



# Dessiner avec Python ou Scratch

## Qu'est-ce qui change ?



### Fiche PROF

**Niveau concerné** : Seconde

**Durée** : Séance de 1h en classe et prolongement possible à la maison

**Type de travail** : Dessiner une figure en Python

**Thèmes du programme** : Algorithmique et notion de fonction

### Introduction

Les élèves ont normalement utilisé Scratch au collège.

Il semble nécessaire, au début, de partir d'un logiciel familier afin de mettre les élèves en confiance.

Pourquoi une activité avec des figures à dessiner ?

On va voir que la programmation sur Scratch et Python sont très proches et faciles à comparer dans ce cas.

On évitera, dans les premières séances de déstabiliser les élèves ; l'interface Python étant moins ludique a priori.

### Scénario

Dans un premier temps, on demandera aux élèves de réaliser une figure sur Scratch.

*On peut imaginer que ce travail soit à réaliser à la maison afin d'avoir plus de temps pour travailler ce qui nous intéresse : la familiarisation avec l'interface Python.*

Cette figure pourra être optimisée en utilisant des boucles et des « blocs ».

Ensuite, on donnera une version en Python, utilisant un module « tortue ».

On essaiera de comparer les deux « scripts » afin de comprendre quelques éléments fondamentaux dans la programmation en Python : import de modules, syntaxe, indentation ...

On fera recopier le script donné en utilisant un environnement du type : **EduPython**

On expliquera comment on exécute le script, comment on l'enregistre ...

On proposera ensuite aux élèves un petit lexique avec d'autres commandes du module « tortue » qui leur permettra d'expérimenter et de complexifier leur dessin initial ; voir en inventer d'autres ...

### Objectifs

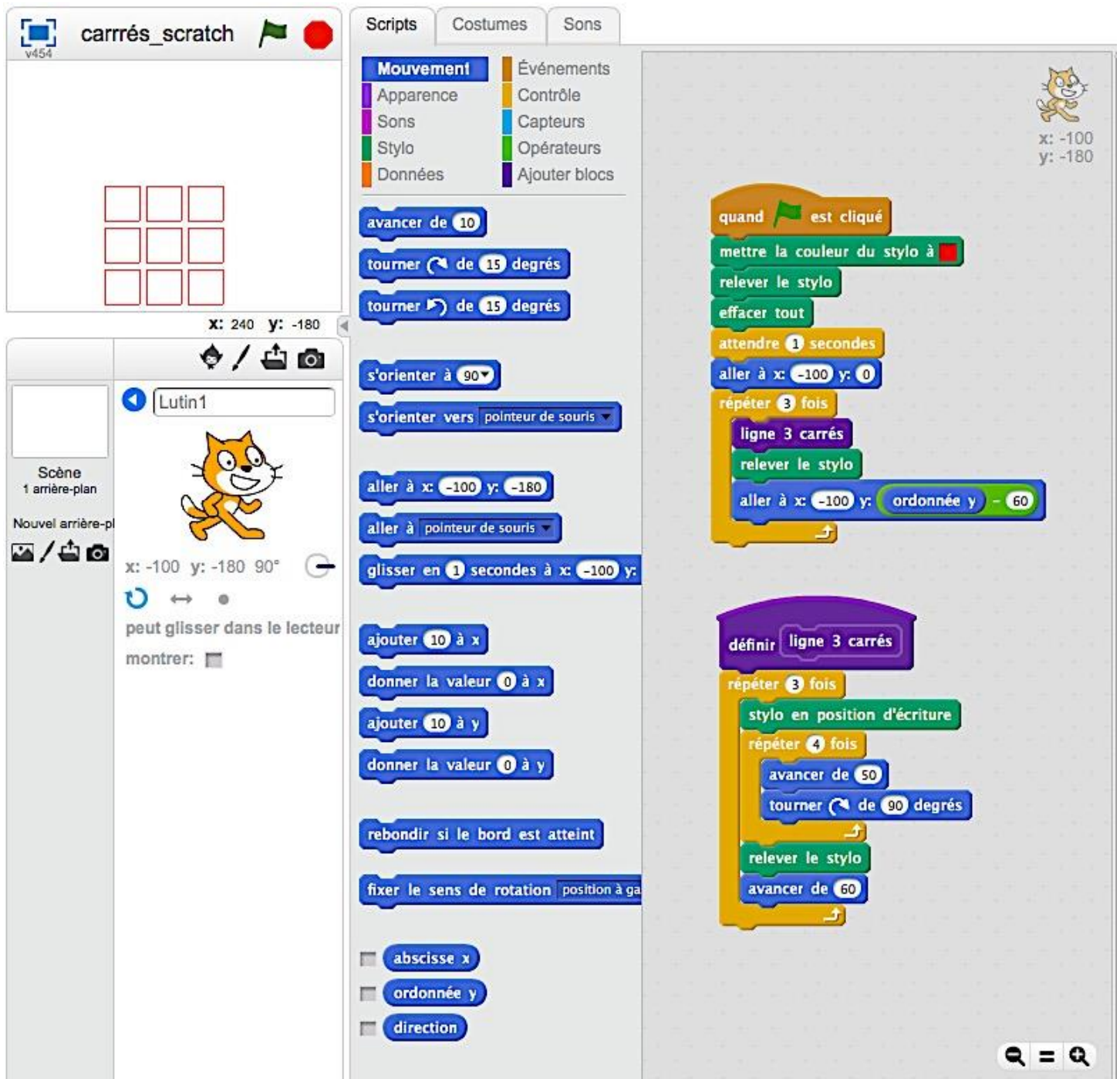
- Retravailler les instructions de base d'algorithmique vues au collège
- Introduire la notion de fonction en informatique

## Exemples de figures à proposer

Écrire un script qui permet de tracer un carré

Compléter ce script pour qu'il exécute une ligne de 3 carrés identiques espacés également

Compléter le script précédent pour obtenir la figure suivante :



The image shows the Scratch IDE interface. On the left, a stage window displays a 3x3 grid of squares. The stage coordinates are x: 240 y: -180. Below the stage, the sprite 'Lutin1' is shown with coordinates x: -100 y: -180 and a rotation of 90°. The script area is divided into three sections: 'Mouvement', 'Événements', and 'Contrôle'. The 'Mouvement' section contains blocks for 'avancer de 10', 'tourner de 15 degrés' (twice), 's'orienter à 90', 's'orienter vers pointeur de souris', 'aller à x: -100 y: -180', 'aller à pointeur de souris', 'glisser en 1 secondes à x: -100 y:', 'ajouter 10 à x', 'donner la valeur 0 à x', 'ajouter 10 à y', 'donner la valeur 0 à y', 'rebondir si le bord est atteint', and 'fixer le sens de rotation position à ga'. The 'Événements' section contains 'quand est cliqué', 'mettre la couleur du stylo à', 'relever le stylo', 'effacer tout', and 'attendre 1 secondes'. The 'Contrôle' section contains 'aller à x: -100 y: 0', 'répéter 3 fois' (containing 'ligne 3 carrés', 'relever le stylo', and 'aller à x: -100 y: ordonnée y - 60'), and 'définir ligne 3 carrés' (containing 'répéter 3 fois' with 'stylo en position d'écriture', 'répéter 4 fois' with 'avancer de 50' and 'tourner de 90 degrés', 'relever le stylo', and 'avancer de 60').

### Remarque

On pourra faire remarquer aux élèves, s'ils ne le savent pas ou qu'ils ne s'en souviennent plus, que l'on peut cacher le lutin pour ne pas interférer avec la figure.

Le script en Python :

```
from turtle import *
def carre () :
    color ("red")
    begin_fill()
    for i in range (4) :
        down()
        forward (50)
        left(90)
    end_fill()
```

```
def ligne():
    for i in range (3):
        carre()
        up()
        forward (60)
```

```
x=0
y=0
for i in range (3) :
    goto(x,y)
    ligne()
    x=0
    y=y-60
```

- Instruction d'importation à identifier
- Affectation : utilisation du signe « = »
- Identification du début et de la fin d'un bloc d'instructions
  - Début par un « : » et un retour ligne
  - Le bloc entier est décalé vers la droite (indentation) de la même façon
- Def « nom » () : permet de définir une fonction

Comparaison avec Scratch : Bloc

Remarque : ()

On souhaite par exemple utiliser la « fonction » **carre()** et pouvoir modifier la longueur du côté, on procédera comme suit :

```
def carre (c) :
    color ("red")
    begin_fill()
    for i in range (4) :
        down()
        forward (c)
        left(90)
    end_fill()
```

Si on veut dessiner ensuite un carré de côté 100, on appelle la fonction : **carre(100)**

Une fonction peut contenir plusieurs attributs à l'intérieur parenthèses, séparés par une virgule.

On demandera aux élèves de faire les correspondances du type :

Instruction scratch	Instruction Python
avancer de 50	forward (50)
relever le stylo	up()
aller à x : y :	goto(x,y)
répéter 4 fois	for i in range (4) :

Relever une instruction qui n'existe pas sur Scratch (pour colorier une forme) :

```
begin_fill()
.....
end_fill()
```

## Fiche ÉLÈVE

Niveau concerné : Seconde

Durée : Séance de 1h en classe et prolongement possible à la maison

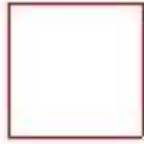
Type de travail : Dessiner une figure en Python

Thèmes du programme : Algorithmique et notion de fonction

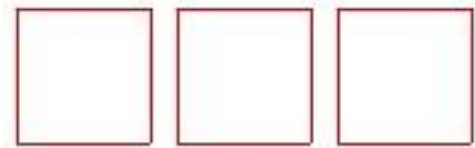
### 1<sup>ère</sup> Partie

En utilisant le logiciel Scratch

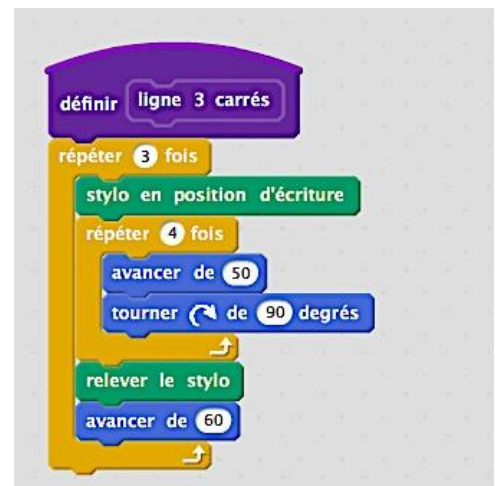
Écrire un script qui dessine un carré :



Compléter le script précédent pour obtenir la figure suivante :



Lire le script ci-contre et préciser ce qu'il fait :



### 2<sup>ème</sup> Partie

En utilisant le logiciel Python

1. Ouvrir le logiciel EduPython, écrire le script ci-contre et l'exécuter. On vérifiera que la figure réalisée est similaire à celle proposée par le script précédent.

```
from turtle import *
def carre () :
    color ("red")
    begin_fill()
    for i in range (4) :
        down()
        forward (50)
        left(90)
    end_fill()
```

```
def ligne():
    for i in range (3):
        carre()
        up()
        forward (60)
```

```
x=0
y=0
for i in range (3) :
    goto(x,y)
    ligne()
    x=0
    y=y-60
```

BIEN RESPECTER LA SYNTAXE ET LA MISE EN FORME (retours à la ligne, retraits ....)

2. Compléter le tableau suivant en identifiant l'instruction Python correspondante dans chacun des cas :

Instruction scratch	Instruction Python
avancer de 50	
relever le stylo	
aller à x: y:	
répéter 4 fois	
tourner dans le sens des aiguilles d'une montre de 90°	
Stylo en position d'écriture	
mettre la couleur du stylo à	

3. Observons la fonction `carre()` ci-contre :

a. Modifions la fonction de la façon suivante :

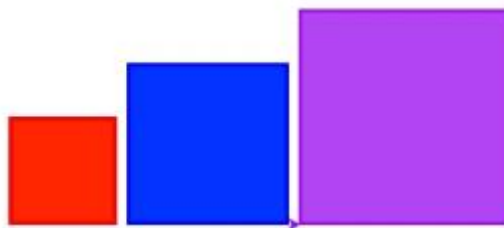
```
def carre (c,couleur) :  
    color (couleur)  
    begin_fill()  
    for i in range (4) :  
        down()  
        forward (c)  
        left(90)  
    end_fill()
```

```
def carre () :  
    color ("red")  
    begin_fill()  
    for i in range (4) :  
        down()  
        forward (50)  
        left(90)  
    end_fill()
```

Que se passe-t-il si on exécute la fonction `carre(100, « green »)` ?

.....

b. Écrire un script en langage Python pour obtenir la figure suivante :



### 3<sup>ème</sup> Partie

*Pour aller plus loin*

Maintenant à vous d'imaginer une figure !  
Simple ou complexe, qu'elle soit jolie !

Voici quelques fonctions supplémentaires à aller chercher sur internet ...

[https://fr.wikibooks.org/wiki/Programmation\\_Python/Turtle](https://fr.wikibooks.org/wiki/Programmation_Python/Turtle)

<https://docs.python.org/3.0/library/turtle.html>

