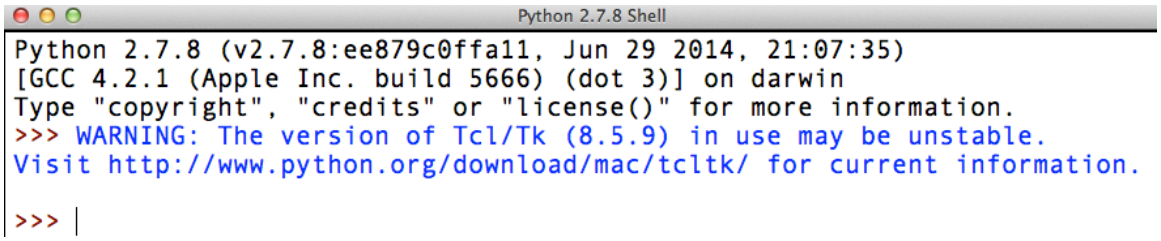
Python Turtle Lessons v9

Content created by Dr. Mark Miller, Chris Miller, and Ed Loeswick - Learningtech.org

1a. Start IDLE

Click:

See:



Introduction to IDLE Video

https://www.youtube.com/watch?v=bOvqYw1SZJg

1b. Hello, World!


```
>>> print("Hello, World!")
```


```
Hello, World!
>>>
```


```
>>> print("Python Rocks!")
```


```
>>>
Python Rocks!
>>>
```

1c. Your Name Here

```
# Use your own name, though!
```


```
>>> name = "Santa"
print(name)
```


```
>>> Santa
```

2a. Get Turtle Ready

Type:

```
>>>from turtle import *
>>>st()
```

See:

2b. Show the Turtle

Type:

```
>>>fd(100)
```

See:

2c. Turtle Right Turn

Type:

```
>>> rt(90)
```

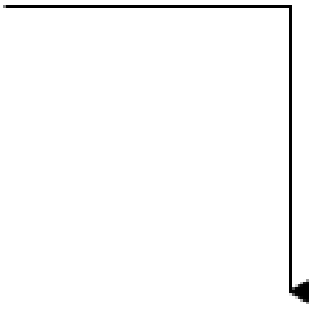
See:

3a. Go Again

Type:

```
>>> fd(100)
>>> rt(90)
```

See:

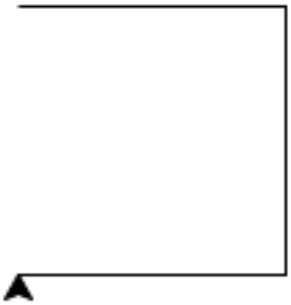


3b. And Again

Type:

```
>>> fd(100)
>>> rt(90)
```

See:

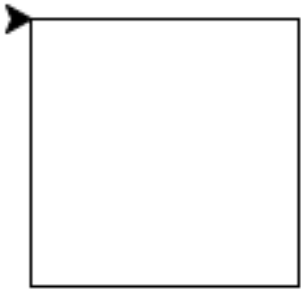


3c. And Yet Again!

Type:

```
>>> fd(100)
>>> rt(90)
```

See:



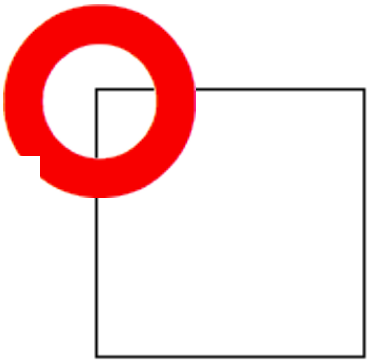
3d. Hiding and Showing

Type:

```
>>> ht()
```

See:

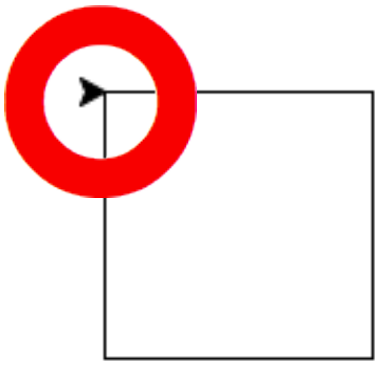
hide the turtle



Type:

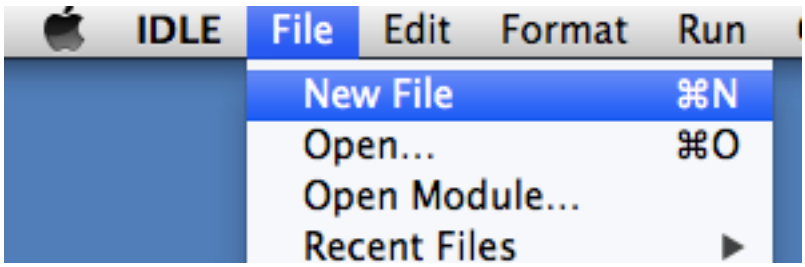
```
>>> st()
```

see the turtle

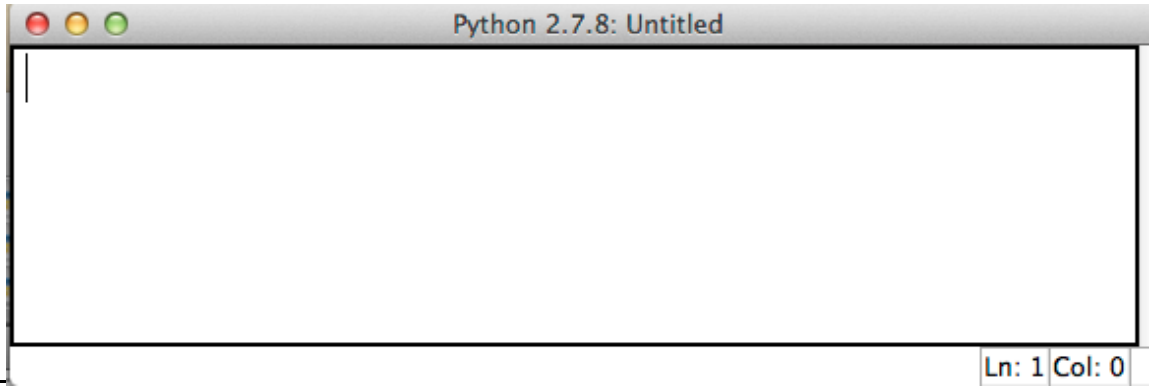


4a. Make a New File

Click:

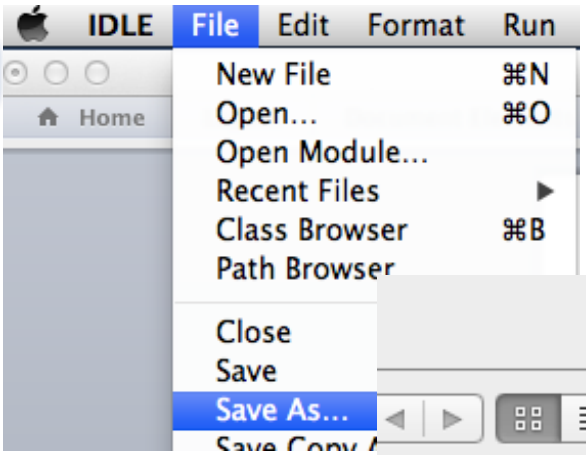


See:

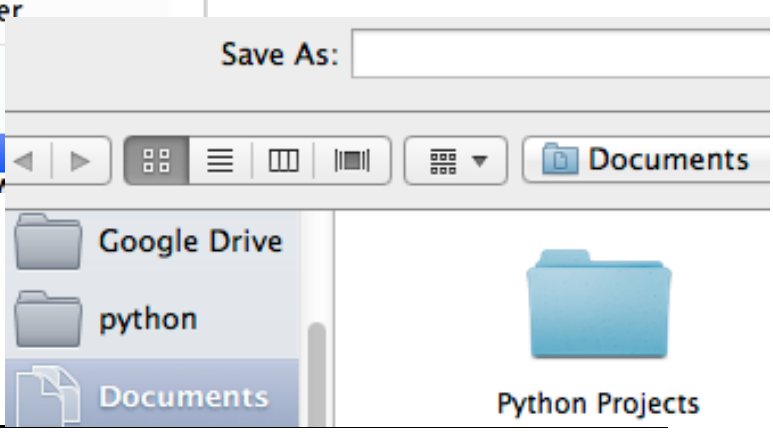


4b. Save As …

Click:



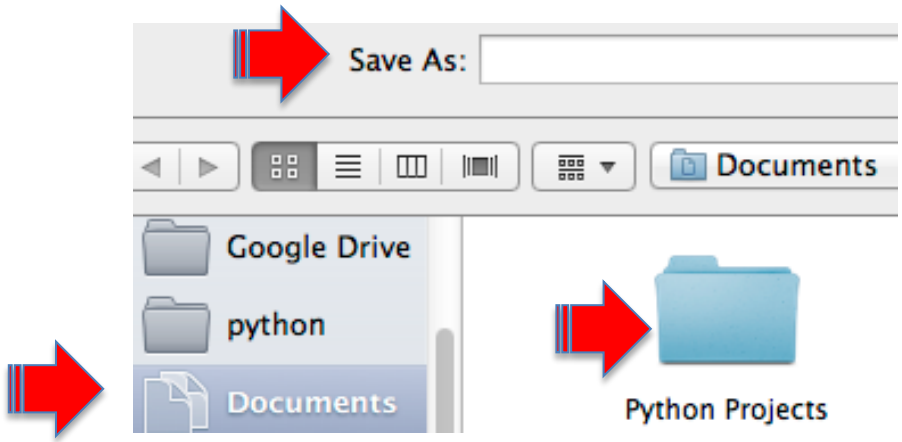
See:



4c. Save As … Python Projects

Click: Documents

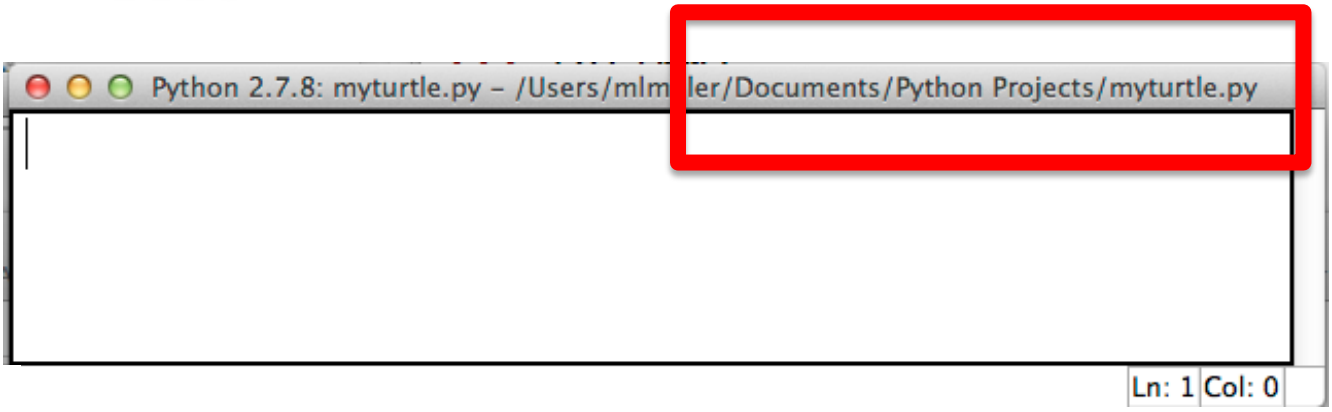
Double-Click: Python Projects



Type: myturtle.py

Click: Save

See:

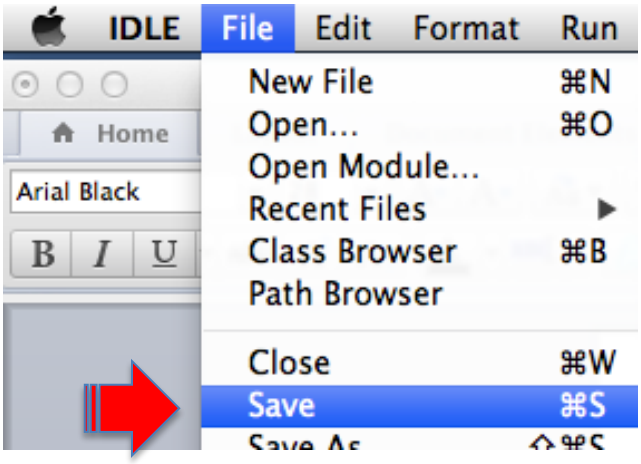


5a. Save to File

Type: (into *myturtle.py*)

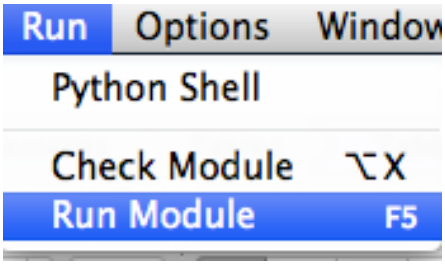
```
from turtle import *
fd(100)
rt(90)
```

Click:

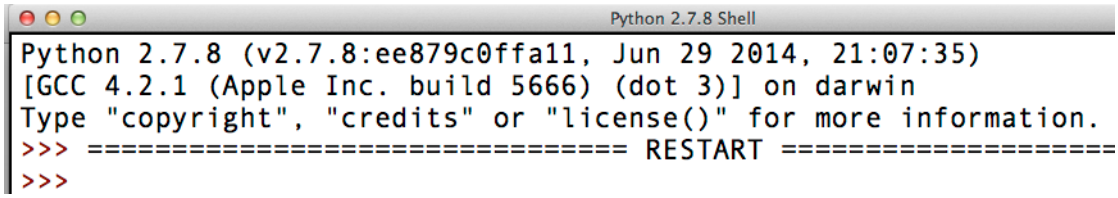


5b. Run from File

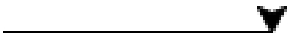
Click:



See:



See (Turtle Window):



5c. Type More into File

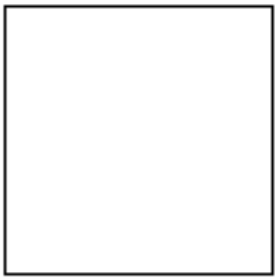
Keep Typing: (into *myturtle.py*)

```
from turtle import *
fd(100)
rt(90)
fd(100)
rt(90)
fd(100)
rt(90)
fd(100)
rt(90)
ht()
```

Click: Save ⌘S

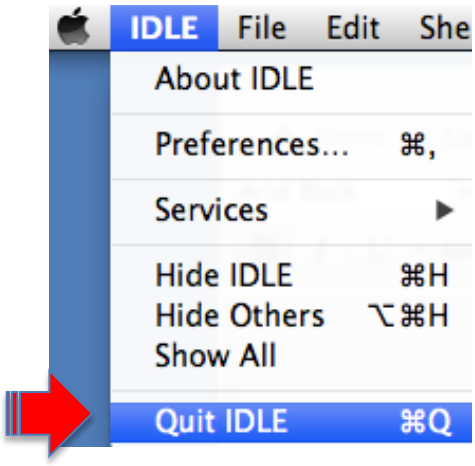
Click: Run Module F5

See (Turtle Window):

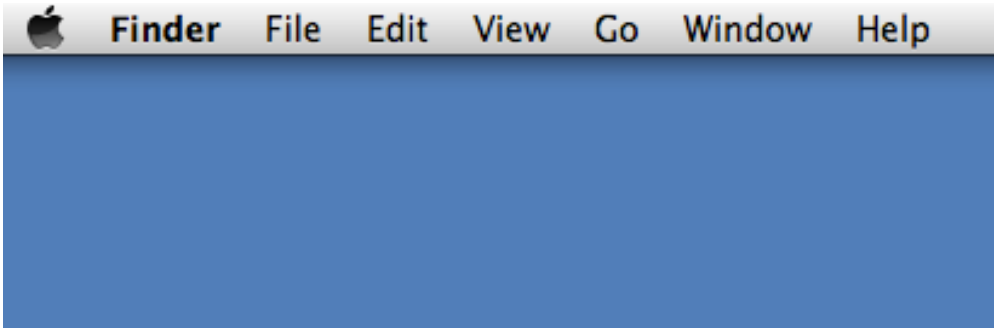


6a. Quit Python

Click:



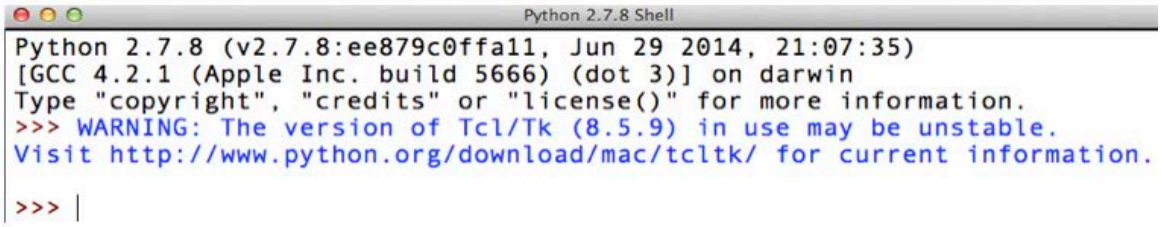
See:



6b. Start IDLE

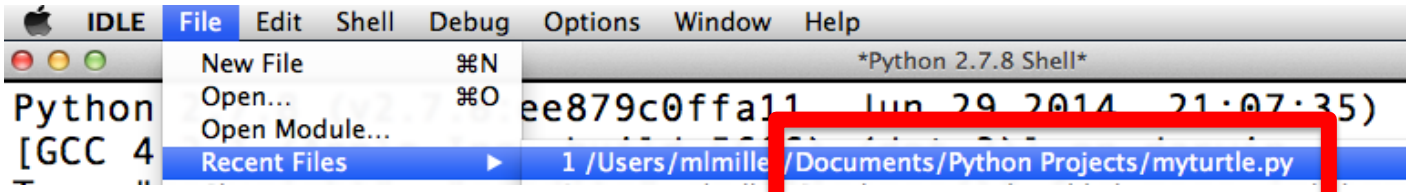
Click:

See:



6c. Open Your File

Click: *(File -> Recent Files -> myturtle.py*

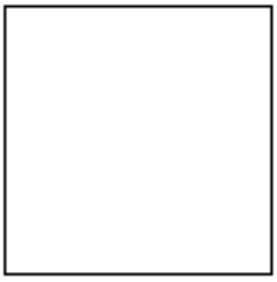


See:

6d. Run Your Code

Click: Save ⌘S **Click:** Run Module F5

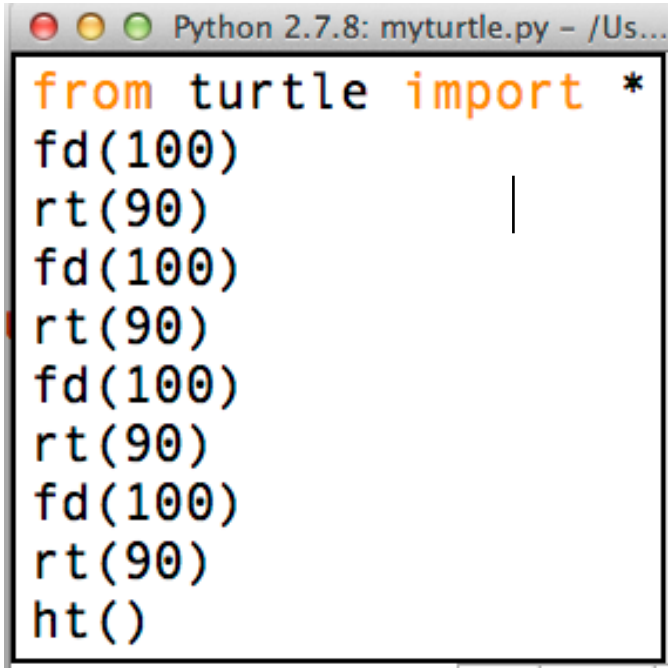
See:



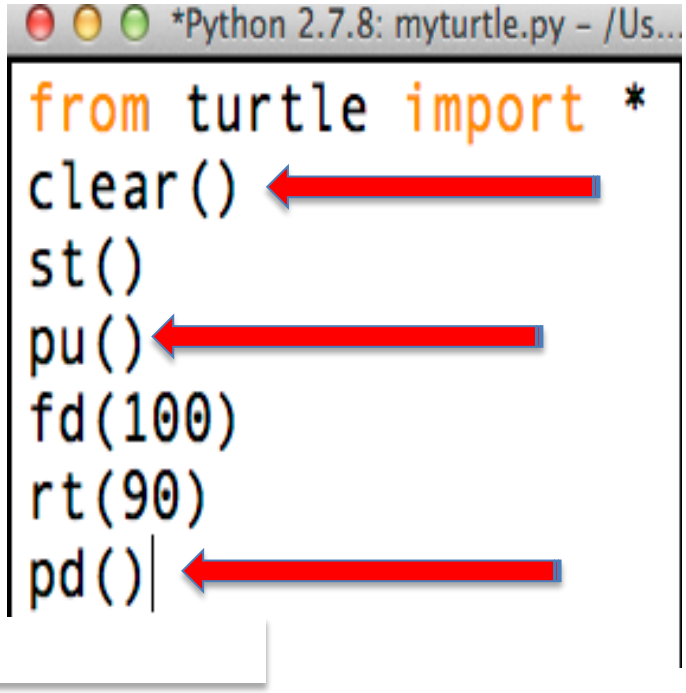
(Turtle Window)

7a. Edit Your File

See:



Type new commands:



Clean Screen

Pen Up

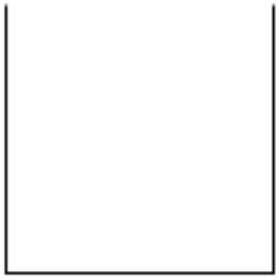
Pen Down

7b. Save and Run

Click: Save ⌘S Click: Run Module F5

See:

```
>>> ========== RESTART ==========
>>>
```



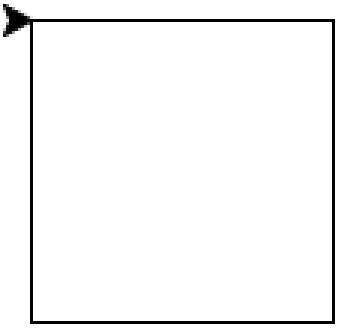
(Turtle Window)

7c. Show Turtle, Pen Down

Type: IDLE/Shell

```
>>>st()
>>>pd()
 >>>fd(100)
 >>>bk(100)
```

See:



8a. Teach Turtle a New Word!

Type:

```
from turtle import *
st()

def sq():
    fd(100)
    rt(90)
    fd(100)
    rt(90)
    fd(100)
    rt(90)
    fd(100)
    rt(90)
```

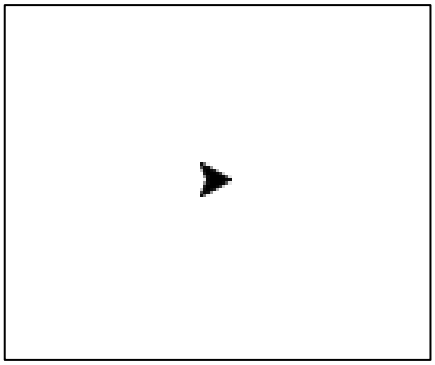
The "dots" matter!

8b. Try Out the New Word

Click: Save ⌘S Click: Run Module F5

>>>

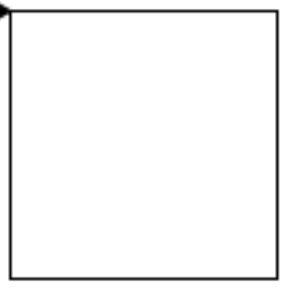
See:



Type - IDLE/Shell

>>>sq()

See



8c. Play with the New Word

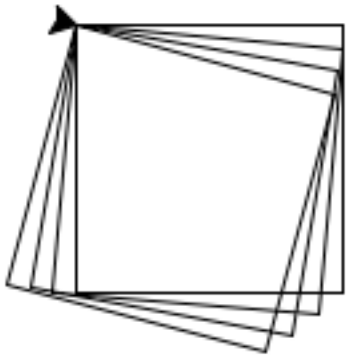
Type - IDLE/Shell

```
>>>clear()

>>>sq()

>>>rt(5)
>>>sq()
>>>rt(5)
>>>sq()
```

See:



9a. Teach Turtle More Words!

Type put after "def sq"

```
def sqspin():
    sq()
    rt(5)
    sqspin()
```

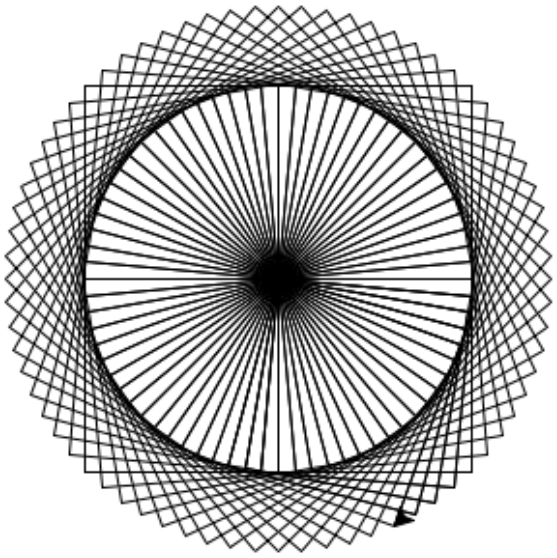
Click: Save ⌘S **Click:** Run Module F5

Type: (IDLE/Shell):

```
>>>sqspin()
```

(Control-C to stop)

See:

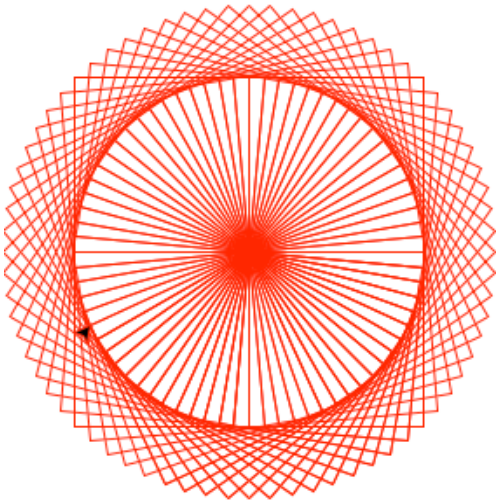


9b. Getting Colorful

Type - IDLE/Shell

```
>>>home()
>>>clear()
>>>pencolor('red')
>>>sqspin()
```

(Control-C to stop)



See:

9c. A List of Colors!

Type - IDLE/Shell

```
>>>colors = ['red', 'green', 'orange']
>>>print(colors[0])
>>>print(colors[1])
>>>print(colors[2])
>>>mycolor = colors[1]
>>>pencolor(mycolor)
```

See:

red

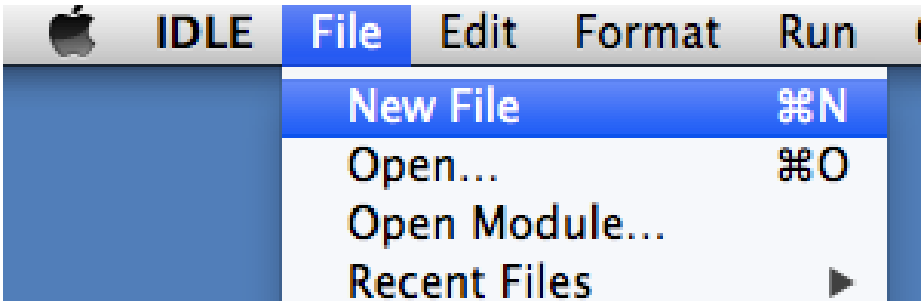
green

orange

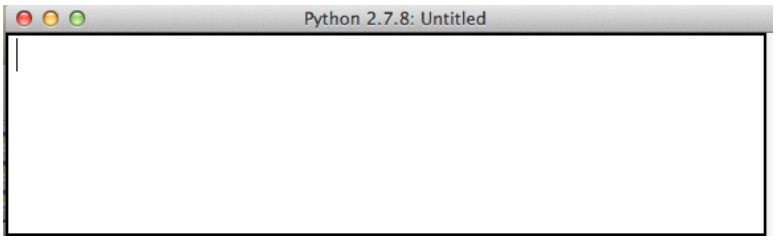
(Turtle Window)

Make New File

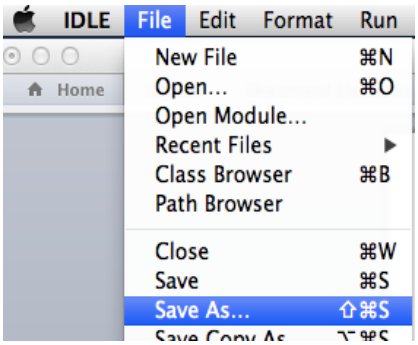
Click:



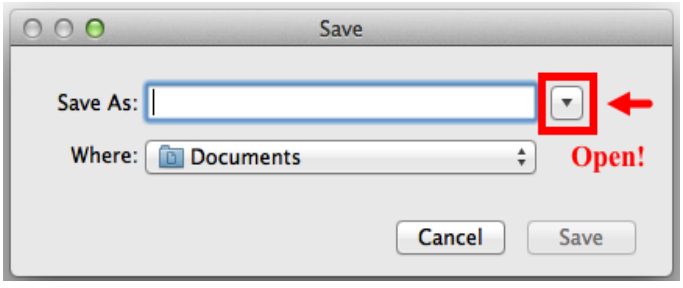
See:



Save As ··· Documents ···



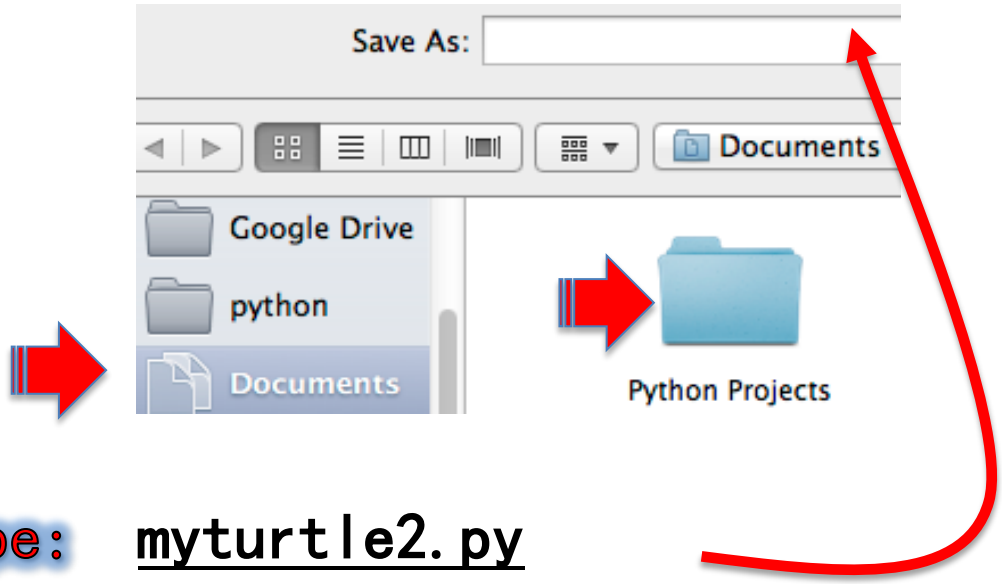
See:



Python Projects/myturtle2.py

Click: Documents

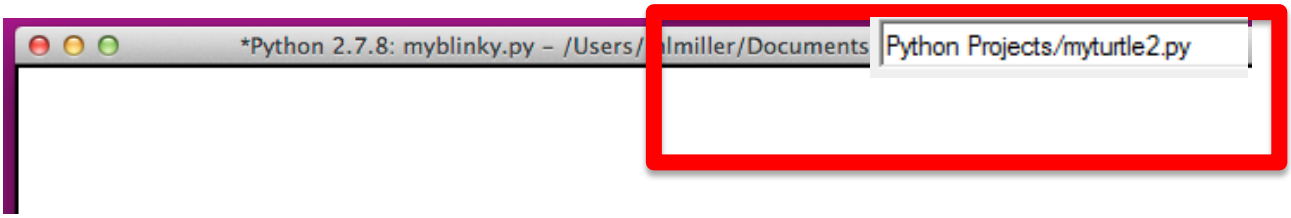
Double-Click: Python Projects



Type: myturtle2.py

Click: Save

See:



10a. Get Random!

Type - Add to Top of Your File:

```
from random import *
colors = ['red','green',
    'yellow','blue','orange']

def pickcolor():
    return choice(colors)
```

Click: Save ⌘S **Click:** Run Module F5

Type - IDLE/Shell

See???

```
>>>pickcolor()
>>>pickcolor()
>>>pickcolor()
```

```
'green'
'blue'
'red'
```

10b. Make a Square Again

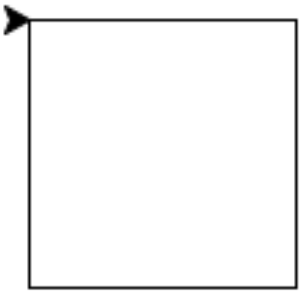
Type:

```
from turtle import *
st()
def sq():
    fd(100)
    rt(90)
    fd(100)
    rt(90)
    fd(100)
    rt(90)
    fd(100)
    rt(90)
```

Type - IDLE/Shell

```
>>>sq()
```

See:



10c. Colorful Spinning Squares

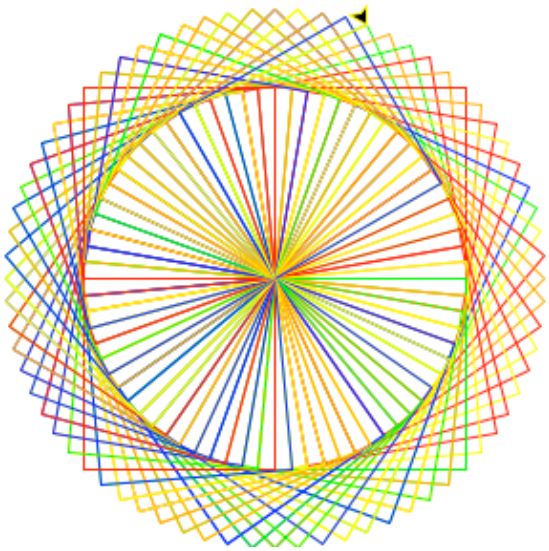
Type:

```
def sqspin():
    sq()
    rt(5)
    mycolor =
    choice( ‘red’ ,’ green’ ,’ blue’ )
    pencolor(mycolor)
    sqspin()
```

Type - IDLE/Shell

```
>>>sqspin()
```

See



(Control-C to stop)

11a. Count Down

Type: Edit sqspin() in your file

```
def sqspin(num):
    print(num)
    sq()
    rt(5)
    mycolor = pickcolor()
    pencolor(mycolor)
    sqspin(num-1)
```

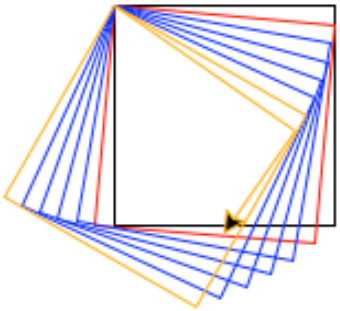
Type IDLE/Shell

```
>>>sqspin(10)
```

Click: Save ⌘S **Click:** Run Module F5

See:

10
9
(Control-C t



8
7
6
5
4
3 ...

11b. Make It Stop!

Type - Edit sqspin(num)

```
def sqspin(num):
    print(num)
    if num > 0:
        sq()
        rt(5)
        mycolor = pickcolor()
        pencolor(mycolor)
        sqspin(num-1)
```

Type - IDLE/Shell

```
>>>sqspin(3)
```

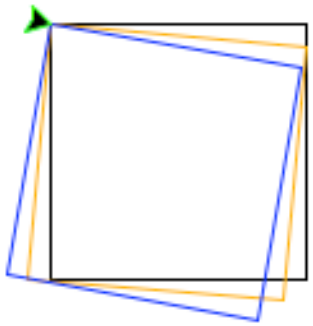
Click: Save ⌘S Click: Run Module F5

See:

3

2

1



12a. Teach Turtle to make Triangle

Type:

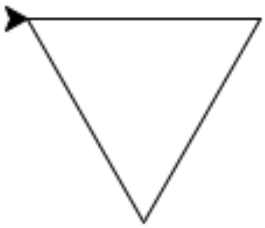
```
def tri():
    fd(100)
    rt(120)   # What happens if rt(60)?
    fd(100)
    rt(120)
    fd(100)
    rt(120)
```

Click: Save ⌘S **Click:** Run Module F5

Type - Shell

```
>>>tri()
```

See:



12b. Teach Turtle to Set Up

Type:

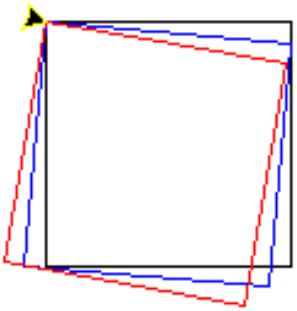
```
def setup():
    home()
    clear()
    st()
    pd()
```

Click: Save ⌘S **Click:** Run Module F5

Type - Shell

```
>>> setup()
>>> sqspin(3)
```

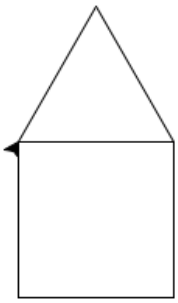
See:



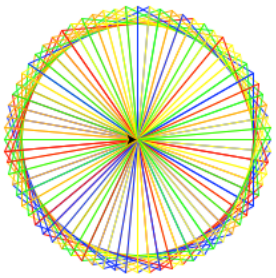
12c. Draw Your Own Picture!

Ideas?

```
#house
setup()
sq()
lt(60)
tri()
```



```
def trispin(ctr):
    if (ctr) > 0:
        tri()
        rt(5)
        pencolor(pickcolor())
        trispin(ctr-1)
```



13a. Try Out *for* Loops

Type IDLE/Shell

```
>>> for i in range(5):
        print(i)
```

Hit <return> key twice

See:

```
0
1
2
3
4
>>>
```

Questions: Why didn't it print **5**? How would you change it to print up to **9**? **10**? Does it matter that the "loop counter" is named ***i***? Could we call it ***j***? ***Santa***?

13b. Another Cool *for* Loop

Type IDLE/Shell

```
>>> for col in ['red', 'blue']:
        print(col)
```

Hit <return> key twice

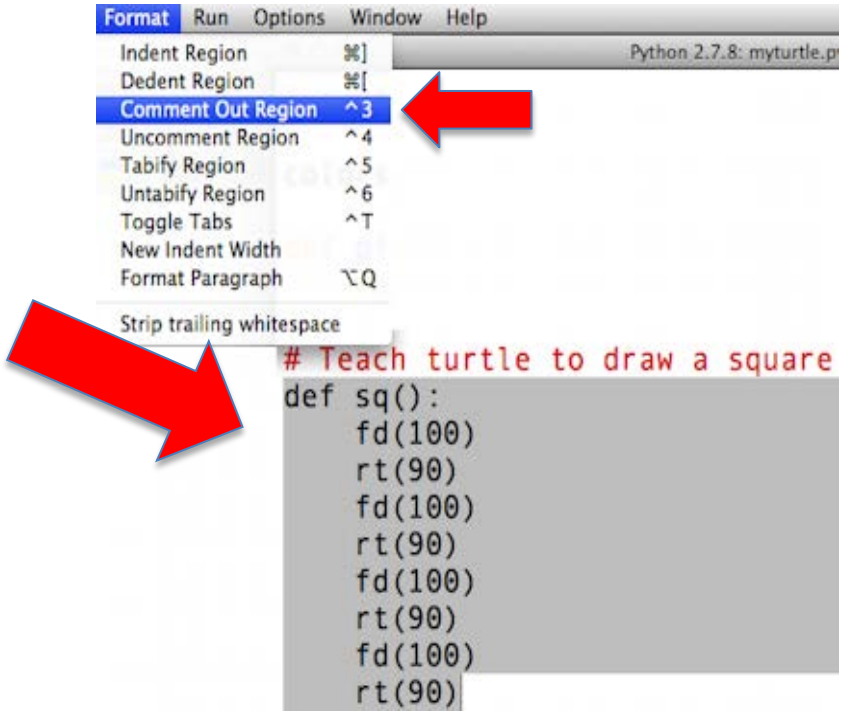
See:

```
red
green
```

13c. Comment Out, *Then* Change

Select definition of sq() with mouse.

Comment Out Region (in *Format* Menu).



See:

```
##def sq():
##    fd(100)
##    rt(90)
##    fd(100)
##    rt(90)
##    fd(100)
```

13d. Improve sq() Using *for* Loop

Type:

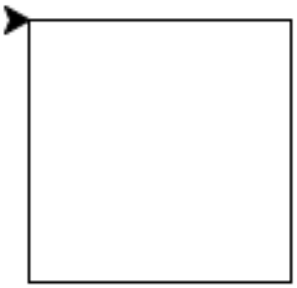
```
def sq():
    for i in range(4):
        fd(100)
        rt(90)
```

14a. Test Improved Version of sq()

Type IDLE/Shell:

```
>>> setup()
>>> sq()
```

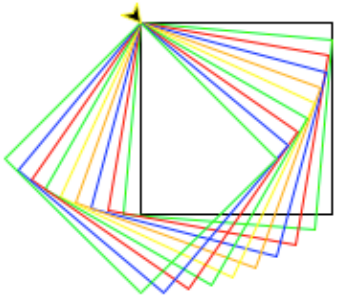
See:



Type IDLE/Shell:

```
>>> sqspin(10)
```

See:



14b. Improve tri() On Your Own

Edit - myturtle.py:

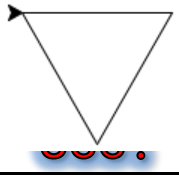
Comment Out current version. Teach turtle new version with for loop. Fill in ?? parts.

```
def tri():
    for santa ?? range(??):
        fd(100)
        rt(??)
```

Click: Save ⌘S **Click:** Run Module F5

Type IDLE/Shell:

```
>>> tri()
```



14c. Big Squares, Little Squares

Comment Out current version of sq():

```
##def sq():
##    for i in range(4):
##        fd(100)
##        rt(90)
```

Teach turtle with an *input* to vary size.

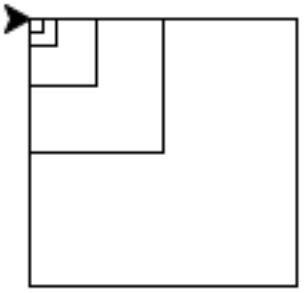
```
def sq(siz):
    for i in range(4):
        fd(siz)
        rt(90)
```

Click: Save ⌘S **Click:** Run Module F5

Type IDLE/Shell:

```
>>> sq(100)
>>> sq(50)
>>> sq(25)
>>> sq(10)
>>> sq(5)
```

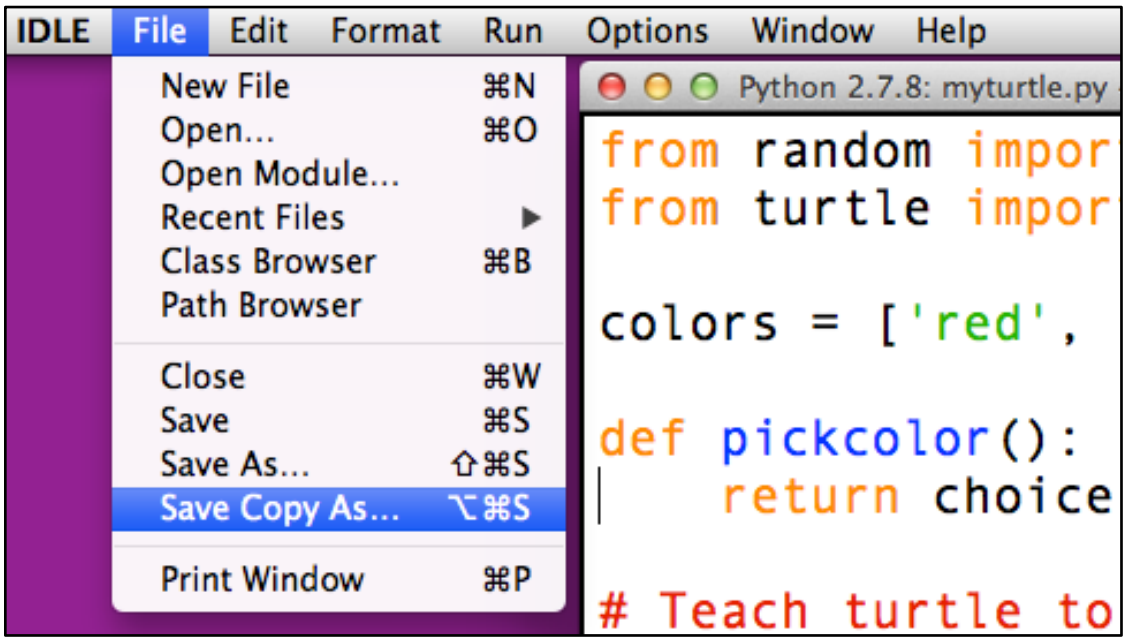
See:

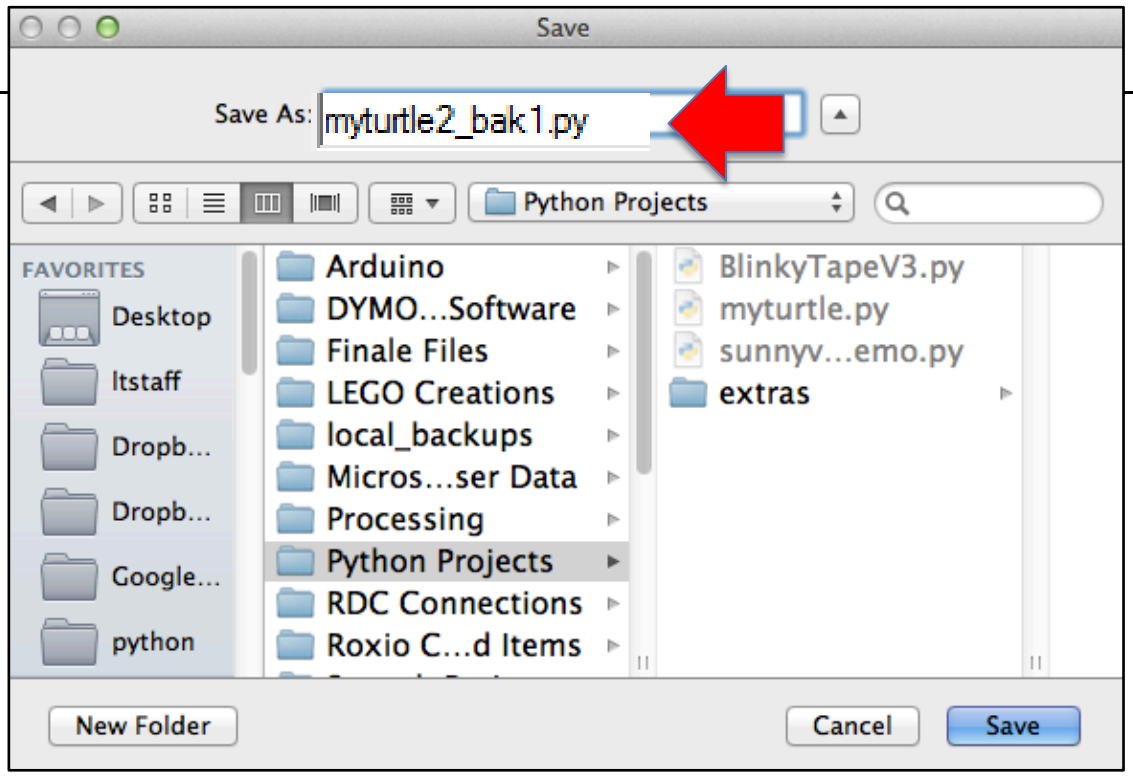


Play around with different sizes, pencolors.

15a. Backup File Versions Often!

In myturtle2.py, CLICK:
File -> Save Copy As: **myturtle2_bak1.py**



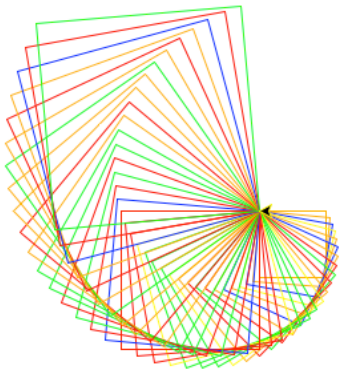


15b. Improve sqspin() for sq(siz)

Edit - myturtle.py:

Comment Out *the current version of sqspin. Teach the turtle a new version, with an input to vary the size, getting bigger each time. Also, skip the* print *statement.*

```
##def sqspin(num):
##    print(num)
##    if num > 0:
##        sq()
##        rt(5)
##        mycolor = pickcolor()
##        pencolor(mycolor)
##        sqspin(num-1)
def sqspin(siz, num):
    if num > 0:
        sq(siz)
        rt(5)
        mycolor = pickcolor()
        pencolor(mycolor)
        sqspin(siz+3, num-1)
```



Click: Save ⌘S Click: Run Module F5

Type - Shell:

```
>>> sqspin(50,36)
```

16a. Improve tri(): Accept Size Input

Comment Out existing tri():

```
##def tri():
##    for santa in range(3):
##        fd(100)
##        rt(120)
```

Type - Teach Turtle Improved Version:

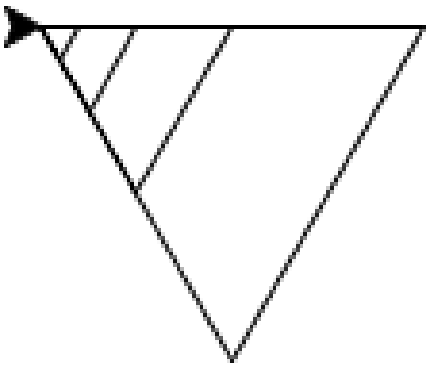
```
def tri(siz):
    for santa in range(3):
        fd(siz)
        rt(120)
```

Click: Save ⌘S **Click:** Run Module F5

Type:

```
>>> tri(100)
>>> tri(50)
>>> tri(25)
>>> tri(10)
```

See:



16b. Teach Turtle trispin(siz, num)

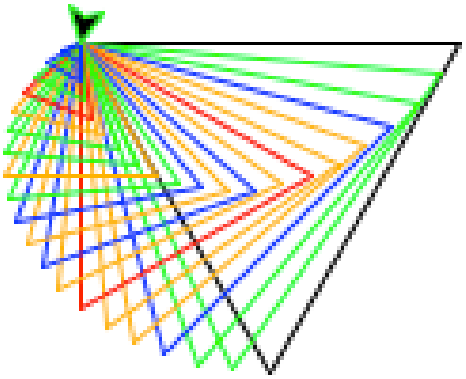
```
def trispin(siz, ctr):
    if ctr > 0:
        tri(siz)
        rt(5)
        pencolor(pickcolor())
        trispin(siz-5, ctr-1)
```

Click: Save ⌘S Click: Run Module F5

Type IDLE/Shell:

```
>>> trispin(100,20)
```

See:



16c. Explore Some Other Inputs

Type - Shell:

```
>>> trispin(??, ??)      # You decide
```

Try making some other designs just using triangles.

17a. Teach Turtle circle(edge)

Type - myturtle.py:

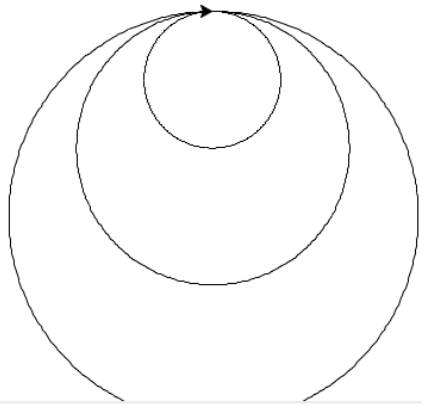
```
def circle(edge):
    for count in range(360):
        fd(edge)
        rt(1)
```

Click: Save ⌘S **Click:** Run Module F5

Type IDLE/Shell:

```
>>> circle(1)
>>> circle(2)
>>> circle(3)
```

See:



17b. Variations on circle(edge)

What would happen to the circle if the right turns were bigger than 1 degree? Would it still draw a circle? Make a copy of circle with a different name to try this.

17c. Teach Turtle About Polygons!

Type - myturtle.py

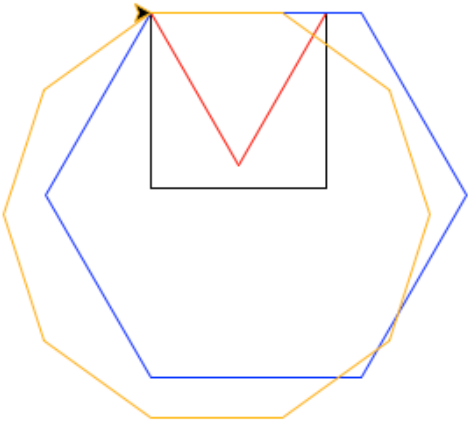
```
def poly(sid,ang,sides):
    for s in range(sides):
        fd(sid)
        rt(ang)
```

Click: Save ⌘S Click: Run Module F5

Type IDLE/Shell:

```
>>> poly(100,90,4)
>>> pencolor('red')
>>> poly(100,120,3)
>>> pencolor('blue')
>>> poly(120,60,6)
>>> pencolor('orange')
>>> poly(75,36,1
```

See:



17d. Teach Turtle to Draw Your Favorite Animal, Using Mainly poly(sid,ang,sides)

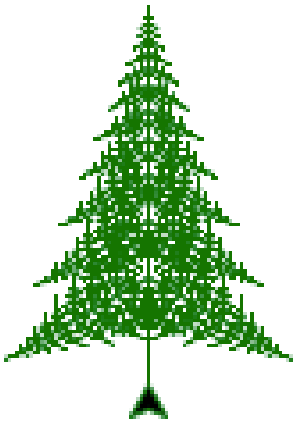
(Free Exploration!)

18a. Turtle Geometry Open Project

Teach the turtle a new word to draw any picture you like. This is an *open project* of your choosing. Below are a few ideas related to *Holiday Amusement Parks*! Please *be creative* -- make your own!

There is no lesson # 19. Begin the Blinky Tape programs starting with #20.

18b. Christmas Tree Project



```
n = 15
speed("fastest")
lt(90)
fd(3*n)

pencolor("dark green")
bk(n*4.8)

def tree(d, s):
    if d <= 0: return
    fd(s)
    tree(d-1, s*.8)
    rt(120)
    tree(d-3, s*.5)
    rt(120)
    tree(d-3, s*.5)
    rt(120)
    bk(s)

tree(15, n)
bk(n/2)
```

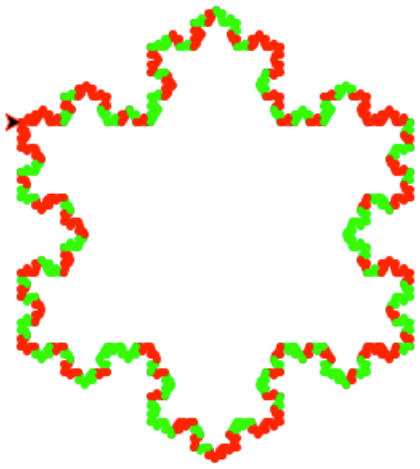
(Credit --http://codegolf.stackexchange.com/questions/15860/make-a-scalable-christmas-tree)

18c. Star of David Project

```
def david(siz):
    pensize(10)
    pencolor("blue")
    tri(siz)
    pu()
    lt(90)
    bk(2*siz/3)
    rt(30)
    pd()
    tri(siz)
    ht()
```

18d. Snow Flake Project

```
def snow(siz, cntr):
# Repeat this 3 times with rt(120) in between
    if cntr < 1:
        fd(siz)
    else:
        pencolor(choice(['red', 'green']))
        snow(siz/3, cntr-1)
        lt(60)
        snow(siz/3, cntr-1)
        rt(120)
        snow(siz/3, cntr-1)
        lt(60)
        snow(siz/3, cntr-1)
```



There is no lesson # 19. Begin the Blinky Tape programs starting with #20.