

# PROJET : Dessine ta rue !

Dans ce projet, on dessine. Plutôt on fait dessiner Python. Exemple :



Et si vous êtes ambitieux :

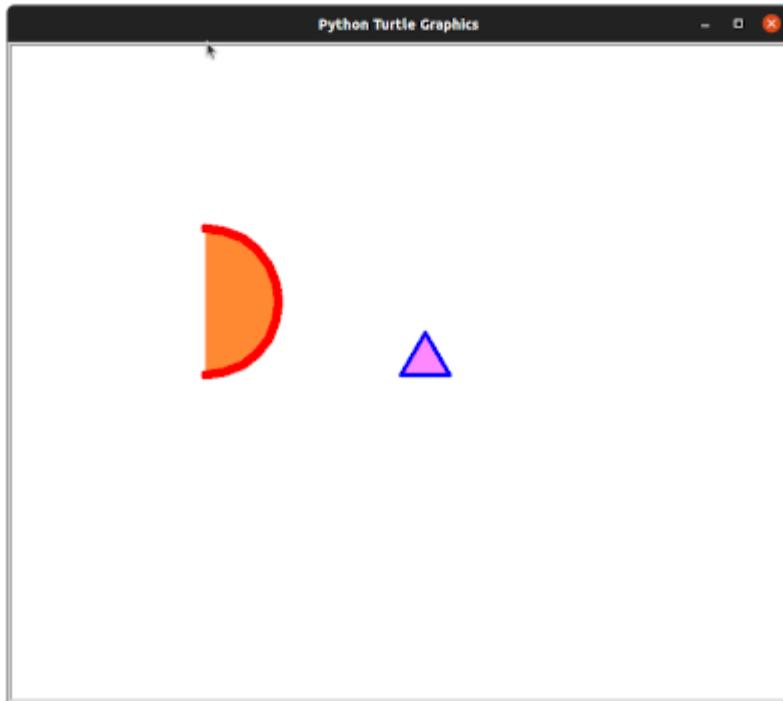


Et si vous voulez un peu plus de verdure par exemple :



## 1 - Idée générale du projet :

- On aura d'abord un premier programme **formes.py** comportant des fonctions capables de dessiner des formes simples. Elles utiliseront le module **Turtle** à l'interne.



- Un deuxième programme nommé **rue.py** devra être capable de dessiner une rue comportant plusieurs immeubles. Aucune ligne de Turtle à l'intérieur de ce programme, juste des appels aux fonctions du premier programme, **formes.py**.



- Votre troisième programme **interpreteur.py** devra lire un fichier-texte contenant une description codifiée de la rue et devra la tracer en utilisant... le programme 2, **rue.py**.

Imaginons qu'on ai un fichier **exemple1.txt** texte qui contienne ceci :

```
rue
4 immeubles
facade rouge - 2 étages - porte rouge au milieu - toit classique
facade verte - 4 étages - porte verte à droite - toit plat
aléatoire
facade violet - 3 étages - porte bleue au milieu - toit bizarre
```

On pourrait alors imaginer cette fonction présente dans **interpreteur.py** :

```
>>> interpreter_dessin('exemple1.txt')
```



## 2 - Cahier des charges

---

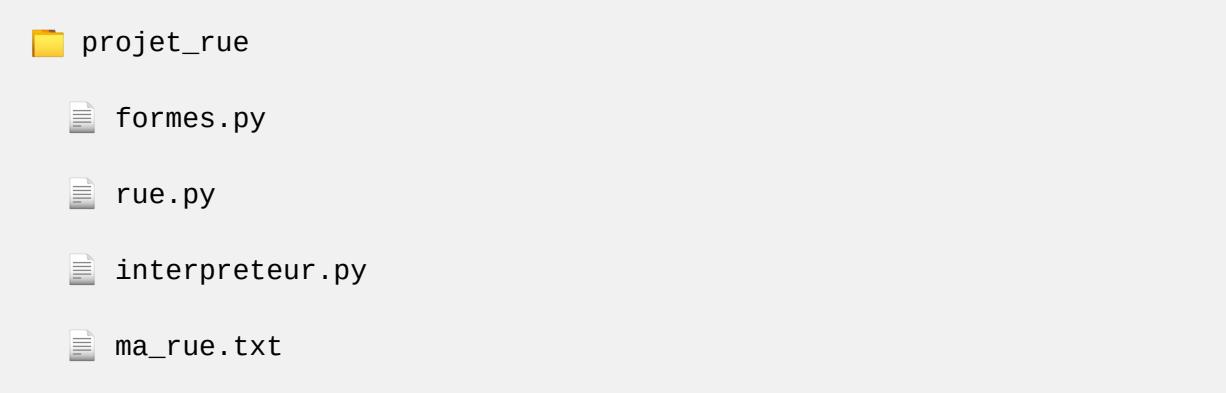
Pour être certain de partir tous dans la même direction au début, et gagner un peu de temps de coordination, voici quelques choix **imposés** :

Chaque immeuble aura les caractéristiques suivantes :

- Des immeubles d'une largeur de 140 pixels,
- Chaque étage fait 80 pixels,
- Entre 1 à 5 étages pour un immeuble
- Chaque fenêtre fait 30 px sur 30 px
- Une porte unique au rez-de-chaussée de 30 px sur 50px
- Deux variations du toit
- Plusieurs couleurs disponibles
- Il est possible que certaines fenêtres soient des portes-fenêtres de 30 px sur 50 px.

On devra pouvoir créer la rue de façon aléatoire. Elle devra alors comporter au minimum 4 immeubles différents.

Votre projet comportera ces 4 fichiers, placés simplement dans le même répertoire.



## 3 - Séquence 1 : **formes.py**

**formes.py** doit contenir des fonctions permettant de dessiner des formes géométriques basiques : carré, rectangle, disque, cercle, triangle...

Ouvrir ce fichier et en s'inspirant des éléments déjà présents, ajouter les fonctions suivantes :

- rectangle(largeur:int, hauteur:int, infos:dict, coordonnees:tuple)
- triangle\_isocèle(cote:int, hauteur:int, infos:dict, coordonnees:tuple)
- cercle(rayon:int, infos:dict, coordonnees:tuple)

Le dictionnaire *infos* contient des éléments de ce type :

```
infos = {'écriture':'blue', 'fond': '#FF8800', 'épaisseur':5}
```

## 4 - Séquence 2 : **rue.py**

**rue.py** doit contenir les fonctions permettant de dessiner des immeubles. Beaucoup de choses à rajouter dans celui-ci. Le détail des fonctions est proposée dans la partie documentation.

Pour l'instant, il trace notamment 4 "immeubles" si le fichier *formes.py* est finalisé.

On utilise encore dans ce fichier un dictionnaire : on pourra envoyer autant de caractéristiques qu'on veut. Il est créé aléatoirement avec la fonction *determiner\_immeuble(numero:int)* pour cette partie.

La prochaine séquence permettra de choisir ces caractéristiques.

Complète les fonctions pour tracer portes, fenêtres et toits.

## 5 - Séquence 3 : *interpreteur.py*

Réaliser le programme **interpreteur.py** qui doit lire un fichier-texte (en suivant une syntaxe qu'il vous faudra inventer au préalable) pour parvenir à retrouver les caractéristiques voulues pour la rue (nombre d'immeubles, caractéristiques de chaque immeuble...). Il ne restera ensuite qu'à faire appel aux fonctions du module **rue.py** pour tracer votre rue.

Voilà un code permettant de récupérer le texte contenu dans un fichier *exemple.txt* pour stocker les lignes dans une liste :

```
rue
facade rouge - 2 étages - porte jaune au milieu - toit plat
facade verte - 3 étages - porte bleue à droite - toit triangulaire
```

```
fichier = open('exemple.txt', 'r', encoding="utf-8")
ligne = fichier.readlines() #ligne est une liste, chaque élément une
                           ligne
lignes = []
for l in ligne:
    print(l)
    x=l.replace('\n', '')
    print(x)
    lignes.append(x) #on s'est débarassé du retour à la ligne \n
print(ligne)
print(lignes)
fichier.close() #ne pas oublier cette ligne pour ne pas corrompre les
                  données
```