

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "/home/angel/mmm/turt.py", line 8

forward(350)

^

IndentationError: expected an indented block

atexit copy gc inspect marshal pickle shelve sys traceback types
warnings weakref

__del__()

decimal (дрібні числа, математична точність) fractions (раціональні числа) math
(стандартні математичні функції) numbers (абстрактні базові класи для
створення чисел різних типів) random (генератор псевдовипадкових чисел)

abc array bisect collections contextlib functools heapq itertools operator

codecs re string struct unicodedata

sqlite3 Модулі доступу до баз даних типу DBM (dbm dbhash dumbdbm shelve)

TurtleScreen

tkinter.Canvas або ScrolledCanvas

RawTurtle (псевдонім: RawPen)

```
from turtle import *  
Screen()
```

```
from turtle import *  
Screen()  
forward(30)  
left(120)  
home()  
clear()
```

```
for i in range(3):  
    forward(30)  
    left(120)
```

```
for i in range(30):  
    circle(5+i*5)
```

```
left(90)
forward(10)
right(90)
```

```
for i in range(50):
    circle(100, 360)
    left(5)
for i in range(100):
    forward(10+i*5)
    left(120)
for i in range(100):
    forward(10+(i/6)*5)
    left(60)
```

```
home();clear()
for i in range(200):
    circle(100, 60-i)
    left(i)
```

```
turt4.py
import turtle
s = turtle.getscreen()
t = turtle.Turtle()
n=10
while n<=40:
    t.circle(n)
    n=n+10
```

```
from turtle import *
#shape('turtle')
#resizemode("user")
#shapeseize(5, 5, 12)
color('green','yellow')
begin_fill()
while True:
    forward(350)
    left(170)
    right(280)
    if abs(pos()) < 1:
        break
end_fill()
done()
```

```

from turtle import *
colors = ['red', 'orange', 'yellow', 'green', 'cyan', 'blue']
p = Pen()
p.speed(0)
bgcolor('black')
begin_fill()
for i in range(360):
    p.pencolor(colors[i%6])
    p.width(i/100+1)
    p.forward(i)
    p.left(59)
end_fill()
done()

```

turt2.py

```

from turtle import *
color('blue','red')
begin_fill()
def polygon(n, size=80):
    if n > 2:
        angle = 360/n          # кут повороту залежно від кількості кутів

        for n in range(0, n):  # малюємо сторони
            left(angle)
            forward(size)
# малюємо фігури з різними кутами
polygon(3)
polygon(4)
polygon(5)
polygon(6)
end_fill()
done()

```

```

home();clear()
for i in range(50):
    circle(180, 100)
    left(90)

home();clear()
for i in range(30):
    circle(100, 200)

```

```
left(99)
```

```
home();clear()
for i in range(600):
    if i % 5 == 0:
        left(3)
        forward(200)
        left(360/5)
```

```
turt3.py
import turtle
s = turtle.getscreen()
t = turtle.Turtle()
c = t.clone()
t.color("blue")
c.color("red")
t.circle(100)
c.penup()
c.fd(100)
c.pendown()
for i in range(4):
    c.fd(100)
    c.rt(90)
```

```
from turtle import *
Screen()
home();clear()
for i in range(100):
    forward(100)
    left(180)
    forward(100)
    left(180)
    left(i*3.6)
left(90)
forward(200)
for i in range(100):
    left(30)
    forward(4+i)
```