



# Maths-SNT : Réalisations de drapeaux en Python

publié le 14/06/2021

## Descriptif :

Le laboratoire de mathématiques du Lycée du Bois d'Amour présente un retour de projet liant les programmes des SNT et de mathématiques autour de l'algorithmique et de la programmation.

## Sommaire :

- Le projet final proposé aux élèves
- Exemples de productions d'élèves
- Objectifs et liens avec les programmes
- Outils et moyens à disposition
- Mise en place et déroulement
- Exemples de fiches d'activités et de codes donnés aux élèves
- Impressions des enseignants
- Prolongements, questionnements et améliorations possibles
- Autres exemples de productions d'élèves

### ● Le projet final proposé aux élèves

Réaliser en Python des drapeaux (réels ou imaginaires).

### ● Exemples de productions d'élèves

```
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 test= Image.new("RGB", (600,400), (255,128,0))
5
6 n=(0,0,0)
7 b=(255,255,255)
8 r=(255,0,0)
9 o=(255,128,0)
10
11 for col in range(200,400):
12     for ligne in range(100,300):
13         if ligne >100 and ligne <300:
14             test.putpixel((col,ligne),b)
15         if col >200 and col<400:
16             test.putpixel((col,ligne),b)
17
18 for col in range(270,330):
19     for ligne in range(170,230):
20         if ligne >170 and ligne <230:
21             test.putpixel((col,ligne),n)
22         if col >270 and col<330:
23             test.putpixel((col,ligne),n)
24
25 for col in range(200,520):
26     for ligne in range(180,220):
27         if ligne >185 and ligne <205:
28             test.putpixel((col,ligne),r)
29         if col >295 and col<305:
30             test.putpixel((col,ligne),r)
```

Exemple de drapeau imaginaire

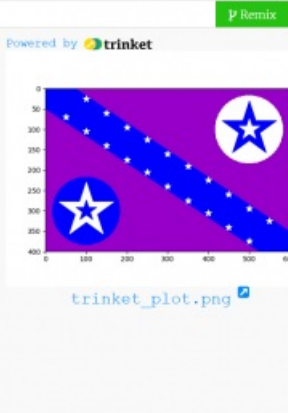
```
74     if (col-71)**2 + (ligne-109)**2 < 62:
75         dessiné.putpixel((col, ligne), blanc)
76     if (col-87)**2 + (ligne-135)**2 < 62:
77         dessiné.putpixel((col, ligne), blanc)
78
79 #yin yang
80 #on parcourt tous les pixels de l'image:
81 for col in range(0, 100):
82     for ligne in range(0, 120):
83         if (col-90)**2 + (ligne-60)**2 < 400:
84             dessiné.putpixel((col, ligne), rouge)
85         if (col-99)**2 + (ligne-64)**2 < 128:
86             dessiné.putpixel((col, ligne), bleu)
87         if (col-98)**2 + (ligne-65)**2 < 128:
88             dessiné.putpixel((col, ligne), bleu)
89         if (col-92)**2 + (ligne-60)**2 < 128:
90             dessiné.putpixel((col, ligne), bleu)
91         if (col-83)**2 + (ligne-65)**2 < 150:
92             dessiné.putpixel((col, ligne), bleu)
93         if (col-85)**2 + (ligne-60)**2 < 100:
94             dessiné.putpixel((col, ligne), bleu)
95         if (col-86)**2 + (ligne-70)**2 < 100:
96             dessiné.putpixel((col, ligne), bleu)
97         if (col-96)**2 + (ligne-68)**2 < 100:
98             dessiné.putpixel((col, ligne), bleu)
99         if (col-81)**2 + (ligne-60)**2 < 120:
100             dessiné.putpixel((col, ligne), rouge)
101 plt.imshow(dessiné)
102 plt.show()
```

Drapeau de la Corée du Sud

```

main.py
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 dessin = Image.new("RGB", (600,400), (150,0,200))
5
6
7 blanc = (255, 255, 255)
8 rouge = (255,0,0)
9 violet = (150, 0, 200)
10 bleu = (0,0,255)
11 black = (0,0,0)
12
13 for col in range(0, 600):
14     for ligne in range(0, 400):
15         if ligne >= (2/3)*col - 50 and ligne <= (2/3)*col + 50:
16             dessin.putpixel((col, ligne), bleu)
17
18
19 image=Image.new("RGB", (100,100), (0,0,128))
20 plt.scatter(100, 100, s = 7000, c = 'white', marker = '*')
21
22
23 plt.scatter(100, 100, s = 1000, c = 'blue', marker = '*')
24
25 plt.scatter(500, 100, s = 7000, c = 'blue', marker = '*')
26
27 plt.scatter(500, 100, s = 1000, c = 'white', marker = '*')
28
29 plt.scatter(50, 70, s = 100, c = 'white', marker = '*')

```

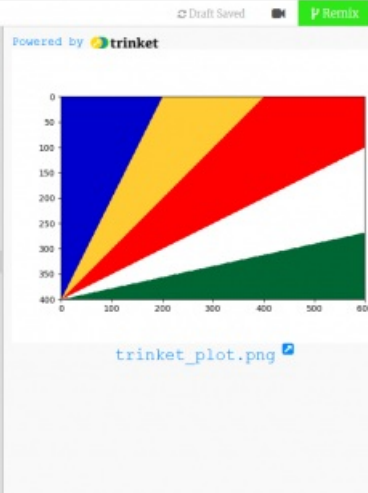


Drapeau imaginaire avec des étoiles

```

main.py
1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 #Creation d'une image blanche RGB 10x10
5 dessin = Image.new("RGB", (600,400), (255,255,255))
6
7 #Definitions des couleurs
8 rouge = (255, 0, 0)
9 jaune = (255, 204, 51)
10 bleu = (0, 0, 204)
11 vert = (0, 102, 51)
12 blanc = (255, 255, 255)
13
14 #On parcourt tous les pixels de l'image
15 for col in range(0, 600):
16     for ligne in range(0, 400):
17         if ligne <= -2*col + 400:
18             dessin.putpixel((col, ligne), bleu)
19         if ligne > -2*col + 400 and ligne <= -1*col + 400:
20             dessin.putpixel((col, ligne), jaune)
21         if ligne > -1*col + 400 and ligne <= -0.5*col + 400:
22             dessin.putpixel((col, ligne), rouge)
23         if ligne > -0.22*col + 400:
24             dessin.putpixel((col, ligne), vert)
25
26
27 plt.imshow(dessin)
28 plt.show()
29

```



Drapeau des Seychelles

Un code Python correspondant (à tester sur <https://trinket.io/python3> par exemple) :

```

import matplotlib.pyplot as plt
from PIL import Image

#Creation d'une image blanche RGB 10x10
dessin = Image.new("RGB", (600,400), (255,255,255))

#Definitions des couleurs
rouge = (255, 0, 0)
jaune = (255, 204, 51)
bleu = (0, 0, 204)
vert = (0, 102, 51)
blanc = (255, 255, 255)

#On parcourt tous les pixels de l'image
for col in range(0, 600):
    for ligne in range(0, 400):
        if ligne <= -2*col + 400:
            dessin.putpixel((col, ligne), bleu)
        if ligne > -2*col + 400 and ligne <= -1*col + 400:
            dessin.putpixel((col, ligne), jaune)
        if ligne > -1*col + 400 and ligne <= -0.5*col + 400:
            dessin.putpixel((col, ligne), rouge)
        if ligne > -0.22*col + 400:
            dessin.putpixel((col, ligne), vert)

```

```
plt.imshow(dessin)
plt.show()
```

## ● Objectifs et liens avec les programmes

Parcourir les notions d'algorithmique et de programmation présentes dans les programmes des SNT et de mathématiques pour accompagner le thème : La photographie numérique.

### ○ Extraits des programmes de Mathématiques :

#### Contenus :

- Séquences d'instructions
- Instruction conditionnelle
- Boucle bornée (for)
- Pente (ou coefficient directeur) d'une droite non parallèle à l'axe des ordonnées.

#### Capacités attendues :

- Concevoir et écrire une instruction d'affectation, une séquence d'instructions, une instruction conditionnelle.
- Programmer, dans des cas simples, une boucle bornée, [...].
- Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme.
- Déterminer une équation de droite à partir de deux points, [...].

### ○ Extraits des programmes SNT :

#### Contenus :

- Instructions conditionnelles
- Boucles bornées
- Pixels

#### Capacités attendues :

- Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.

## ● Outils et moyens à disposition

### ○ Conditions d'enseignements au lycée BDA pour l'année 2020-2021 :

Au lycée BDA, dans la plupart des classes de seconde, le/la professeur(e) de mathématiques est aussi en charge de l'enseignement des SNT pour la classe ce qui permet de faciliter le lien entre les différents programmes et d'ajuster certaines heures de cours selon les besoins. Le lycée ne disposant pas vraiment de salles informatiques permettant un travail simultané de 18 binômes ou de salles permettant de couper la classe en deux dans de bonnes conditions, les heures de dédoublements en Mathématiques et en SNT facilitent grandement l'accompagnement de l'enseignant(e) lors de phases d'apprentissages de nouvelles notions de programmation parfois complexes.

#### Outils :

L'éditeur Pyzo est installé sur l'ensemble des postes du lycée mais nous avons surtout utilisé une émulation de Python en ligne : Trinket qui fonctionne entièrement dans le navigateur. Cela permet aux élèves de travailler chez eux sans rien installer.

Dans sa version de base, Trinket (Python2), offre une émulation de la librairie Turtle.

Afin de permettre aux élèves de compléter leurs productions à la maison, Trinket/python3 offre des alternatives pour l'utilisation en ligne de la librairie PIL (Python Imaging Library).

Pour le retour des productions, les élèves envoient par mail à leurs professeurs le lien Trinket correspondant.

**Attention** : Les codes Trinket/python3 peuvent être modifiés par une personne disposant du lien. Il faudra donc veiller à ne pas effacer le code d'un élève. Aussi, il sera préférable de diffuser les codes d'exemples sans lien Trinket.

### ● Mise en place et déroulement

Le contexte sanitaire (2020-2021) ainsi que la difficulté de délimiter clairement le début et la fin du projet ne permettent pas de donner un nombre précis de semaines. Cela dépend aussi de la façon dont l'enseignant(e) aborde les différentes notions (débats ou simples curiosités, programmes à modifier ou à construire entièrement, contraintes faibles ou fortes dans le projet final...)

#### ○ Dessiner sous contraintes :

Dans un premier temps, les élèves découvrent (redécouvrent) la notion d'algorithme par un retour débranché aux instructions élémentaires et la nécessité de s'entendre sur des normes de communication pour être compris par l'humain et aussi par la machine.

La programmation en langage Python est alors introduite avec ses contraintes et la syntaxe des boucles bornées est installée pour reproduire certains dessins.

Trinket (python2) est alors utilisé pour le passage sur machine avec l'utilisation de la librairie Turtle.

#### ○ Pixels et images numériques :

Le thème : La photographie numérique permet d'aborder la notion d'image numérique matricielle, les pixels et le codage RVB. Là aussi des activités branchées/débranchées de production/reproduction d'images simples (pixel par pixel) sont mises en place.

Un problème arrive rapidement : Comment créer, reproduire ou transformer des images dont la dimension est bien plus élevée ?

#### ○ Boucles bornées imbriquées :

Pour des images de taille supérieure, la nécessité d'utiliser des boucles se fait rapidement sentir. Les élèves jouent sur les bornes des boucles pour dessiner des rectangles. De nombreux drapeaux sont alors à reproduire. Hélas, certains drapeaux (comme l'Union Jack) restent encore inaccessibles...

#### ○ Boucles bornées imbriquées et les instructions conditionnelles :

Une autre façon de dessiner des rectangles est alors découverte : Parcourir tous les pixels de l'image et n'agir que sur les pixels dont les coordonnées vérifient une ou plusieurs conditions.

#### ○ Liens avec les fonctions affines :

Un travail en parallèle sur les fonctions affines permet alors de proposer d'autres types de conditions pour agir sur des ensembles de pixels dont les frontières ne sont pas forcément horizontales ou verticales. Calculs d'équations réduites de droites non verticales passant par deux points de coordonnées et débats autour des notions croissante/décroissante (en effet, le sens de l'axe des ordonnées dans les images numériques est inversé).

#### ○ Demandes de productions :

Après avoir travaillé ces différentes approches, les élèves ont pour mission de produire/reproduire un drapeau. D'autres contraintes apparaissent pour certains projets (cercles, étoiles). Ce travail de production finale s'est principalement effectué à la maison. Un retour par mail a été demandé et certains élèves ont su trouver, voire même construire entièrement des façons de dessiner des étoiles à cinq branches.

#### ○ Traitements des images numériques :

Des discussions et/ou activités autour du traitement d'images numériques (filtres, niveaux de gris, passage au négatif, suppression d'un fond vert...) permettent de réinvestir les notions parcourues. La méthode `getpixel()` est


alors découverte.

## ● Exemples de fiches d'activités et de codes donnés aux élèves

### Images numériques et couleurs :

 [Activité 2 : Découverte des couleurs RVB](#) (PDF de 420 ko)  
Maths-SNT : Réalisations de drapeaux en Python - Académie de Poitiers.

### Dessiner des formes avec des conditions :

 [Un exemple de TP : TP boucles et conditions](#) (PDF de 170.4 ko)  
Maths-SNT : Réalisations de drapeaux en Python - Académie de Poitiers.

D'autres exemples sur :

► [Les images numériques - Tableaux Maths](#) 

## ● Impressions des enseignants

De nombreux élèves, qu'ils aient ou non des difficultés ou des manques d'appétence pour le numérique et la programmation, se sont engagés dans les projets et se sont sentis valorisés par la réussite de leurs productions. Les œuvres retournées montrent, dans l'ensemble, un bel investissement et certaines productions révèlent une recherche très poussée de créations d'éléments non travaillés en classe (étoiles, portions de disques, polygones et stratégies d'assemblages complexes...)

## ● Prolongements, questionnements et améliorations possibles

Dans certaines classes, des motifs circulaires ont été dessinés. Il est donc possible de réinvestir dans ce cadre un travail sur la distance entre deux points dans un r.o.n.

Afin de lier davantage le projet avec le traitement des images numériques, on pourrait demander aux élèves de construire des œuvres à partir de photographies numériques.

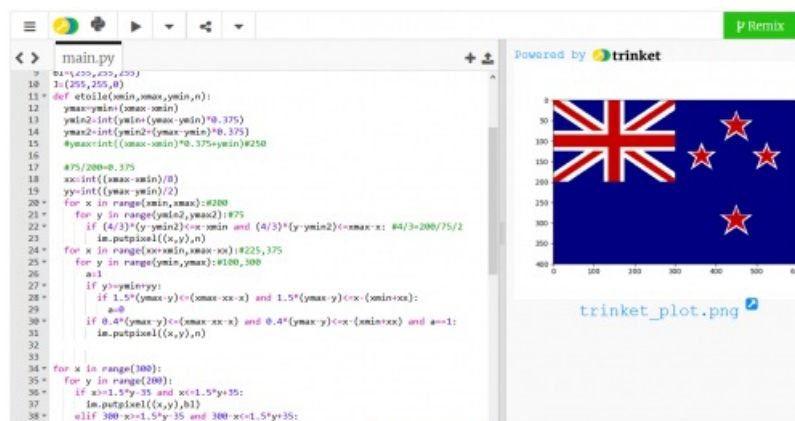
Par exemple, construire une image à partir de plusieurs images par extraction de formes.

Un travail sur les dégradés de couleurs peut permettre de travailler davantage les variables informatiques.

Nous sommes partis sur des noms de compteurs de boucles « col » et « ligne » pensant qu'il serait plus simple pour les élèves de s'y retrouver dans la construction de leurs premières images. Les noms de compteurs « x » et « y » semblent plus pertinents sur l'ensemble du projet.

## ● Autres exemples de productions d'élèves

### Création par un élève d'une procédure permettant de réaliser une étoile à cinq branches :



```
main.py
8 n=1000;e=25;e99
9 i=(25;25;0)
10 def etoile(smin,smax,ymn,n):
11     ymax=ymn+(smax-smin)
12     ymn2=int(ymn+(smax-ymn)*0.375)
13     ymn2=int(ymn2+(smax-ymn)*0.375)
14     #smax=int((smax-smin)*0.375+ymn)#250
15
16
17     #75/200=0.375
18     xx=int((smax-smin)/0)
19     yy=int((ymax-ymn)/2)
20
21     for x in range(smin,smax):#200
22         for y in range(ymn2,ymax):#75
23             if (4/3)*(y-ymn2)<=smin and (4/3)*(y-ymn2)<=smax-x: #4/3=200/75/2
24                 im.putpixel((x,y),n)
25
26         for x in range(sx=smin,smax-xx):#225;375
27             for y in range(ymn,ymax):#100;300
28                 a=1
29                 if y==ymn+yy:
30                     if 1.5*(ymax-y)<=(smax-xx-x) and 1.5*(ymax-y)<=(smin+xx):
31                         a=0
32                     if 0.4*(ymax-y)<=(smax-xx-x) and 0.4*(ymax-y)<=(smin+xx) and a==1:
33                         im.putpixel((x,y),n)
34
35
36 for x in range(100):
37     for y in range(200):
38         if x<=1.5*y-35 and x<=1.5*y+35:
39             im.putpixel((x,y),b1)
40         elif 300-x<=1.5*y-35 and 300-x<=1.5*y+35:
```

Un élève créé lui-même son code pour dessiner une étoile à cinq branches.

### Ensemble des productions d'une classe :

 [Ensemble de productions d'une classe : Drapeaux de la classe de 2A](#) (PDF de 285.5 ko)

## Exemples de productions d'une autre classe :

 [Exemples de productions dans une autre classe](#) (PDF de 330.8 ko)

Maths-SNT : Réalisations de drapeaux en Python - Académie de Poitiers.

## Deux productions de fiches complètes autour de drapeaux :

 [Fiche de présentation d'un drapeau : Ambre propose une fiche complète autour d'un drapeau](#) (PDF de 70 ko)

Maths-SNT : Réalisations de drapeaux en Python - Académie de Poitiers.

 [Fiche complète Zia : Zia propose une fiche complète sur un drapeau avec des calculs de coefficients directeurs](#)  
(PDF de 198.7 ko)

Maths-SNT : Réalisations de drapeaux en Python - Académie de Poitiers.



**Académie  
de Poitiers**

**Avertissement : ce document est la reprise au format pdf d'un article proposé sur l'espace pédagogique de l'académie de Poitiers.**

**Il ne peut en aucun cas être proposé au téléchargement ou à la consultation depuis un autre site.**