

## **Kit de programación Código Pi**

**Arte moderno con Python**

**APRENDER  
CONECTADOS**



Ministerio de Educación,  
Cultura, Ciencia y Tecnología  
Presidencia de la Nación



## Autoridades

### **Presidente de la Nación**

Mauricio Macri

### **Jefe de Gabinete de Ministros**

Marcos Peña

### **Ministro de Educación, Cultura, Ciencia y Tecnología**

Alejandro Finocchiaro

### **Secretario de Gobierno de Cultura**

Pablo Avelluto

### **Secretario de Gobierno de Ciencia, Tecnología e Innovación Productiva**

Lino Barañao

### **Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología**

Manuel Vidal

### **Secretaria de Innovación y Calidad Educativa**

Mercedes Miguel

### **Subsecretario de Coordinación Administrativa**

Javier Mezzamico

### **Directora Nacional de Innovación Educativa**

María Florencia Ripani

ISBN en trámite



Este material fue producido por el Ministerio de Educación, Cultura, Ciencia y Tecnología en base a contenidos provistos sin cargo por la Fundación Raspberry Pi mediante licencias Creative Commons y han sido desarrollados en función de los Núcleos de Aprendizajes Prioritarios de educación digital, programación y robótica y los recursos tecnológicos propuestos en el marco del Plan Aprender Conectados.

# Índice

Arte moderno en Python.....	5
Colores aleatorios .....	6
Lugar aleatorio .....	8
Reto: Arte de tortuga .....	9
Crea arte moderno rectangular.....	10
Reto: Más arte moderno .....	13

## Arte moderno en Python

### Introducción

En este proyecto crearás arte moderno generado por computadora. Usarás funciones para escribir un código que podrás usar una y otra vez.



### Notas para los Líderes de Club

#### Introducción:

Este proyecto introduce las funciones a través de un generador de arte moderno colorido. Las funciones se usan para contener códigos de gráficas de tortuga útiles que puedan usarse fácilmente para crear arte increíble.

Recursos offline: Este proyecto puede completarse con Python 3 offline.  
Retos

- Arte de tortuga - definí una nueva función para completar el generador de arte de tortuga.

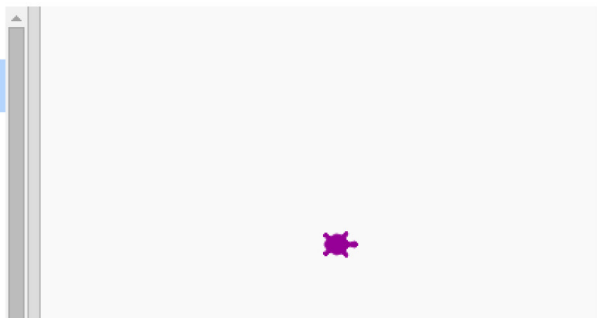
- Más arte moderno - creá una nueva función que recupere otras funciones para generar arte moderno.

## Colores aleatorios

- Podés ajustar el color de una tortuga indicando la cantidad de rojo, verde y azul deseada de 0 a 255.

Añadí el siguiente código para obtener una tortuga violeta.

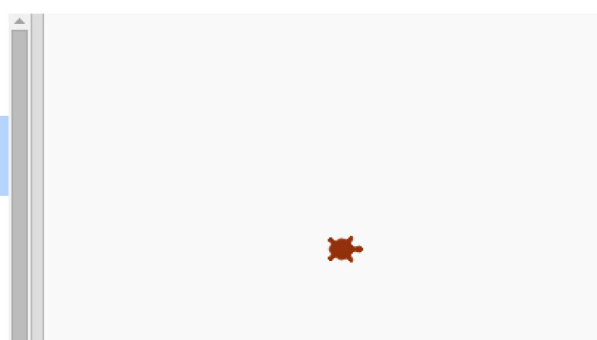
```
from turtle import *  
shape("turtle")  
color(150, 0, 150)
```



El color violeta se logra mezclando el color rojo y el color azul.

- Probá con números distintos para lograr colores diferentes.  
Recordá que cada número puede ajustarse entre 0 y 255.
- ¿Por qué no intentar seleccionar un color aleatorio?  
Actualizá tu código de modo que seleccione un número aleatorio entre 0 y 255 para los valores del color rojo, verde y azul:

```
from turtle import *  
from random import *  
  
shape("turtle")  
red = randint(0, 255)  
green = randint(0, 255)  
blue = randint(0, 255)  
color(red, green, blue)
```



- Hacé clic en 'Run' varias veces para obtener tortugas de colores distintos.
- Divertido; sin embargo, tenés que recordar y escribir demasiadas cosas cada vez que quieras una tortuga de color aleatorio y, además, no resulta fácil de leer.

En Python podemos escribir `def` para definir una función que recuperaremos cada vez que querramos crear una tortuga con un color aleatorio.

Ya has recuperado funciones anteriormente. `color()` y `randint()` son funciones que ha sido definidas para vos.

Apliquemos un código de color aleatorio usando `for` :

```
from turtle import *
from random import *

def randomcolour():
    red = randint(0, 255)
    green = randint(0, 255)
    blue = randint(0, 255)
    color(red, green, blue)

shape("turtle")
```

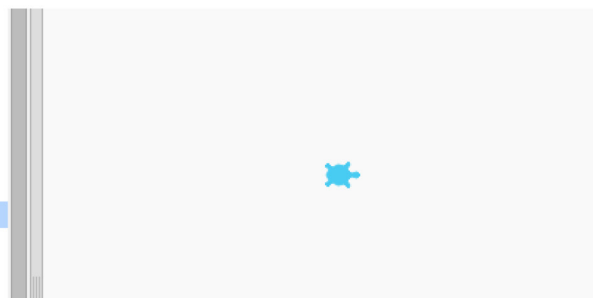


Asegurate de **indentar** el código dentro de la función. Las funciones normalmente se encuentran en la parte superior del script después de las importaciones.

- Si ejecutás tu código en estos momentos, no obtendrás una tortuga de color aleatorio. Esto se debe a que haz definido la función pero todavía no la haz llamado.
- Añade una línea para llamar tu nueva función:

```
def randomcolour():
    red = randint(0, 255)
    green = randint(0, 255)
    blue = randint(0, 255)
    color(red, green, blue)

shape("turtle")
randomcolour()
```



Tené en cuenta que tu nuevo código es mucho más fácil de entender porque la parte más compleja es la función. Es muy sencillo averiguar qué es lo que hace `randomocolor()`.

## Lugar aleatorio

Creemos otra función que mueva la tortuga a un lugar aleatorio de la pantalla. El centro de la pantalla es (0,0), por lo que colocaremos a las tortugas en un cuadrado alrededor del centro.

- Añadí una función `randomplace()`:

```
def randomcolour():  
    red = randint(0, 255)  
    green = randint(0, 255)  
    blue = randint(0, 255)  
    color(red, green, blue)
```

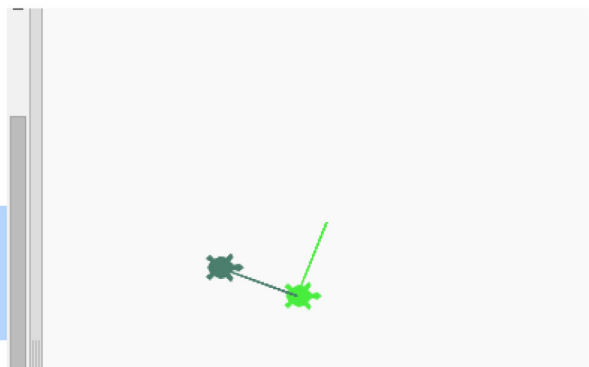
```
def randomplace():  
    x = randint(-100, 100)  
    y = randint(-100, 100)  
    goto(x, y)
```

```
shape("turtle")  
randomcolour()
```

- Probá tu nueva función recuperándola y recuperando `stamp()`; podés recuperarla más de una vez:

```
def randomplace():  
    x = randint(-100, 100)  
    y = randint(-100, 100)  
    goto(x, y)
```

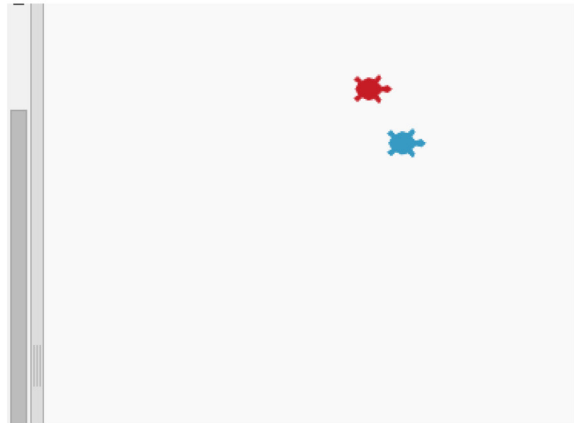
```
shape("turtle")  
randomcolour()  
randomplace()  
stamp()  
randomcolour()  
randomplace()  
stamp()
```





- ¡La tortuga dibuja mientras se mueve! Subamos el lápiz al principio y al final de modo que la tortuga no dibuje mientras se mueve:

```
def randomplace():  
    penup()  
    x = randint(-100, 100)  
    y = randint(-100, 100)  
    goto(x, y)  
    pendown()  
  
shape("turtle")  
randomcolour()  
randomplace()  
stamp()  
randomcolour()  
randomplace()  
stamp()
```



¿Notaste que solamente tenés que arreglar el código en un solo lugar? Esta es otra de las ventajas de las funciones.

- Ahora probá tu código varias veces.

## Reto: Arte de tortuga

¿Poidés definir una función `randomheading()` que haga que la tortuga apunte en una dirección aleatoria y lograr que el siguiente código funcione?

```
shape("turtle")  
for i in range(30):  
    randomcolour()  
    randomplace()  
    randomheading()  
    stamp()
```



Sugerencias:

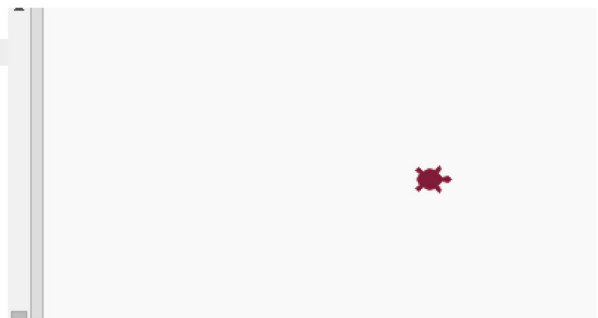
- `setheading(<number>)` cambiará la dirección de la tortuga.
- `<number>` debe encontrarse entre 1 y 360 (los grados de un círculo)
- Podés usar `randint(1, 360)` para seleccionar un número entero aleatorio entre 1 y 360.

## Crea arte moderno rectangular

Creemos arte moderno dibujando muchos rectángulos de distintos tamaños y colores.

- En primer lugar, añadí el siguiente código en la parte inferior de tu script, después de tu código reto, para borrar la pantalla y orientar la tortuga en la dirección estándar:

```
clear()  
setheading(0)
```



- Podés comentar el código de arte de tortuga colocando una `#` al principio de cada línea de modo que no se ejecute mientras trabajás en el rectángulo. (A continuación, podrás eliminar el comentario para mostrar completamente tu obra de arte).

```
#for i in range(1, 30):  
#    randomcolour()  
#    randomplace()  
#    randomheading()  
#    stamp()
```

**Puedes comentar el código usando  
"#" para que este no se ejecute**

- ¡Añadamos una función que dibuje un rectángulo de color y tamaño aleatorios en una ubicación aleatoria!

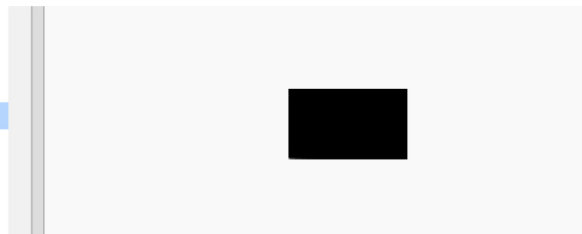
Añadí una función `drawrectangle()` después de las otras funciones:

```
def drawrectangle():  
    hideturtle()  
    length = randint(10, 100)  
    height = randint(10, 100)  
    begin_fill()  
    forward(length)  
    right(90)  
    forward(height)  
    right(90)  
    forward(length)  
    right(90)  
    forward(height)  
    right(90)  
    end_fill()
```

Consultá `snippets.py` para lograr asistencia con los códigos si querés ahorrar tiempo escribiendo.

- Añadí el siguiente código a la parte inferior de `main.py` para llamar tu nueva función:

```
clear()  
setheading(0)  
drawrectangle()
```



Ejecutá tu script un par de veces para ver los cambios de altura y anchura.

- El rectángulo siempre es del mismo color y comienza siempre en la misma ubicación.

Ahora necesitarás asignar a la tortuga un color aleatorio y moverla a un lugar aleatorio. ¡Un momento!, ¿no has creado ya funciones que lo hagan? Fantástico. Simplemente recuperalas desde el principio de la función 'drawrectangle':

```
def drawrectangle():  
    randomcolour()  
    randomplace()  
    hideturtle()  
    length = randint(10, 100)  
    height = randint(10, 100)  
    begin_fill()  
    forward(length)  
    right(90)  
    forward(height)
```



Increíble. ¡Casi no ha costado nada y es mucho más fácil de leer!

- Ahora, recuperemos `drawrectangle()` en un bucle para crear increíbles obras de arte moderno.

```
clear()  
setheading(0)  
for i in range(20):  
    drawrectangle()
```



- Anda un poco lento ¿verdad? Afortunadamente podés hacer que la tortuga vaya más rápido.

Encontrá la línea en donde estableciste la forma de la tortuga y añadí el código marcado:

```
shape("turtle")  
speed(0)
```

`speed(0)` es la velocidad más rápida; o podés usar números entre 1 (lento) y 10 (rápido). Experimentá hasta que encuentres la velocidad deseada.

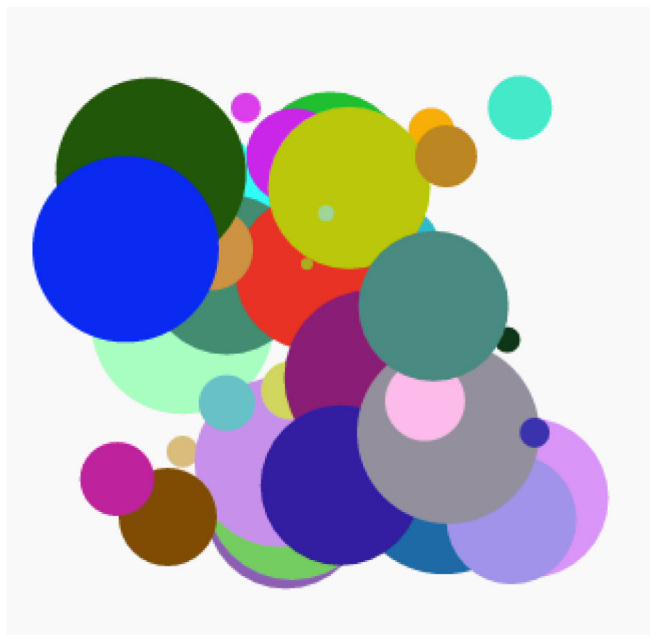
## Reto: Más arte moderno

¿Podés crear una función que dibuje una forma y llame a tus funciones `randomocolor()` y/o `randomoplace()`?

Podés llamar tu función desde un bucle `for` al igual que hiciste con el rectángulo para generar arte moderno.

### Ideas:

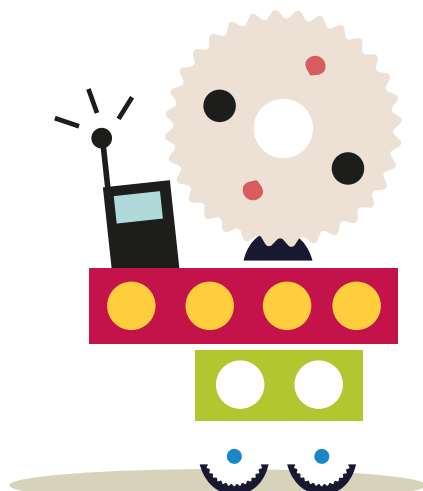
- Las tortugas tienen una función denominada `dot` (punto) que usa un radio (distancia desde el centro del borde del círculo) como entrada. P. ej., `turtle.dot(10)` Podés crear una función `drawcircle()` que dibuje un círculo con un radio aleatorio.



- Consulta en `snippets.py` el código de ejemplo para dibujar estrellas con la tortuga.



La fundación Raspberry Pi suministra contenidos de aprendizaje de programación sin cargo. Encuentre más información en <https://projects.raspberrypi.org/en/> (inglés)



**APRENDER  
CONECTADOS**



Ministerio de Educación,  
Cultura, Ciencia y Tecnología  
**Presidencia de la Nación**