

# Trabajo libre Phyton

Árbol fractal con colores y hojas

25/01/2019

Raúl Moral y Pablo Zamorano

## Contenido

|    |                      |   |
|----|----------------------|---|
| 1- | Introducción .....   | 1 |
| 2- | Código.....          | 1 |
| 3- | Funcionamiento ..... | 2 |

## 1-Introducción

A partir del código, que hemos utilizado para dibujar un fractal, hemos hecho las modificaciones pertinentes, y hemos conseguido añadirle la forma de un árbol y dibujar hojas de colores. También hemos jugado con los ángulos en los brotan las ramas del árbol.

## 2-Código

```
import turtle }  
import random }
```

Empezamos con estos códigos, que sirven para importar el paquete turtle y el paquete random directamente desde phyton sin tener que descargar nada

```
ANG = 30 }  
RAND = 30 }  
REL = 4/5 }  
RANDT = 60 }  
GROSORTRONCO = 0 }  
TAMINIC = 100 }  
TAMHOJA = 5 }  
ANGHOJA = 180 }  
PROF = 10 }
```

Esta variable es la que indica el ángulo de inclinación inicial de las ramas del árbol

Que define el ángulo de variación que se le sumará o restará al ángulo base de cada rama

Esta variable indica la relación que hay de medida entre la rama que se va a crear y la rama de la que viene

Indica el porcentaje de variación aleatorio que se le va a aplicar a la rama

Es una constante que indica cuántos pixeles de grosor debe sumársele al dibujo en general

Indica el tamaño en pixeles del segmento principal del árbol

Definen la apariencia de las hojas del árbol, en este caso el tamaño de la hoja

Definen la apariencia de la hoja pero en este caso el ángulo de la punta

Indica cuántos niveles de profundidad se van a utilizar en la construcción del árbol.

```
def arbol(t, d):
    if d==0:
        turtle.forward(t)
        hoja(TAMHOJA, ANGHOJA)
        turtle.penup()
        turtle.back(t)
        turtle.pendown()
        turtle.color(CTRONCO)
        return
    else:
        angulo1 = ANG + random.randrange(-RAND, RAND+1)
        angulo2 = ANG + random.randrange(-RAND, RAND+1)
        tamano = t + t*random.randrange(-RANDT, RANDT+1)/100
        colortronco = variacioncolor(CTRONCO, CTRONCOVAR)
        turtle.color(colortronco)
        turtle.pensize(d+GROSORTRONCO)
        turtle.forward(tamano)
        turtle.left(angulo1)
        arbol(t*REL, d-1)
        turtle.right(angulo1+angulo2)
        arbol(t*REL, d-1)
        turtle.color(colortronco)
        turtle.left(angulo2)
        turtle.penup()
        turtle.back(tamano)
        turtle.pendown()
```

Aquí es cuando nace la definición de árbol y utilizaremos un condicional con todos los ángulos y tamaños y variables puestos anteriormente para formar el tronco y el principio del árbol

Tras haber formado el principio del árbol, toca formar todas las hojas y ramas, las primeras líneas sirven para modificar los ángulos y que sean aleatorios y lo mismo con el color.

```
def hoja(t, a):
    turtle.color(variacioncolor(CHOJAS, CHOJASVAR))
    turtle.begin_fill()
    turtle.right(a/2)
    turtle.circle(t, a)
    turtle.left(180-a)
    turtle.circle(t, a)
    turtle.left(180-a/2)
    turtle.end_fill()
```

```
turtle.exitonclick()
```

Se terminará la animación cuando se clique en el dibujo

Las hojas deben darse sólo en las ramas que no se dividen más, es aquí donde deben ubicarse los comandos para dibujar la hoja. Se utiliza una función que se encarga de dibujar dos segmentos de círculo iguales unidos por sus extremos formando ángulos que serán las puntas de la hoja.

### 3-Funcionamiento

Tras poner el código en python, comienza a dibujarse un árbol variando los colores de cada hoja, y el ángulo de apertura de cada rama. Comienza con el tronco, el cual cuanto más se ramifica más pequeñas las ramas se van haciendo. Esto se debe a la relación puesta anteriormente, señalada por arriba, que es  $2/3$ .

Tras definir "árbol", se pondrán los movimientos y colores que dibujará el lápiz, en este caso los movimientos están más o menos predefinidos aunque hay una nota de aleatoriedad y en el caso de los colores sí que es totalmente aleatorio.

Después de empezar el tronco la ramificación será cada vez mayor y las ramas más finas acabando con la formación de las hojas. Una vez que se han dibujado una rama completa con sus hojas, el lápiz retrocederá para provocar otra entrada para hacer otra bifurcación y así hasta el final.

## 4- Conclusión

En este trabajo hemos tenido que aprender a programar más de lo que nos han enseñado en la clase para tratar de aspirar a la perfección, hemos estado trabajando desde casa también y visitando un montón de páginas para poder formarlos poco a poco y sobre todo entendiéndolo parte por parte. Al final juntando códigos hemos conseguido hacer un árbol fractal muy diferente partiendo del árbol fractal que dimos en clase.

## 5- Vídeo



2019-01-25-13-58-4  
4\_Trim.mp4