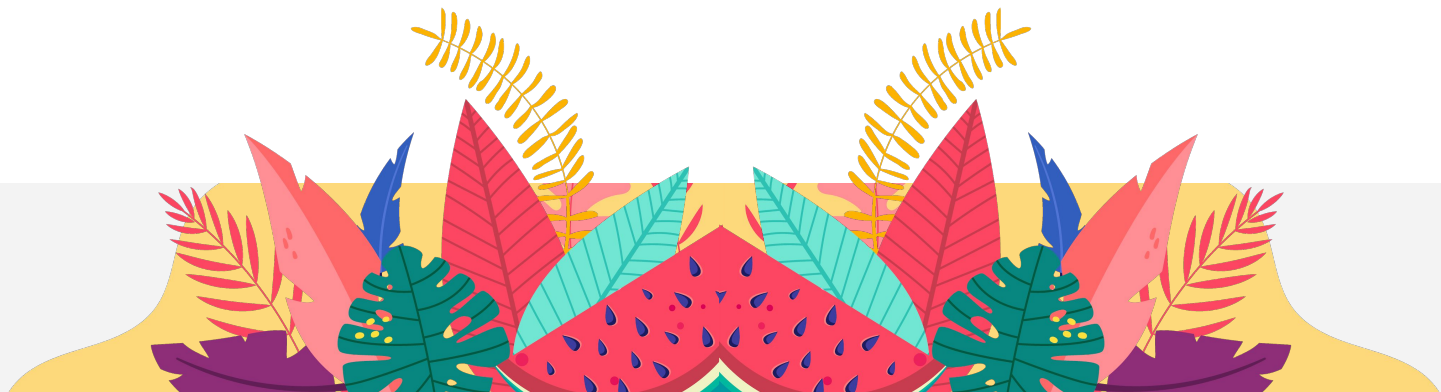
<https://repl.it/>

Cliquer sur le bouton **new repl**, sélectionner **Python** et appuyer sur **Create Repl.**



<https://jupyter.org/try>

Cliquer sur le bouton **Try Classic Notebook** contenant le logo Python.



# MODE SCRIPT

## PYCHARM

1. Ouvrir PyCharm.
2. Créer un nouveau projet et le sélectionner.
3. Créer un nouveau fichier : **Alt + insert**
4. Choisir **Python file** et entrer le nom de votre fichier.
5. Taper le texte suivant :  
`print("Hello World!")`
6. Enregistrer le fichier :  
**Ctrl + S**

## TERMINAL ou INVITE DE COMMANDE

7. Entrer **python** suivi du **chemin absolu du script python**.
8. Appuyer sur la touche Entrée.

**Conseil :** Dans PyCharm, pour copier le chemin absolu de votre script, sélectionner le et appuyer sur **Ctrl + Maj + C**.

```
C: \Users\quentin>python F:\Workspaces\dev\test.py  
Hello World!
```





# LES IDENTIFIANTS

Un identifiant Python est un nom utilisé pour identifier une variable, une fonction, une classe, un module ou un autre objet.

Un identifiant commence par une lettre de A à Z ou de a à z ou un trait de soulignement “\_” suivi de zéro ou de plusieurs lettres, traits de soulignement et chiffres (0 à 9).

Python n'autorise pas les caractères de ponctuation tels que @, \$ et % dans les identifiants. Python est un langage de programmation sensible à la casse. Ainsi, *Ananas* et *ananas* sont deux identifiants différents en Python.

## Les conventions d'appellation :

- Les noms de classe commencent par une **lettre majuscule**. Tous les autres identifiants commencent par une **lettre minuscule**.
- Le fait qu'un identifiant commence par un **seul trait de soulignement** indique que l'identifiant est **privé**.
- Un identifiant qui commence par **deux traits de soulignement** indique qu'il s'agit d'un identifiant **fortement privé**.
- Si l'identifiant se termine également par **deux traits de soulignement**, l'identifiant est un **nom spécial** défini par Python.





# LES MOTS RÉSERVÉS

Ce sont des mots-clés Python que vous ne pouvez pas utiliser comme constantes ou variables ou tout autre nom d'identifiant.

Tous les mots-clés Python contiennent uniquement des lettres minuscules.

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

# LIGNES & INDENTATIONS



Les blocs de code sont indiqués par une indentation de ligne.

Toutes les lignes du bloc doivent être indentées de la même façon.

Par convention, une indentation représente 4 espaces.

```
# Indentations
```

```
fruits = ["apple", "banana", "cherry"]
```

```
for x in fruits:
```

```
    print(x)
```

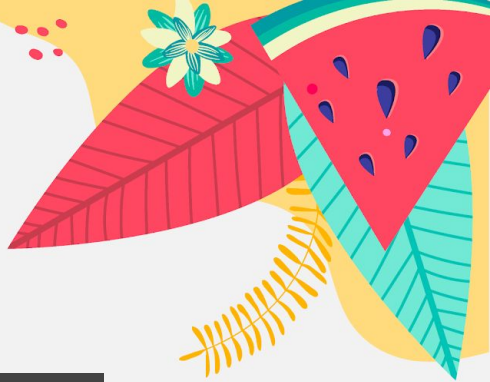
```
    if x == "banana":
```

```
        break
```

```
print("I love this fruits!")
```



# DÉCLARATIONS MULTILIGNES



Les déclarations se terminent généralement par une nouvelle ligne.

Cependant Python permet l'utilisation du caractère de **continuation de ligne** “\” pour indiquer que **celle-ci doit continuer**.

Les déclarations contenues entre **parenthèses ()**, **crochets []** ou **accolades {}** n'ont pas besoin d'utiliser le caractère de continuation de ligne.

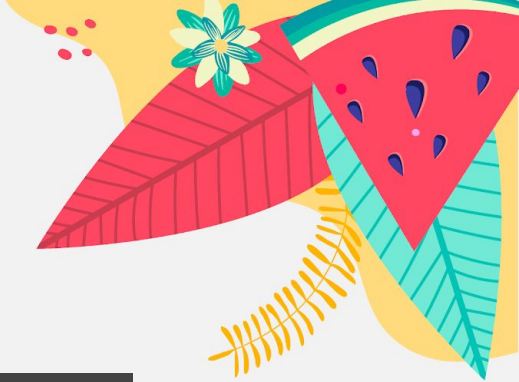
# Déclarations multilignes

```
total = item_one + \  
        item_two + \  
        item_three
```

```
days = ['Monday',  
        'Tuesday',  
        'Wednesday',  
        'Thursday',  
        'Friday']
```



# LES CITATIONS



Pour désigner des chaînes de caractères, Python accepte les guillemets suivants :

- **simples** (')
- **doubles** (")
- **triples** (''' ou ''')

À condition que le même type de guillemet commence et finisse la chaîne de caractères.

Les guillemets triples sont utilisés pour couvrir la chaîne sur plusieurs lignes.

## # Citations

```
fruit = 'apple'
```

```
sentence = "I love fruits!"
```

```
paragraph = """I love fruits but I don't  
like vegetables."""
```

# LES COMMENTAIRES



Un signe dièse “#” qui n'est pas à l'intérieur d'une chaîne de caractères commence un commentaire.

Tous les caractères après le “#” et jusqu'à la fin de la ligne font partie du commentaire et l'interpréteur Python les ignore.

La chaîne qui suit le **triple guillemet** est également ignorée par l'interpréteur Python et peut être utilisée comme un **commentaire multiligne**.

## # Commentaires

```
fruits = ["apple", "banana", "cherry"]
```

```
# Loop on fruits list  
for x in fruits:
```

```
    print(x) # print current fruit  
    if x == "banana":  
        """Stop the loop if the current fruit  
        is equal to banana."""  
        break
```



**Félicitation vous avez terminé  
le chapitre sur les bases de  
Python!**

A decorative border of various tropical leaves in vibrant colors like red, orange, yellow, green, and purple surrounds the central white area.

# CRÉDITS

- Modèle de la présentation par [Slidesgo](#)
- Icônes par [Flaticon](#)
- Images et infographies par [Freepik](#)