



Python on Trinket – Leçon 4

Révision

- Boucles
- Variables
- Nombres entiers aléatoires
- Angles de rotation


Discussion

- **Fonction**
 - o Les fonctions sont des catégories de codes réutilisables organisées en blocs qui permettent d'exécuter des actions.
 - o Fonctions définies par l'utilisateur : écrites par le programmeur
 - Évite de réécrire plusieurs lignes de code (pour des tâches que nous voulons exécuter plusieurs fois)
 - Créer un code exécutant une action qui sera ensuite définie par une fonction, et utilise une commande pour intégrer cette fonction à chaque fois que cette action doit s'exécuter
 - La fonction « pass in » permet de paramétrer des variables afin de créer des programmes flexibles
 - Créer des sorties variées en fonctions des valeurs paramétrées par la fonction « pass in »
 - Exemple : créer des flocons de neige de couleurs et de tailles variées, en paramétrant différentes couleurs et tailles avec la fonction « pass in »
 -

Fonctions déterminées par l'utilisateur	Appel à la fonction
<pre>def fonctionname(parameters): #function body return</pre>	<pre>fonctionname(parameters)</pre>

- **Des variables changeantes au fil de temps**
 - o Permet d'avoir plusieurs entrées sans devoir les modifier manuellement

Leçon & pratique

1a. Flocon de neige (fonction déterminée par l'utilisateur)	
<pre>def snowflake(): size = 10 for count in range(size): t.color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255)) t.forward(50) t.stamp() t.backward(20) t.left(45) t.forward(20) t.backward(20) t.right(90) t.forward(20) t.backward(20) t.left(45) t.backward(30) t.left(360/size) return</pre>	
<p>#function call:</p> <pre>snowflake()</pre>	

1b. Formes en utilisant des fonctions déterminées par l'utilisateur avec des paramètres.

```
def snowflake_parameters(length, number_of_rays, colour):
```

```
    for count in range(number_of_rays):
```

```
        t.color(colour)
```

```
        t.forward(length)
```

```
        t.stamp()
```

```
        t.backward(length/3)
```

```
        t.left(45)
```

```
        t.forward(length/3)
```

```
        t.backward(length/3)
```

```
        t.right(90)
```

```
        t.forward(length/3)
```

```
        t.backward(length/3)
```

```
        t.left(45)
```

```
        t.backward(length*(2/3))
```

```
        t.left(360/number_of_rays)
```

```
    return
```

```
t.penup()
```

```
t.goto(0, -100)
```

```
t.pendown()
```

```
for count in range (5):
```

```
    snowflake3(40, 10, "blue")
```

```
    t.penup()
```



<pre> t.goto(random.randint(-100, 100), random.randint(-100, 100)) t.pendown() snowflake3(15, 6, "violet") t.penup() t.goto(random.randint(-100, 100), random.randint(-100, 100)) t.pendown() </pre>	
---	--

2. Variables changeant au fil du temps (Flocons de neige aux couleurs variées)	
<pre> r = 0 g = 21 b = 255 n_snowflakes = 10 for count in range (n_snowflakes): snowflake_parameters(30, 10, (r, g, b)) t.penup() t.goto(random.randint(-150, 150), random.randint(-150, 150)) t.pendown() b = b - 25 r = r + 25 </pre>	