commentcoder.com

# 9 idées de projets en Python pour débutants

10-13 minutes

---

Article mis à jour le jeudi 21 avril 2022.


Vous cherchez un premier projet Python ? Ou comment coder des mini projets qui vous permettent de vous améliorer rapidement ? Qu'est-ce qu'un débutant peut faire en Python pour évoluer ?

Développer des projets en Python est le meilleur moyen de vous améliorer en code en général, surtout si vous êtes débutant.

D'ailleurs si vous voulez vous améliorer en Python, pensez à consulter les meilleurs outils pour apprendre Python en 2022.

Après avoir développé plusieurs des projets Python dans cette liste, vous serez capables :

- De montrer des projets dans votre portfolio

- De créer un jeu simple avec la libraire PyGame
- De construire des interfaces utilisateurs en Python

Découvrons quelques exemples de projets et exercices destinés aux débutants en Python.

## 1. Le juste prix

Ce premier projet est un jeu amusant pour les débutants connu de tous. Le programme génère un prix rond aléatoire. Le but pour l'utilisateur est de deviner le prix. Chaque fois que l'utilisateur se trompe, l'ordinateur lui dit si c'est plus ou moins que le prix qu'il a donné. À chaque aide de l'ordinateur, le score final atteignable par le joueur baisse.

Au programme, vous apprendrez à saisir des entrées clavier par un utilisateur, créer des fonctions pour valider que le nombre entré est bien un nombre entier, comparer une variable de référence (le prix) avec une autre variable et de calculer la différence entre deux nombre.

## 2. Pierre Papier Ciseaux

Créer un pierre papier ciseaux est un bon exercice pour vous entraîner en Python et réaliser vos premiers projets.

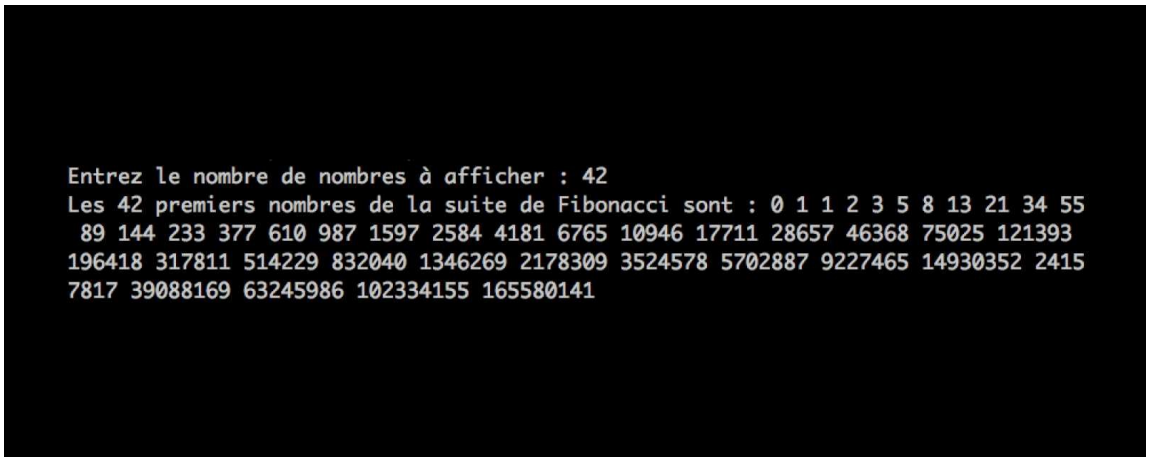
Au programme, vous devrez créer :

- une fonction qui génère de l'aléatoire : pierre, papier ou ciseaux
- une fonction pour vérifier et valider le coup qui vient d'être joué
- une fonction de résultat pour déclarer le vainqueur du tour
- un compteur de points pour suivre le score total

Le programme demande à l'utilisateur d'effectuer le premier coup avant d'effectuer un coup. Une fois le coup validé, l'entrée

est évaluée, l'entrée saisie pouvant être une chaîne de caractères, une lettre ou un nombre. Après évaluation de la chaîne de caractères, la fonction de résultat détermine le gagnant et la fonction de comptabilisation des points actualise le score total.

## 3. Générateur des nombres de la suite de Fibonacci




La série mathématique connue sous le nom de suite de Fibonacci a été l'une des questions informatique les plus populaires. Essentiellement, vous commencez avec deux nombres, de préférence 0 et 1, et vous les ajoutez pour créer votre troisième nombre de Fibonacci. Ensuite, il suffit d'additionner la somme et l'avant-dernier terme de Fibonacci pour générer le suivant.

Ce qui donne :

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987...
```

Dans ce projet, vous demandez la position du nombre de Fibonacci requise par l'utilisateur et vous le générez simplement. Une fois généré, vous pouvez afficher le nombre correspondant à l'utilisateur. Vous pouvez aller plus loin et montrer à l'utilisateur la série entière jusqu'à ce point avec le

fonctionnement mathématique. C'est l'un des meilleurs projets Python pour s'initier au concept de la fonction récursive.

**Voici l'exemple d'une implémentation basique d'un générateur de nombres de la suite de Fibonacci' :**

```
nombres = int(input("Entrez la quantité de nombres
à afficher : "))

a = 0
b = 1
total = 0

print("Les {0} premiers nombres de la suite de
Fibonacci sont :".format(nombres), end = " ")
for _ in range(nombres):
  print(total, end = " ")
  a = b
  b = total
  total = a + b
```

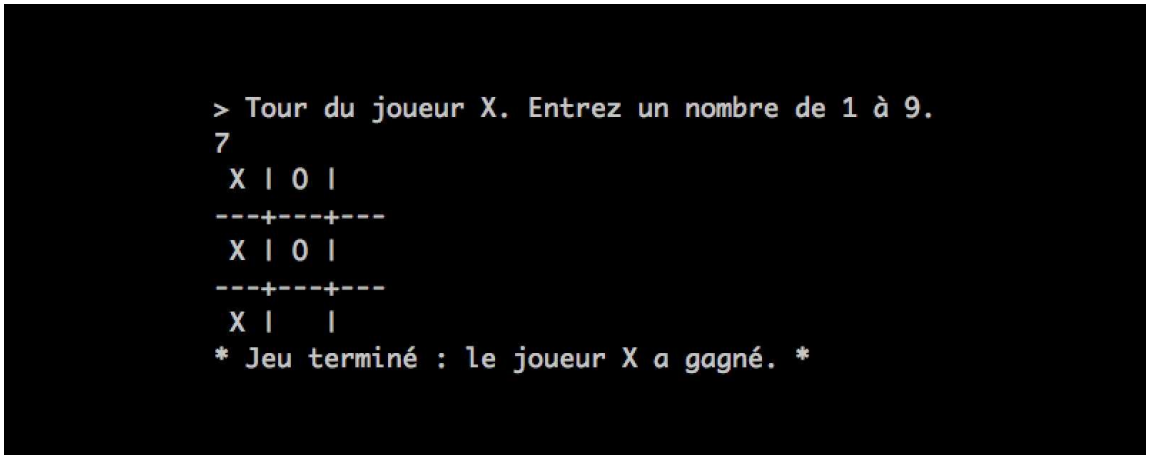
## 4. Algorithme de recherche dichotomique

Pour vous introduire à la complexité algorithmique, vous pouvez créer un algorithme simple de recherche dichotomique.

Vous utiliserez un tableau trié et diviserez le tableau à chaque itération. Si le nombre désiré est dans la première partie, vous continuez avec la première moitié du tableau et rejetez la seconde. Ensuite divisez la première moitié en 2 et répétez l'operation jusqu'à trouver le nombre voulu.

- Article Wikipedia sur la recherche dichotomique

## 5. Le jeu du Morpion




Le Morpion, aussi appelé “tic-tac-toe” ou “oxo” en Belgique, est un jeu très courant et facile à jouer. Le principe du jeu est simple. C’est un jeu au tour par tour, où le but est d’aligner un trio de cercles ou de croix en diagonale, horizontalement ou verticalement sur une grille de 3×3 carrés pour obtenir la victoire.

Le défi de la création de ce jeu consiste principalement à se familiariser avec l’indexation des tableaux en 2D et à comprendre comment vérifier les alignements en diagonale. Une fois ces problèmes résolus, le codage devrait être simplifié.

Pour aller plus loin, vous pouvez aussi vous amusez à créer une interface graphique avec PyGame ou une autre bibliothèque graphique Python.

**Voici l’exemple d’une implémentation basique du jeu de morpion :**

```
plateau = [" " for _ in range(9)]  # crée un
tableau de 9 caractères espaces " "


def afficherPlateau(p, gagnant=None):
    print(" " + p[0] + " | " + p[1] + " | " + p[2]
+ " ")
    print("---+---+---")
```

```
    print(" " + p[3] + " | " + p[4] + " | " + p[5]
+ " ")
    print("---+---+---")
    print(" " + p[6] + " | " + p[7] + " | " + p[8]
+ " ")
    if gagnant:
        print("""* Partie terminée : le joueur {0}
a gagné. *""".format(gagnant))


def morpion():
    joueur = "X"
    tour = 0

    while True:
        afficherPlateau(plateau)
        print("> Tour du joueur " + joueur + ".
Entrez un nombre de 1 à 9.")

        move = int(input()) - 1  # notre tableau
est de 0 à 8, donc on retire 1

        if plateau[move] == " ":
            plateau[move] = joueur
            tour += 1
        else:
            print("! Case déjà occupée, choisissez-
en une autre.")
            continue  # on passe au prochain
passage de boucle sans exécuter le code ci-dessous
```

```
        if plateau[0] == plateau[1] == plateau[2]
!= " " \
        or plateau[3] == plateau[4] == plateau[5]
!= " " \
        or plateau[6] == plateau[7] == plateau[8]
!= " " \
        or plateau[0] == plateau[3] == plateau[6]
!= " " \
        or plateau[1] == plateau[4] == plateau[7]
!= " " \
        or plateau[2] == plateau[5] == plateau[8]
!= " " \
        or plateau[0] == plateau[4] == plateau[8]
!= " " \
        or plateau[2] == plateau[4] == plateau[6]
!= " ":
            afficherPlateau(plateau, joueur)
            break

        if tour == 9:
            print("Égalité")
            break

        joueur = "O" if joueur == "X" else "X"  #
on change de joueur


if __name__ == "__main__":
    morpion()
```

## 6. Le jeu du pendu

Dans les idées de projets Python d'entrée de gamme, le Pendu est l'un des jeux les plus populaires. Un mot est choisi soit par le joueur adverse, soit par le programme. Et le joueur dispose de tout l'alphabet pour deviner les lettres.

Le but pour le joueur est de devenir le mot en choisissant les lettres bonnes lettre. Si la lettre est correcte, le mot est complété. Si la lettre choisie est incorrecte, vous perdez une vie et le pendu apparaît davantage.

La partie se termine soit par une victoire si le mot entier est trouvé, soit par une défaite sir le bonhomme du pendu apparaît en intégralité (vous n'avez plus de vie). Traditionnellement, six erreurs sont autorisées avant que le joueur ne perde le jeu, mais ce nombre peut être modifié en fonction de la manière dont vous souhaitez créer votre itération du jeu.

## 7. Développer un Bot en Python

Pour apprendre python en réalisant des projets simple, vous pouvez aussi vous intéresser au scripting et à l'automatisation avec des bots. Vous pouvez facilement faire un bot qui execute des tâches très simples en quelques dizaines de lignes de code. Par exemple, j'ai fait un Bot Discord en Python avec la librairie `discord.py` sur lequel vous pouvez vous baser pour construire un bot Discord en Python.

## 8. Visualiser de la data avec Python

Python excelle dans le monde de la data science, il sera dommage de ne pas explorer cette partie qu'offre ce language de programmation. Je vous propose donc de visualiser de la data avec la librairie matplotlib.

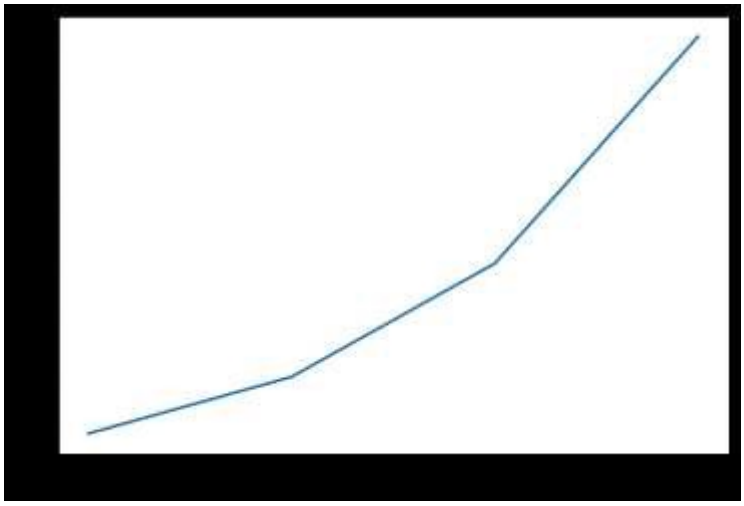
```
# On importe le module pour visualiser la data
from matplotlib import pyplot as plt

# Les valeurs pour l'axe X
x = [1, 2, 3, 4]

# Les valeurs pour l'axe Y
y = [2, 4, 8, 16]

# Fonction pour créer le graph
plt.plot(x, y)

# Function pour montrer le graph
plt.show()
```



💡 Pour cet exemple, j'utilise `jupyter` (anciennement iPython) qui permet d'écrire du Python de manière interactive dans votre navigateur. Jupyter est totalement gratuit et permet d'apprendre Python en ayant des résultats visuels directement, je recommande ce package à tous les débutants. Vous pouvez tester Jupyter gratuitement en suivant ce notebook.

Amusez vous à complexifier l'affichage de ce `plot`, importez des données, changez de type de chart (vous trouverez des

exemples sur le site de matplotlib) et partagez vos créations en section commentaires !

## 9. Générer des images avec Python et Pillow

Vous voulez automatiser la création de contenu pour votre Instagram, Pinterest, votre site ou projet perso ? Alors **Pillow** vous fera gagner un temps fou.

Comme dernier projet de cette liste d'idées de projets en Python pour débutants, je vous propose d'explorer le package Pillow qui vous permet de créer des images automatiquement. Avec **Pillow** ou **Python Imaging Library (PIL)** vous pourrez en effet définir une taille d'image puis y placer des couleurs, du texte et d'autres images.

Par exemple, pour apprendre une langue étrangère, vous pouvez créer une liste avec le mot en français, le mot dans la langue que vous voulez apprendre et l'image associée au mot. Ensuite, créez votre image avec Pillow, ajoutez ces 3 éléments et voilà, vous avez des flashcards pour apprendre plus de vocabulaire dans une nouvelle langue !

## Plus d'idées de projets en Python

J'espère vous avoir inspiré à créer un ou plusieurs projets en Python.

Vous devriez maintenant être capable de réaliser des petits jeux avec ou sans interfaces graphiques, d'utiliser des bibliothèques python et de comprendre les structures de contrôle ainsi que la complexité algorithmique.

Comme vous l'avez vu, Python est très versatile et permet de toucher à un peu tous les domaines du développement informatique.


Si vous cherchez plus d'idées de projets pour débutants à réaliser en Python, je vous conseille

La formation Python par la pratique : 101 Exercices Corrigés.

Ou le livre en anglais Learn Python 3 the Hard Way qui vous proposera une cinquantaine d'exercices !

**Les lecteurs de cet article ont aussi aimé :**

[11 idées de projets pour progresser en JavaScript]

[Top 10 des meilleurs livres pour apprendre Python en 2022]

[Top 7 des meilleurs livres Django en 2022]

[8 idées de projets pour progresser en PHP]

[5 idées de projets pour progresser en Java]