

## Découverte de la structure:

# Si alors sinon

### 1- Test du programme 1 :

Ouvrir le fichier Si\_Ex1.py du dossier Si\_Alors\_Sinon qui contient le code ci-contre.

Exécuter le plusieurs fois.

Vérifier dans quels cas les parties rouge et verte sont exécutées. (cf affichage de la valeur i)

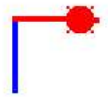
L'instruction `random.random()` donne un nombre aléatoire de  $[0;1[$  .

Expliquer le fonctionnement du programme.

```
from gturtle import *
import random

joe=Turtle()
i=random.random()
joe.forward(50)
if (i<0.5) :
    joe.setColor('red')
    joe.setPenColor('red')
    joe.right(90)
else :
    joe.setColor('green')
    joe.setPenColor('green')
    joe.left(90)

joe.forward(45)
```



### 2- Test du programme 2 :

Ouvrir le fichier Si\_Ex2.py du dossier Si\_Alors\_Sinon qui contient le code ci-contre.

Exécuter le plusieurs fois.

Expliquer le fonctionnement du programme.

```
from gturtle import *
import random

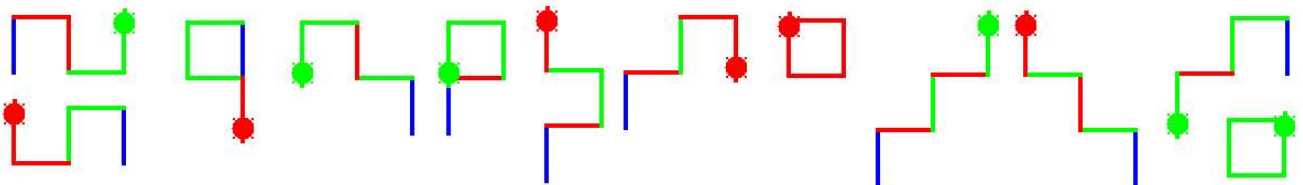
joe=Turtle()

for j in range(4) :
    joe.forward(50)
    i=random.random()

    if (i<0.5) :
        joe.setColor('red')
        joe.setPenColor('red')
        joe.right(90)
    else :
        joe.setColor('green')
        joe.setPenColor('green')
        joe.left(90)

joe.forward(45)
```

Voici quelques exécutions possibles:



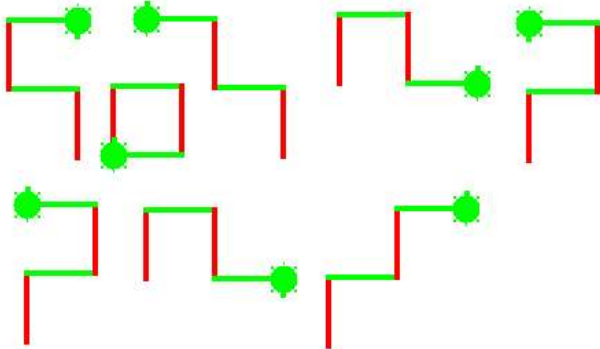
### 3- Test du programme 3 :

Ouvrir le fichier Si\_Ex3.py du dossier Si\_Alors\_Sinon qui contient le code ci-contre.

$i \% n$  vaut 0 si et seulement si  $i$  est un multiple de  $n$ .

Exécuter le plusieurs fois.

Voici quelques exécutions possibles:



```
for i in range(4) :
```

```
if (i%2==0) : # i est pair
```

```
    joe.setColor(Color.red)
```

```
    joe.setPenColor(Color.red)
```

```
else :
```

```
    joe.setColor(Color.green)
```

```
    joe.setPenColor(Color.green)
```

```
r = random.random()
```

```
joe.forward(50)
```

```
if (r<0.5) :
```

```
    joe.right(90)
```

```
else :
```

```
    joe.left(90)
```

Expliquer le fonctionnement du programme.

Le **Si alors** s'écrit sous la forme suivante:

#### Langage naturel

**Si** test est vrai **alors**  
instructions

**Fin si**

#### Langage Python

```
if ( test ) :  
    instructions
```

Le **Si alors sinon** s'écrit sous la forme suivante:

#### Langage naturel

**Si** test est vrai **alors**  
instructions

**Sinon**  
instructions

**Fin si**

#### Langage Python

```
if ( test ) :  
    instructions
```

```
else :  
    instructions
```

Par exemple, on veut qu'un robot range des objets dispersés sur le sol dans des bacs.

- ◆ Les objets sur le sol sont bleus ou verts.

**Si** l'objet est bleu **alors**  
mettre dans le bac bleu

**Sinon**  
mettre dans le bac vert

**Fin si**  
Ici tous les objets sont rangés.

- ◆ Les objets sur le sol sont de couleurs différentes.

**Si** l'objet est bleu **alors**  
mettre dans le bac bleu

**Sinon**  
**Si** l'objet est vert **alors**  
mettre dans le bac vert

**Fin si**  
**Fin si**  
Ici seuls les objets bleus et verts sont rangés.

- ◆ Les objets sur le sol sont de couleurs différentes.

**Si** l'objet est bleu **alors**  
mettre dans le bac bleu

**Fin si**

- ◆ On range les objets bleus et verts dans leurs bacs, les autres sont mis dans le bac blanc.

**Si** l'objet est bleu **alors**  
mettre dans le bac bleu

**Sinon**  
**Si** l'objet est vert **alors**  
mettre dans le bac vert  
**Sinon**  
mettre dans le bac blanc

**Fin si**  
Ici tous les objets sont rangés.

## If imbriqués en langage Python

```
if ( test 1 ) :
    instructions
else :
    if ( test 2 ) :
        instructions
    else :
        if ( test 3 ) :
            instructions
        else :
            instructions
```

<=>

```
if ( test 1 ) :
    instructions
elif ( test 2 ) :
    instructions
elif ( test 3 ) :
    instructions
else :
    instructions
```

équivalent du switch en java

Attention:

```
if ( test 1 ) :
    if ( test 2 ) :
        instructions
    else :
        instructions
else :
    instructions
```

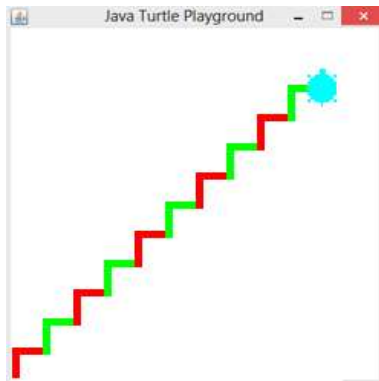
<≠>

```
if ( test 1 ) :
    instructions
elif ( test 2 ) :
    instructions
else :
    instructions
```

## Les défis

### ♦ Avec un si

Défi A01



Déplacement de 30px.

Défi A03

Idem défi A02 mais en alternant les couleurs à chaque tour.

### ♦ Avec deux si imbriqués

Défi A04

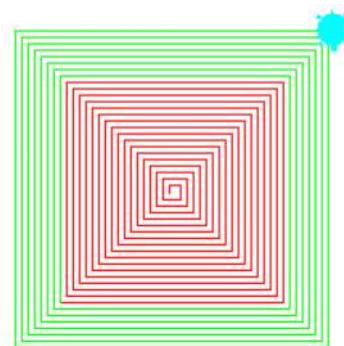
Idem défi A01 mais en alternant le rouge, le vert et le bleu.

Défi A06

Idem défi A05

Valeur de n	n<0,3	0,3≤n≤0,6	0,6<n
Couleur de la marche	rouge	green	Couleur de la marche précédente

Défi A02



Déplacement initial: 4px

Espace entre deux segments: 2px

On change de couleur quand on a effectué 25 tours.

Défi A05

Idem défi A01

On tire à chaque marche un nombre aléatoire n de [0;1[.

Valeur de n	n<0,3	0,3≤n≤0,6	0,6<n
Couleur de la marche	rouge	green	bleu