

FRACTAL EN PYTHON. UN ARBOL EN COLOR

1. Explicacion del código

```
1 import turtle
2 import random
3
4 #Variables globales
5 ANG = 35 #ángulo de inclinación para las ramas
6 RAND = 40 #factor de aleatoriedad del ángulo de inclinación (grados)
7 REL = 4/5 #relación entre la rama y las sub ramas
8 RANDT = 50 #factor de aleatoriedad en el tamaño de las ramas (%)
9 GROSORTRONCO = 0 #pixeles que se le suman al grosor del árbol
10 TAMINIC = 70 #tamaño del tronco inicial en pixeles
11 TAMHOJA = 10 #tamaño de la hoja
12 ANGHOJA = 70 #ángulo de las puntas de las hojas (180 = círculos)
13 PROF = 9 #cantidad de niveles en el árbol (más de 10 puede durarmucho dibujándose)
14 CTRONCO = (139,69,19) #color del tronco. tres números entre 0 y 255
15 CTRONCOVAR = 40 #factor de aleatoriedad en el color del tronco
16 CHOJAS = (42,147,17) #color de las hojas
17 CHOJASVAR = 100 #factor de aleatoriedad en el color de las hojas
18 CFONDO = (255,255,255) #color de fondo
19
20 #Función que dibuja un árbol fractal
21 # Entradas:# t: tamaño del segmento inicial en pixeles
22 # d: profundidad total del árbol
23 # Salidas:# Dibujo del árbol en pantalla
24 # Restricciones: no
25 def arbol(t, d):
26     if d==0:
27         turtle.forward(t)
28         hoja(TAMHOJA, ANGHOJA)
29         turtle.penup()
30         turtle.back(t)
31         turtle.pendown()
32         turtle.color(CTRONCO)
33     return
```

La primera rama del código es una definición de las variables que vamos a estar utilizando, por ejemplo, el ángulo de las ramas, el color de las hojas, el tamaño inicial del tronco o la relación de tamaño entre las ramas y el tronco. También se ve la definición de lo que es el árbol, lo primero, la T, es el tamaño inicial del segmento y D la profundidad. Si la profundidad es 0, se inicia el proceso de dibujar el tronco inicial, y vuelve hacia atrás "return". Cuando ha dibujado el tronco, se empiezan a dibujar las curvaturas con grado, tamaño y color aleatorio de las ramas que crecen con proporción 4/5 como vemos abajo aunque el color siempre dentro de los parámetros que escribimos arriba.

```
34 else:
35     angulo1 = ANG + random.randrange(-RAND, RAND+1)
36     angulo2 = ANG + random.randrange(-RAND, RAND+1)
37     tamano = t + t*random.randrange(-RANDT, RANDT+1)/100
38     colortronco = variacioncolor(CTRONCO, CTRONCOVAR)
39     turtle.color(colortronco)
40     turtle.pensize(d+GROSORTRONCO)
41     turtle.forward(tamano)
42     turtle.left(angulo1)
43     arbol(t*REL, d-1)
44     turtle.right(angulo1+angulo2)
45     arbol(t*REL, d-1)
46     turtle.color(colortronco)
47     turtle.left(angulo2)
48     turtle.penup()
49     turtle.back(tamano)
50     turtle.pendown()
51     # Función que dibuja una hoja# Entradas:
52     # t: tamaño de la hoja# a: ángulo de las puntas de las hojas# Salidas:
53     # Dibujo de una hoja en la posición actual de la tortuga
54     # Restricciones: n
55 def hoja(t, a):
56     turtle.color(variacioncolor(CHOJAS, CHOJASVAR))
57     turtle.begin_fill()
58     turtle.right(a/2)
59     turtle.circle(t, a)
60     turtle.left(180-a)
61     turtle.circle(t, a)
62     turtle.left(180-a/2)
63     turtle.end_fill()
64     # Función que genera una variación de un color en RGB
65     # Entradas:
66     # color: tupla con tres valores enteros entre 0 y 255
67     # var: cantidad máxima de variación permitida en los valores RGB
68     # Salidas:# Tupla de tres valores enteros entre 0 y 255 que es una variación# del color original.
69     # Restricciones: no
```

Al terminar esta función se pasa a definir las hojas con su variación de color, tamaño, ángulo de la punta de la hoja, y profundidad.

```

70 def variacioncolor(color, var):
71     Rd = random.randrange(-var, var+1)
72     Gd = random.randrange(-var, var+1)
73     Bd = random.randrange(-var, var+1)
74     R, G, B = color
75     R += Rd
76     G += Gd
77     B += Bd
78     if R > 255:
79         R = 255
80     elif R < 0:
81         R = 0
82     if G > 255:
83         G = 255
84     elif G < 0:
85         G = 0
86     if B > 255:
87         B = 255
88     elif B < 0:
89         B = 0
90     return R, G, B
91
92     # Funcion que inicia la posicion de la tortuga e invoca la funcion
93     # de dibujar arbol fractal
94
95 def init():
96     turtle.speed(100)
97     turtle.colormode(255)
98     turtle.clear()
99     turtle.penup()
100    turtle.home()
101    turtle.left(90)
102    turtle.back(200)
103    turtle.pendown()
104    turtle.hideturtle()
105    turtle.color(CTRONCO)
106    turtle.bgcolor(CFONDO)
107    arbol(TAMINIC, PROF)
108    turtle.done()

```

En este apartado pasamos a definir las variaciones de color en RGB, arriba definimos el color marrón como (139,69,19) y el verde como (42,147,17). También arriba definimos los porcentajes de variabilidad de los colores, que pusimos en 100% para que cada hoja fuese de un color, al igual que las ramas.

Por ultimo este bloque de código lo utilizamos para dibujar el fractal con todas las definiciones y variables que hemos “explicado” y determinado antes.

```

109
110 init()

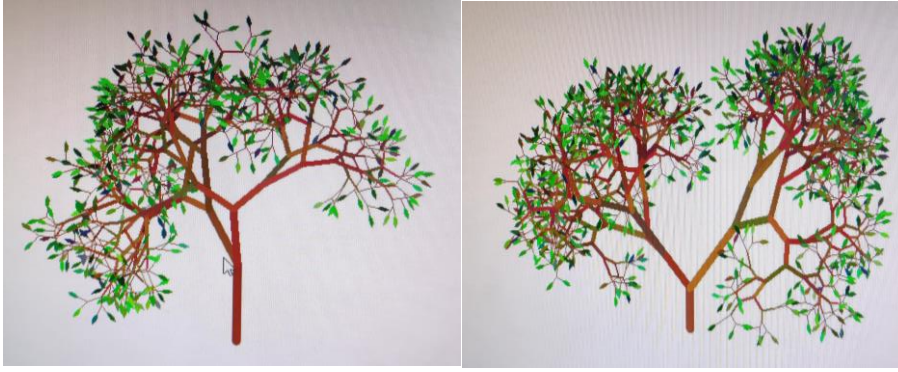
```

Esta última línea de código inicia el código INIT que es el fractal.

2. Imágenes del resultado de este fractal

Ya que este fractal tiene variabilidad en aspectos como el ángulo de las ramas o el grosor, cada vez que se ejecuta se consigue un resultado diferente.





3. Conclusión

Con los fractales se pueden conseguir cosas bastante increíbles, mas aun si sabe dominar este tipo de códigos. Es capaz de generar formas muy creativas.