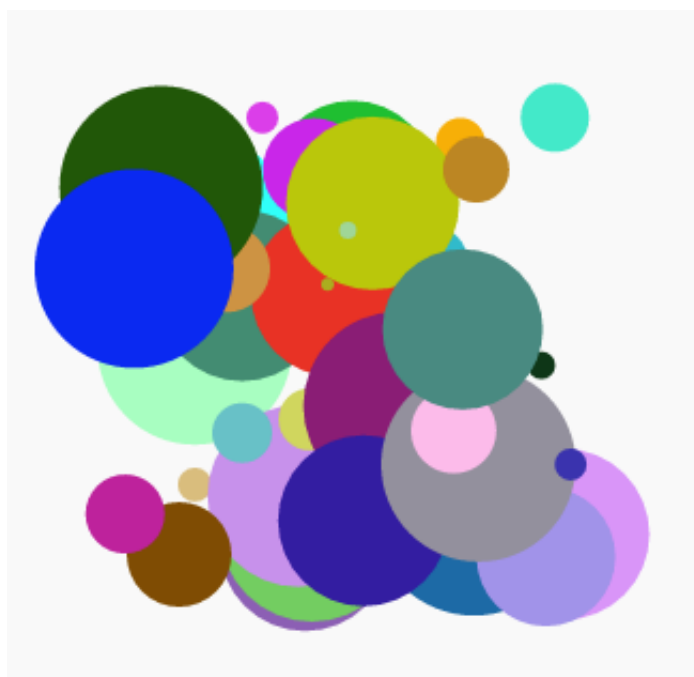


All Code Clubs must be registered. By registering your club we can measure our impact, and we can continue to provide free resources that help children learn to code. You can register your club at codeclubworld.org.

Uvod

U ovom projektu naučit ćeš pomoću računala napraviti modernu umjetnost. Pomoću funkcija napisat ćeš kôd koji ćeš moći koristiti kad god budeš htio.



Activity Checklist

Follow these **INSTRUCTIONS** one by one



Test your Project

Click on the green flag to **TEST** your code



Save your Project

Make sure to **SAVE** your work now

Korak 1: Nasumične boje



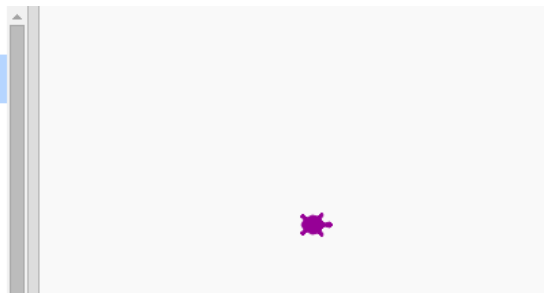
Zadatci

- Otvori ovaj trinket: jump.to/cc/modern-go. Ako čitaš ovo online, možeš koristiti i ugrađenu verziju ovog trinketa koja se nalazi ispod. ☐

- Možeš podesiti boju kornjače tako da odabereš koliko želiš crvene, zelene i plave boje od 0 do 255. ☐

Dodaj sljedeći kôd kako bi tvoja kornjača bila ljubičasta:

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



Ljubičasta se dobije miješanjem crvene i plave boje.

- Pokušaj odabrati neke druge brojeve kako bi dobio drugačije boje. ☐

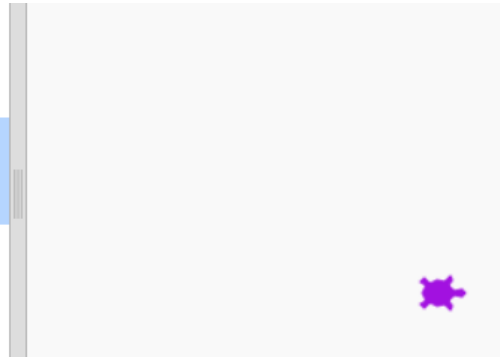
Zapamti da svaki broj može biti između 0 i 255.

- Kako bi bilo da nasumično odaberemo boju? ☐

Izmijeni svoj kôd tako da za vrijednosti crvene, zelene i plave boje odabire nasumični broj između 0 i 255:

```
from turtle import *
from random import *

shape("turtle")
crvena = randint(0, 255)
zelena = randint(0, 255)
plava = randint(0, 255)
color(crvena, plava, zelena)
|
```



- Klikni na 'Run' nekoliko puta da dobiješ kornjače različitih boja. ☐
- Zabavno je, ali i teško za zapamtiti i unositi svaki put kada želiš da se boja kornjače nasumično odredi. Također je i teško za čitati. ☐

U Pythonu možemo upisati `def` i definirati funkciju koju možemo pozvati svaki put kada želimo nasumično odabrati boju kornjače.

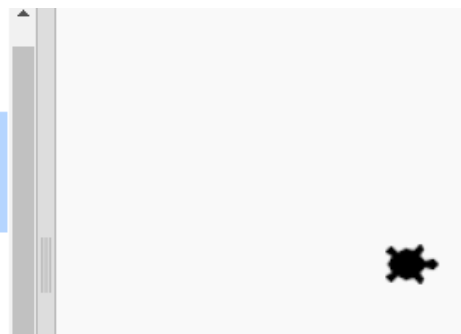
Već si pozivao funkcije, poput `color()` i `randint()` koje su bile unaprijed definirane.

Iskoristimo `def` kako bi u funkciju stavili dio kôda koji nasumično određuje boju:

```
from turtle import *
from random import *

def nasumicnaBoja():
    crvena = randint(0, 255)
    zelena = randint(0, 255)
    plava = randint(0, 255)
    color(crvena, plava, zelena)

shape("turtle")
|
```



Ne zaboravi uvući kôd unutar funkcije. Funkcije se obično pišu na početku programa, nakon naredbi `import`.

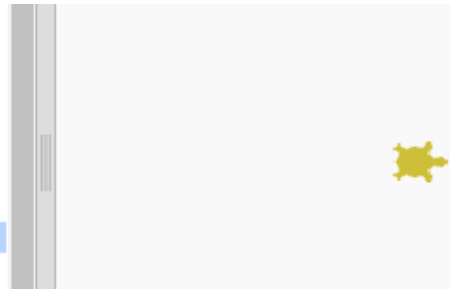
- Ako sada klikneš na 'Run' nećeš dobiti nasumično obojenu kornjaču. To je zato što si definirao svoju funkciju, ali nisi ju pozvao. ☐

- Dodaj sljedeću liniju kôda za pozivanje svoje nove funkcije:



```
def nasumicnaBoja():  
    crvena = randint(0, 255)  
    zelena = randint(0, 255)  
    plava = randint(0, 255)  
    color(crvena, plava, zelena)
```

```
shape("turtle")  
nasumicnaBoja()  
|
```



Primijeti da je novi kôd puno lakši za razumijeti jer je komplicirani dio unutar funkcije. Lako je zaključiti što funkcija `nasumicnaBoja()` radi.

Korak 2: Nasumična pozicija

✓ Zadatci

Definirajmo još jednu funkciju koja će pomicati kornjaču na nasumičnu poziciju na ekranu. Sredina ekrana ima koordinate (0,0) pa ćemo kornjače smjestiti oko centra u prostor u obliku kvadrata.

- Dodaj funkciju `nasumicnaPozicija()`:



```
def nasumicnaBoja():  
    crvena = randint(0, 255)  
    zelena = randint(0, 255)  
    plava = randint(0, 255)  
    color(crvena, plava, zelena)
```

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

```
shape("turtle")  
nasumicnaBoja()
```

- Isprobaj svoju novu funkciju tako da prvo pozoveš nju, a zatim funkciju `stamp()`. Možeš ju pozvati više puta:



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

shape("turtle")
nasumicnaBoja()
nasumicnaPozicija()
stamp()
nasumicnaBoja()
nasumicnaPozicija()
stamp()
```

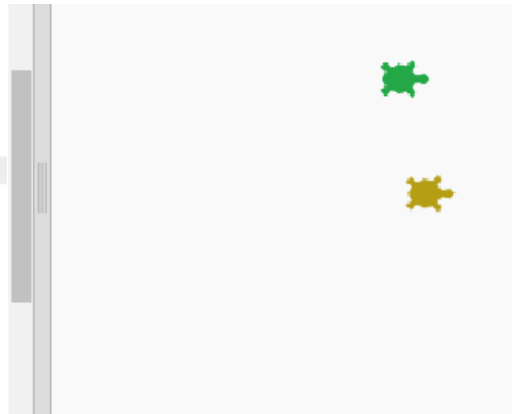


- Uuups, kornjača crta dok se pomiče. Podignimo olovku na početku i spustimo ju na kraju tako da kornjača ne crta dok se pomiče:



```
def nasumicnaPozicija():
    penup()
    x = randint(-100, 100)
    y = randint(-100, 100)
    goto(x, y)
    pendown()

shape("turtle")
nasumicnaBoja()
nasumicnaPozicija()
stamp()
nasumicnaBoja()
nasumicnaPozicija()
stamp()
```



Primjećuješ li da si morao ‘popraviti’ kôd samo na jednom mjestu? To je još jedna prednost funkcija.

- Testiraj sada svoj kôd nekoliko puta.



Spremi projekt

Izazov: Kornjačina umjetnost

Možeš li definirati funkciju `nasumicanSmjer()` kojom će kornjača pokazivati u nasumičnom smjeru i koja će učiniti da sljedeći kôd radi?

```
shape("turtle")

for i in range(30):
    nasumicnaBoja()
    nasumicnaPozicija()
    nasumicanSmjer()
    stamp()
```

Pomoć:

- ☐ `setheading(<broj>)` će promijeniti smjer u kojem je kornjača okrenuta.
- ☐ `<broj>` mora biti između 1 i 360 (broj stupnjeva unutar kruga).
- ☐ Možeš koristiti `randint(1, 360)` za odabir broja između 1 i 360.



Spremi projekt

Korak 3: Kreiraj modernu umjetnost s pravokutnicima



Zadatci

Kreirajmo sada modernu umjetnost crtajući puno pravokutnika različitih veličina i boja.

- Prvo dodaj sljedeći kôd na kraj svog programa, nakon kôda koji si zapisao dok si rješavao Izazov. Tako ćeš očistiti ekran



od crteža koje je napravila tvoja kornjača i usmjerit ćeš kornjaču u njenom uobičajenom smjeru:

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



- Kôd zbog kojeg je tvoja kornjača crtala po ekranu možeš staviti u komentar tako da na početak svake linije dodaš `#`. Tako se taj dio kôda neće pokretati dok budeš radio na crtanju pravokutnika. (Kasnije možeš maknuti znak `#` i pokazati sve što si napravio/la.)



```
#for i in range(30):
#    nasumicnaBoja()
#    nasumicnaPozicija()
#    nasumicanSmjer()
#    stamp()
```

**Znakom '#' možeš komentirati
kôd tako da se ne pokreće**

- Dodajmo sada funkciju koja će na nasumičnim pozicijama crtati pravokutnike nasumičnih veličina i boja!



Dodaj funkciju `crtajPravokutnik()` nakon svojih ostalih funkcija:

```
def crtajPravokutnik():
    hideturtle()
    duzina = randint(10, 100)
    visina = randint(10, 100)
    begin_fill()
    forward(duzina)
    right(90)
    forward(visina)
    right(90)
    forward(duzina)
    right(90)
    forward(visina)
    right(90)
    end_fill()
```

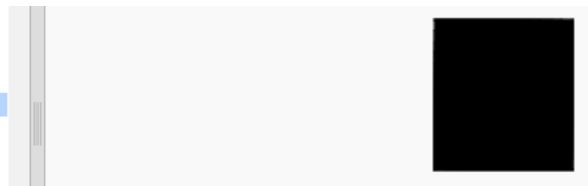
Pogledaj u `snippets.py` za pomoć sa kôdom ako želiš uštedjeti nešto vremena na tipkanju.

- Dodaj sljedeći kôd na kraj programa `main.py` kako bi pozvao svoju novu funkciju:



```
clear()
setheading(0)

crtajPravokutnik()
|
```



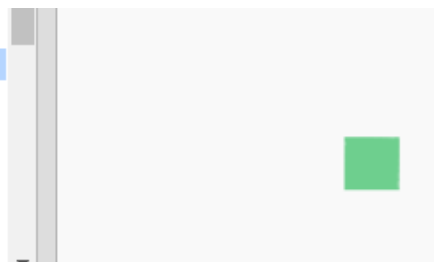
Pokreni program nekoliko puta da vidiš kako se mijenjaju visina i širina.

- Pravokutnik je uvijek iste boje i nalazi se na istom mjestu.



Sada moraš kornjaču obojati nasumičnom bojom, a zatim ju pomaknuti na nasumičnu poziciju. Hej, nisi li već definirao funkcije koje to rade? Odlično. Jednostavno ih pozovi na početku funkcije `crtajPravokutnik`:

```
def crtajPravokutnik():
    nasumicnaBoja()
    nasumicnaPozicija()
    hideturtle()
    duzina = randint(10, 100)
    visina = randint(10, 100)
    begin_fill()
    forward(duzina)
```



Opa, to je bilo brzo, a i puno je jednostavnije za pročitati.

- Pozovimo sada funkciju `crtajPravokutnik()` unutar petlje za kreiranje fora moderne umjetnosti:



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

```
for i in range(20):
    crtajPravokutnik()
```



- Uh, ovo je bilo malo sporo, zar ne? Srećom, možeš malo ubrzati kornjaču.



Pronađi liniju kôda gdje si postavio oblik na 'turtle' i dodaj označeni kôd:

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

Korištenje `speed(0)` funkcije je najbrže, ali možeš koristiti i brojeve od 1 (sporo) do 10 (brzo). Isprobavaj dok ne pronađeš brzinu koja ti odgovara.

Izazov: Još moderne umjetnosti

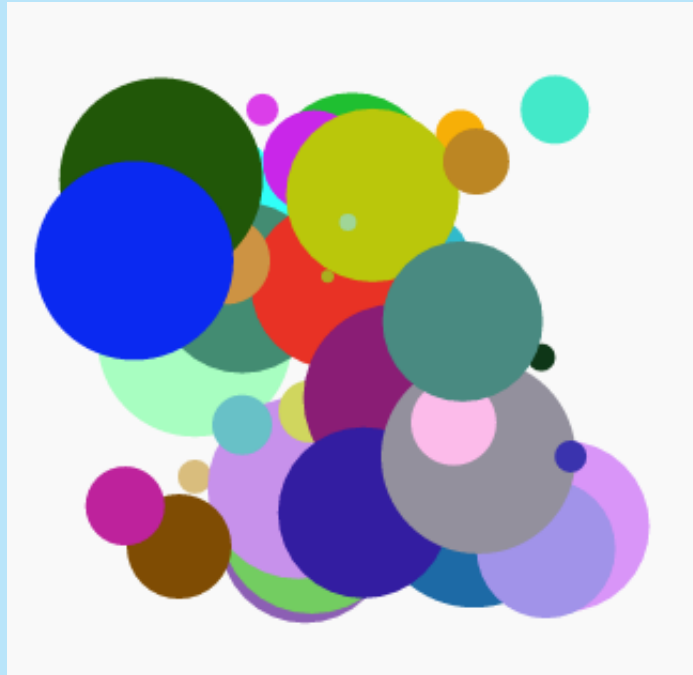
Možeš li napisati funkciju koja crta neki oblik i poziva tvoje `nasumicnaBoja()` i/ili `nasumicnaPozicija()` funkcije?

Možeš pozvati svoju funkciju unutar `for` petlje kao što si napravio kada smo kreirali modernu umjetnost s pravokutnicima.

Ideje:

- ☐ Kornjače imaju funkciju koja se zove `dot` i koja kao početnu vrijednost uzima polumjer (udaljenost od sredine

početna vrijednost polumjera (odajemo od centra do ruba kružnice). Primjerice, `turtle.dot(10)`. Možeš definirati funkciju `crtajKruznicu()` koja crta kružnicu s nasumičnom vrijednosti polumjera.



- ☐ Primjer kôda za crtanje zvijezda s kornjačama potraži u `snippets.py`.



Spremi projekt