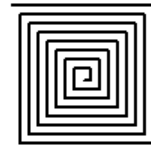


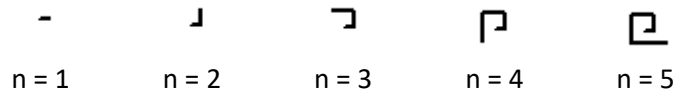
1. Spirale

À l'aide du module turtle, nous allons dessiner une spirale.



Description du tracé :

- Pour une profondeur de 1 : tracé d'un segment de longueur L
- Pour une profondeur de 2 : tracé précédent + rotation de 90° + tracé d'un segment de longueur L + 3
- Pour une profondeur de 3 : tracé précédent + rotation de 90° + tracé d'un segment de longueur L + 3
- ...
- Pour une profondeur de n : tracé précédent + rotation de 90° + tracé d'un segment de longueur L + 3



Algorithme :

```

si n = 1 alors
    tracé du segment de longueur L
sinon
    tracé du segment L
    rotation de 90°
    répéter cet algorithme pour n - 1 et pour une longueur L = L + 3
    
```

✍ Implémenter un programme incluant la fonction récursive `spirale(taille, n)` afin de dessiner la spirale.

- `taille` : longueur du motif pour une profondeur de 1 ;
- `n` : profondeur désirée.

Rappel :

```

import turtle as t1 # import du module turtle
t1.left(90) # rotation de 90°
t1.forward(10) # tracé d'un trait d'une longueur de 10
t1.pensize(2) # définit l'épaisseur du tracé (ici 2)
    
```

✍ Modifier le programme et la fonction `spirale(taille, n)` afin de colorier chaque segment aléatoirement.

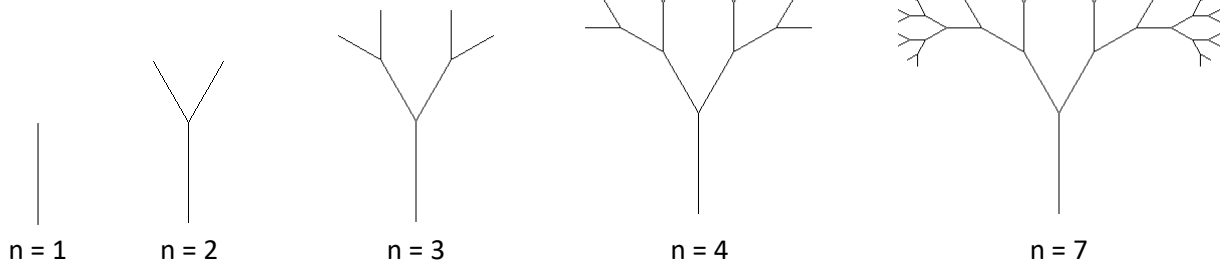
```

t1.colormode(255) # définit le mode de couleur en système RGB
t1.pencolor(r, g, b) # définit la couleur du stylo
    
```



2. Arbre fractal

Description du tracé



- Pour une profondeur de 1, la fonction dessine le tronc de hauteur h et revient à sa base.
- Pour une profondeur de 2, la fonction rajoute deux branches de même taille de hauteur $h \times 0,7$ partant du bout de la branche initial et formant un angle de 30° de part et d'autre du segment initial.
- Pour une profondeur de 3, la fonction rajoute deux branches de même taille de hauteur $h \times 0,7$ partant du bout de chaque branche obtenue avec une profondeur de 2 et formant un angle de 30° de part et d'autre du segment précédent.
- Pour une profondeur de n , la fonction rajoute deux branches de même taille de hauteur $h \times 0,7$ partant du bout de chaque branche obtenue avec une profondeur de $n-1$ et formant un angle de 30° de part et d'autre du segment précédent.

Algorithme :

si profondeur = 1

alors dessin du motif de profondeur 1

sinon dessin du motif de profondeur 2 avec **appel récursif de arbre**(en profondeur - 1) **pour le dessin de chaque branche**

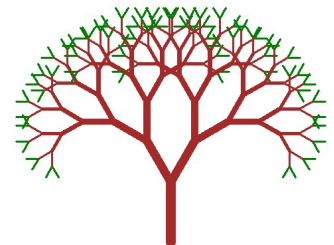
☞ Implémenter un programme incluant la fonction récursive **arbre**(hauteur, profondeur) afin d'obtenir l'arbre.

☞ Modifier l'implémentation du programme et de la fonction arbre afin d' :

- avoir le bout des branches en vert et les branches en marron ;
- ajouter un paramètre **épaisseur** à la fonction afin de modifier les épaisseurs des tracés (du tronc aux branches finales) ;
- ajouter un paramètre **rapport** afin de modifier le rapport de hauteur (0,7 dans la version initial).

arbre(hauteur, profondeur, épaisseur, rapport)

Le résultat ci-contre est obtenu avec les valeurs ci-dessous :
arbre(90, 8, 15, 0.8)

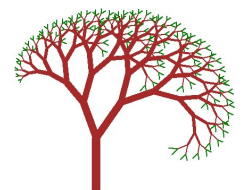


Encore plus fort !

☞ Modifier la fonction arbre précédente afin de pouvoir proposer un rapport de hauteur différent à gauche et à droite ainsi qu'un angle différent à gauche et à droite.

arbre(hauteur, profondeur, épaisseur, angle_g, angle_d, rapport_g, rapport_d)

Le résultat ci-contre est obtenu avec les valeurs ci-dessous :
arbre(100, 9, 16, 20, 35, 0.7, 0.8)



Source : <https://lewebpedagogique.com/infoloth/>