

# Tortugues i funcions

...

Pau Giuli

# Introducció a les lliçons

- 3 Lliçons
  - Introducció a turtle
    - Instruccions
    - Repte
  - Funcions i Arguments
    - Sintaxi
    - Tortuga
    - Arguments
    - Return

# La tortuga

El *mòdul* turtle ens permet crear una finestra on podem controlar un objecte de tipus Turtle amb instruccions predeterminades.



```
1  import turtle
2
3  t = turtle.Turtle()
```

# Atributs/Objectes del mòdul

El mòdul ens permet:

- Configurar la finestra
- Crear un objecte de Tortuga
- Controlar una tortuga directament

# Atributs/Objectes del mòdul



```
1 import turtle
2
3 turtle.screensize(800, 600)
```



```
1 tortuga = turtle.Turtle()
2
3 tortuga.forward(100)
4 tortuga.left(90)
```



```
1 import turtle
2
3 turtle.forward(100)
4 turtle.color("red")
5 turtle.pencolor("blue")
```

# Funcions de l'objecte tortuga

- Moviment i gir
  - forward/fd(distància)
  - backward/back/bk(distància)
  - right/rt(angle)
  - left/lt(angle)
  - goto(coordx,coordy)
- Bolígraf
  - pendown/pd()
  - penup/pu()
  - pencolor("color")
  - pensize(mida)

# Crear un programa amb la tortuga

El nostre programa ha de tenir aquests elements necessaris:

- Importació del mòdul
- Creació de l'objecte Turtle o crida directa al mòdul
- Línies funcionals
- Fi (o no :P `turtle.done()`)

# El nostre primer programa

- Importem
- Objecte
- Configuració
- Instruccions

Què dibuixa aquest programa?



```
1  import turtle
2
3  tortuga = turtle.Turtle()
4
5  tortuga.goto(0, 0)
6  tortuga.pensize(5)
7  tortuga.speed(1)
8
9  tortuga.forward(100)
10 tortuga.left(90)
11 tortuga.forward(100)
12 tortuga.left(90)
13 tortuga.forward(100)
14 tortuga.left(90)
15 tortuga.forward(100)
16 tortuga.left(90)
```



# El bolígraf

Per defecte, la tortuga “apareix” al centre de la finestra (posició 0,0) i amb el bolígraf “baixat”. Controlar si el boli “pinta” o no, ens permetrà moure la tortuga a posicions sense dibuixar línies durant els moviments que fem.

`.penup()` - Aixequem el boli *també es pot utilitzar .pu()*

`.pendown()` - Abaixem el boli *també es pot utilitzar .pd()*

# Ara és el vostre torn!

Creeu un programa que empri totes les instruccions i entregueu el vostre codi.

[trameses.giuli.cat](https://trameses.giuli.cat)

# El Repte: Constraints/Restriccions

Com a programadors ens trobarem situacions on hem d'estalviar recursos, així doncs, hem de practicar el pensament computacional per a trobar solucions eficients als problemes.

La següent tasca consisteix a escriure codi amb restriccions, en aquest cas, treballarem utilitzant el mínim nombre de línies possibles.

# El Repte: Capses

Escriu el codi necessari per a dibuixar aquesta figura emprant el menor nombre de línies.

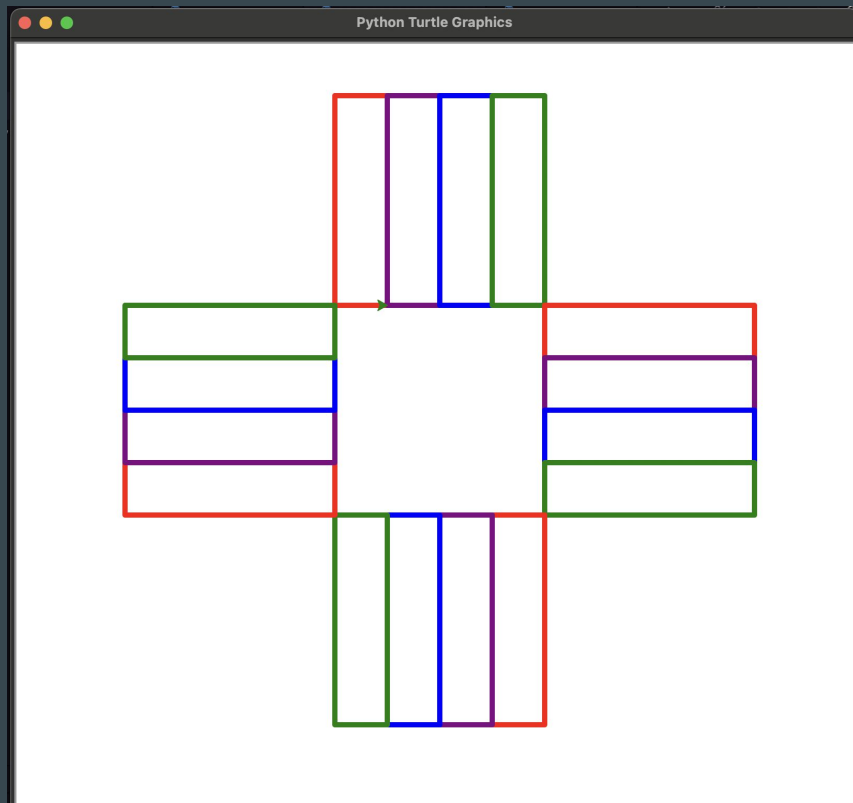
Puntuacions: (fet a color apuja nota)

<25: AE

25-29: AN

29-39: AS

>40: NA



# Funcions: Definició

Les funcions ens permeten “embolicar” un conjunt de mètodes, algoritmes, etc. que després podem fer actuant “trucant” aquesta.



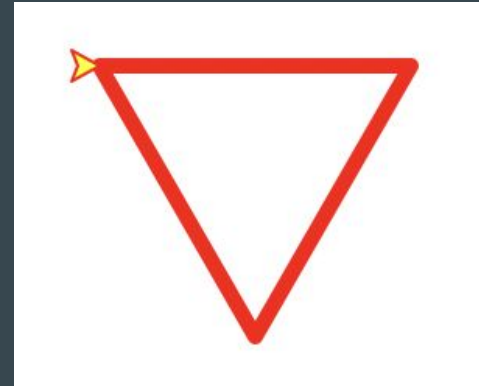
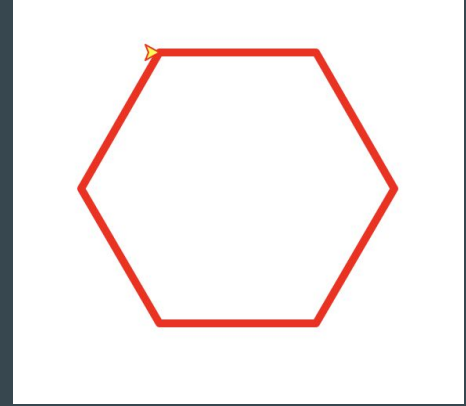
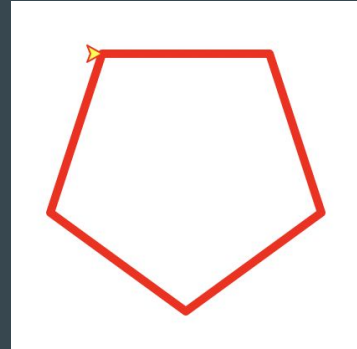
```
1  import turtle
2
3  def quadrat():
4      for i in range(4):
5          turtle.forward(100)
6          turtle.right(90)
```

# Funcions: Tasca 1

Crea un programa amb una funció que:

Dibuixi amb l'objecte Tortuga un polígon regular de costats aleatoris.

[trameses.giuli.cat](http://trameses.giuli.cat)



# Funcions: Arguments

Si necessitem utilitzar dades independents dins d'una funció, podem afegir arguments en la definició d'aquesta.



```
1 def suma_si_parells(a, b):  
2     if a % 2 == 0 and b % 2 == 0:  
3         return a + b  
4     else:  
5         return 0  
6  
7 # suma_si_parells(2, 4) # 6  
8 # suma_si_parells(2, 3) # 0
```

# Funcions: Variables globals i locals

Aquelles variables que assignem fora d'una funció es podran accedir dins d'aquesta però no a l'inrevés.



```
1  a = 2
2  b = 4
3
4  def suma_si_parells(a, b):
5      if a % 2 == 0 and b % 2 == 0:
6          return a + b
7      else:
8          return 0
9
10 suma_si_parells(a,b) # 6
11
12 salut = "Hola"
13 def saludar():
14     print(salut)
```



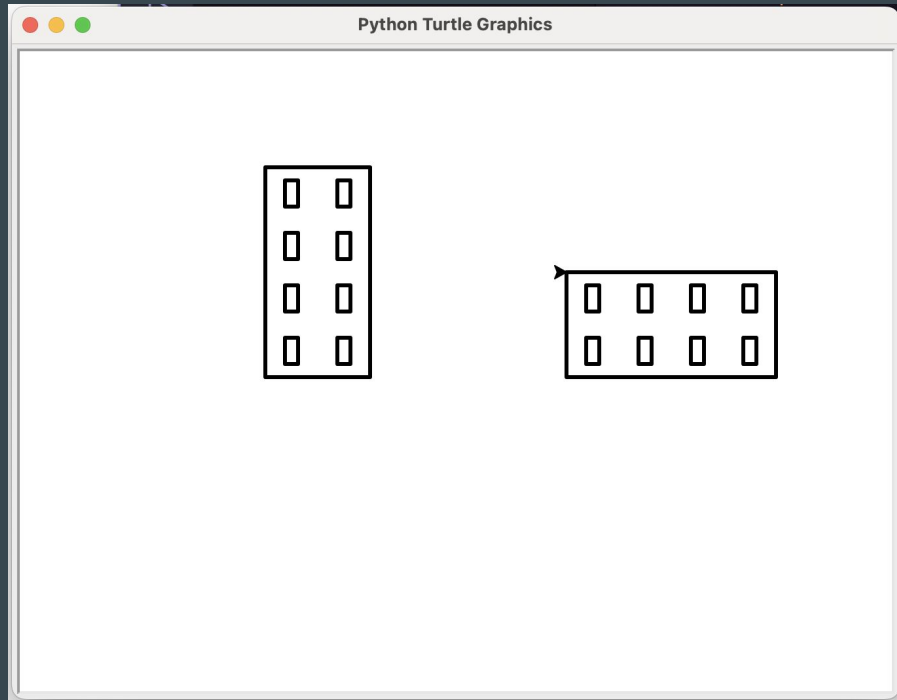
# Funcions: Tasca 2, edificis

Crea una funció que dibuixi un edifici especificant el nombre de pisos i la seva amplada (habitacions horitzontals) tal com mostra la imatge.

edifici(pisos, habitacions)

Aprox. 25-45 Línies

[trameses.giuli.cat](http://trameses.giuli.cat)

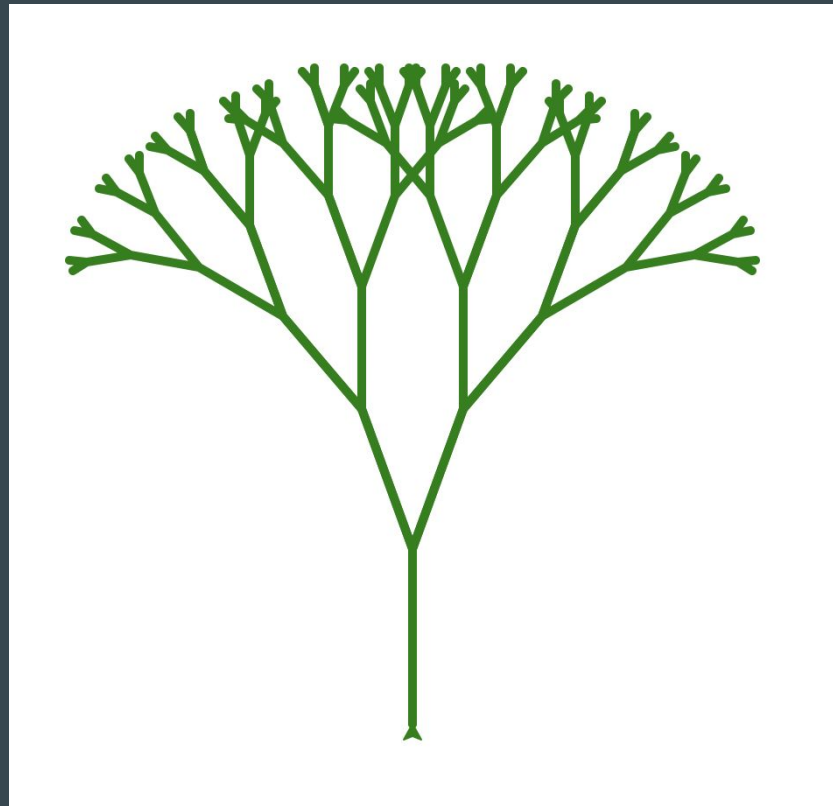


# Tasca Final

Fins aquí la teoria. Com a tasca final us toca combinar tots els coneixements que heu guanyat en un programa que entregareu abans de la següent sessió.

El tema és obert, però es puntuaran les idees creatives i el codi eficient i “bonic”.

[trameses.giuli.cat](http://trameses.giuli.cat)



Exemple d'un arbre fractal