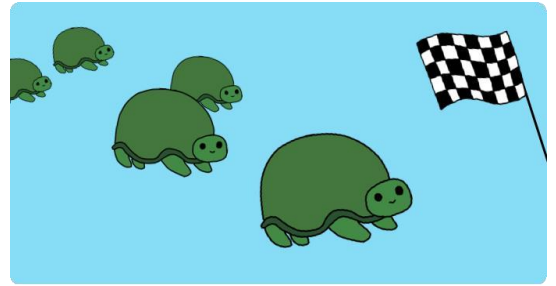




¡Carrera de Cupas!

¡Haz que los cupas compitan entre sí!

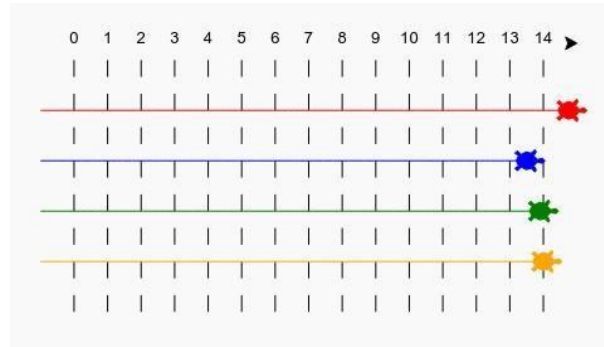


Paso 1 Introducción

Usa bucles para dibujar una pista de carreras y crear un juego de carreras de

cupas. Lo que harás

Este proyecto te inicia en bucles a través de un divertido juego de carreras de cupas. Los bucles se utilizan para dibujar la pista de carreras y para hacer que los cupas se muevan un número aleatorio de pasos cada turno. Si tienes un grupo de personas para jugar, cada persona elige un cupa y la que llega más lejos gana.



Lo que aprenderás

Al hacer tu juego de carreras de cupas, aprenderás

- cómo: Escribir bucles for en Python
- Usar números aleatorios en Python
- Dibujar líneas de diferentes colores con Python Turtle

Paso 2 Lo que necesitarás

Hardware

- Un computador conectado a Internet

Software

Este proyecto utiliza Python 3

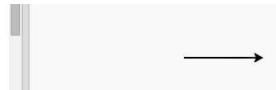
. ————— .

Paso 3 Pista de carreras

Vas a crear un juego con carreras de cupas. Lo primero que necesitarás es una pista de carreras.

- Abre la plantilla Python
- Añade el siguiente código para dibujar una línea usando la «tortuga»:

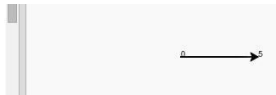
```
from turtle import *
forward(100)
```



- Ahora vamos a usar la tortuga para dibujar algunas marcas de pista para la carrera. La función write de la tortuga permite escribir texto en la pantalla.

Pruébalo:

```
from turtle import *
write(0)
forward(100)
write(5)
```



- Ahora necesitas completar los números intermedios para crear marcas:

```
write(0)
forward(20)
write(1)
forward(20)
write(2)
forward(20)
write(3)
forward(20)
write(4)
forward(20)
write(5)
forward(20)
```



- ¿Te das cuenta que tu código es muy repetitivo? Lo único que cambia es el número que se escribe. Hay una mejor manera de hacer esto en Python. Puedes usar un bucle for. Actualiza tu código para usar un bucle for:

```
1 from turtle import *
2
3 for paso in range(5):
4     write(paso)
5     forward(20)
```



- Mmm, eso solo imprime números hasta el 4. En Python range(5) devuelve cinco números, de 0 a 4. Para que también devuelva el 5, deberás utilizar range(6):

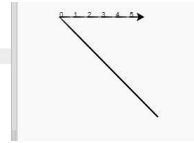
```
for paso in range(6):
    write(paso)
    forward(20)
```



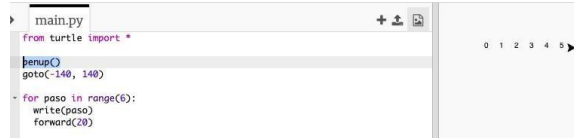
- Ahora podemos dibujar algunas marcas de pista. La tortuga comienza en las coordenadas (0,0) en el centro de la pantalla.

Mejor mueve la tortuga a la parte superior izquierda:

```
from turtle import *
goto(-140, 140)
for paso in range(6):
    write(paso)
    forward(20)
```



- ¡Ah, primero querrás levantar el lápiz!



- En lugar de dibujar una línea horizontal, dibujemos líneas verticales para crear una pista:



`right(90)` hace que la tortuga gire a la derecha 90 grados (en ángulo recto). Al mover `forward(10)` antes de dejar el lápiz, deja un pequeño espacio entre el número y el inicio de la línea. Después de dibujar la línea, levanta el lápiz y ve hacia atrás 160: `backward(160)`, la longitud de la línea más la separación.

- Se ve mejor si centras los números:

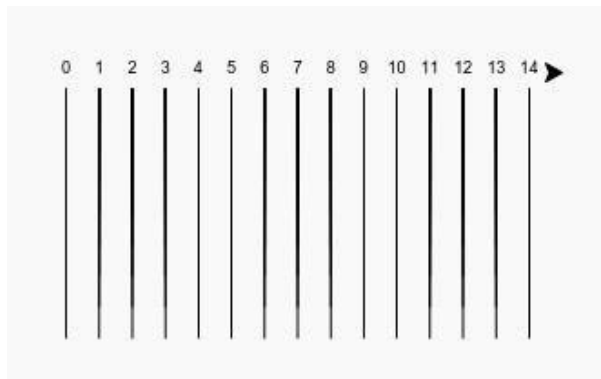


- Y puedes acelerar la tortuga para que se dibuje más rápido:

```
from turtle import *
speed(10)
penup()
goto(-140, 140)
```

Paso 4 Desafío: Más líneas

¿Puedes cambiar tu código para que las líneas de la pista crucen la pantalla?



Si quieres que la tortuga vaya más rápido aún, puedes utilizar `speed(0)`.

Paso 5 Carreras de tortugas

Ahora vamos a la parte divertida. Añadamos algunas cupas de carreras. Sería realmente aburrido si los cupas hicieran siempre lo mismo, así que se moverán un número aleatorio de pasos en cada turno. Quien gana es la tortuga que llega más lejos en 100 turnos.

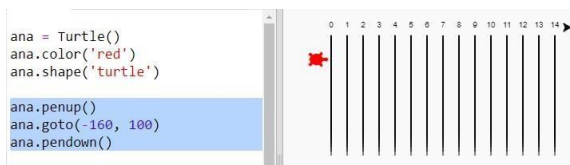
- Cuando usas comandos como `forward(20)` estás usando una sola tortuga, pero puedes crear más cupas. Añade el siguiente código al final de tu script (pero asegúrate de que no tenga sangría):



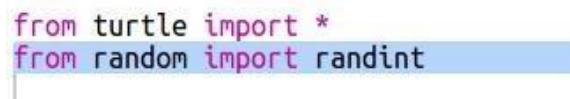
La primera línea crea una tortuga llamada «ana». Las siguientes líneas establecen el color y la forma de la tortuga.

¡Ahora realmente parece una tortuga!

- Vamos a enviar la tortuga a la línea de partida:



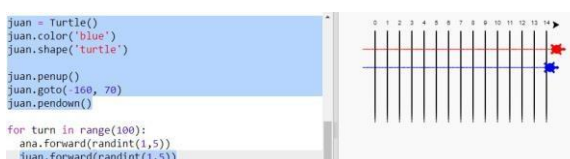
- Ahora necesitas hacer que la tortuga se mueva un número aleatorio de pasos a la vez. Necesitarás la función `randint` de la librería de Python `random`. Añade esta línea `import` en la parte superior de tu script:



- La función `randint` devuelve un número aleatorio (número entero) entre los valores elegidos. La tortuga avanzará 1, 2, 3, 4 o 5 pasos en cada turno.



- ¡Una tortuga no es suficiente para una gran carrera! Añadamos otra:

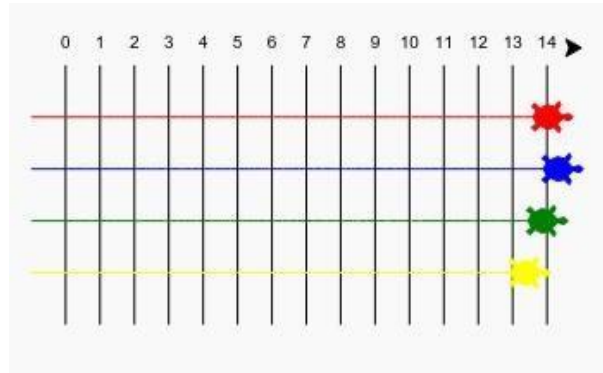


Ten en cuenta que el código para mover la tortuga azul debe estar en el mismo bucle `for` al igual que el código para mover la tortuga roja para que cada una haga un movimiento en cada turno.

Paso 6 Desafío: ¡Hora de correr!

Ahora estás listo para correr. Elige una tortuga, un oponente y mira quién gana.

¿Puedes agregar más cupas para poder competir con más amigos?



Los colores incluyen: anaranjado, morado, violeta, rojo, turquesa, magenta y marrón, ¡o puedes ir a jumpto.cc/colours (<http://jumpto.cc/colours>) y elegir cualquier otro color que te guste!

Paso 7 Desafío: Hacer una pirueta

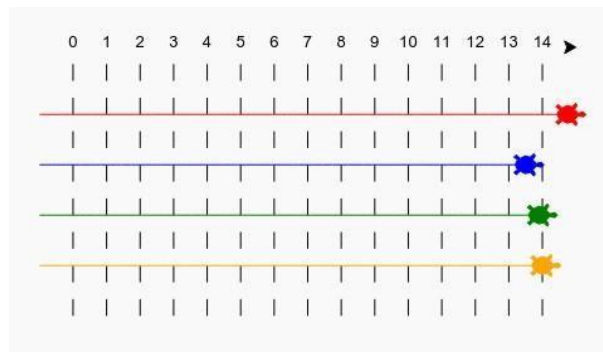
¿Puedes usar un bucle `for` giro `in range()`: para hacer que cada tortuga gire 360 grados después de llegar a la línea de salida? ¡Deberás asegurarte de que estén orientadas en la dirección correcta al comienzo de la carrera!

`ana.right(36)` girará 36 grados a la derecha la tortuga roja.

Pista: Un giro completo es 360 grados. Una tortuga puede girar a la derecha 10 grados 36 veces, o a la izquierda 5 grados 72 veces, ¡o cualquier otro número que haga 360!

Paso 8 Desafío: Líneas discontinuas

¿Puedes usar un bucle para hacer que las líneas de la pista sean discontinuas en lugar de continuas?



Pista: encuentra el código que dibuja una línea recta. Intenta usar: `for`, `forward()`, `penup()` y `pendown()`
