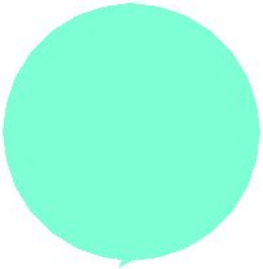
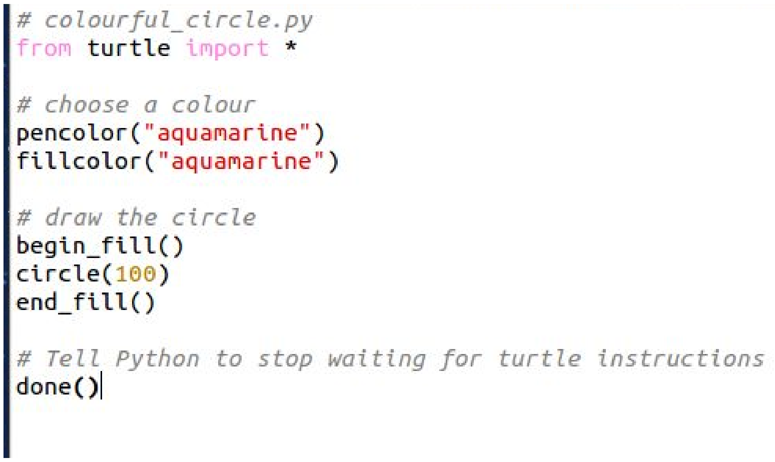
Circle:



Code to get it



```
# colourful_circle.py
from turtle import *

# choose a colour
pencolor("aquamarine")
fillcolor("aquamarine")

# draw the circle
begin_fill()
circle(100)
end_fill()

# Tell Python to stop waiting for turtle instructions
done()
```

Well done! :) Things you can try now:

- Change the color

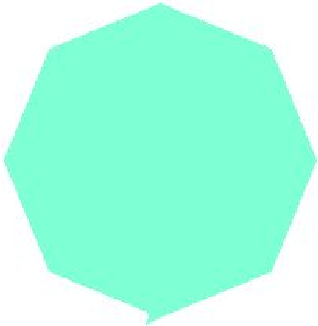
  Hint: Look for where it says pencolor and fillcolor.

- Change the size

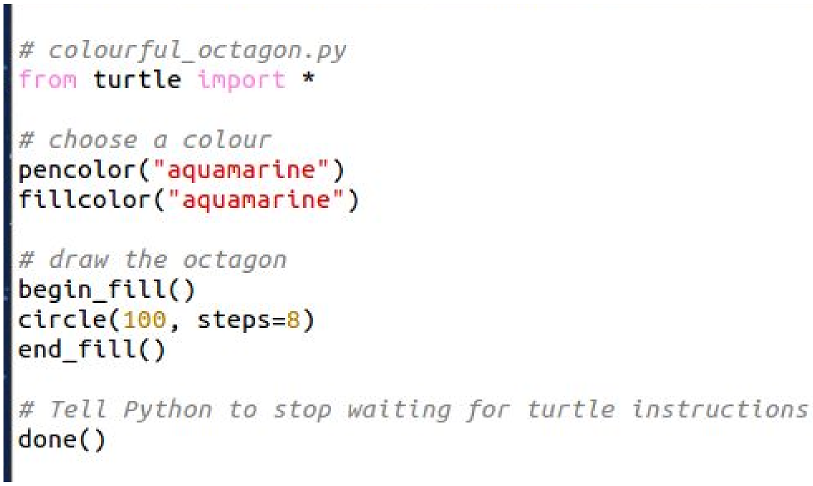
  Hint: Look where it draws the circle.

Octagon:

The **circle()** command is a turtle function that treats a circle as a polygon with many sides. The **steps** argument tells how accurate to be - how many straight lines to make the circle from. So the octagon is a “circle” with 8 steps.



Code to get it



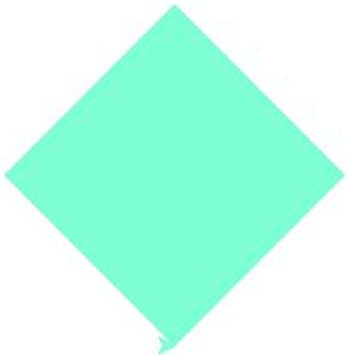
```
# colourful_octagon.py
from turtle import *

# choose a colour
pencolor("aquamarine")
fillcolor("aquamarine")

# draw the octagon
begin_fill()
circle(100, steps=8)
end_fill()

# Tell Python to stop waiting for turtle instructions
done()
```

Well done! :) Things you can try now:

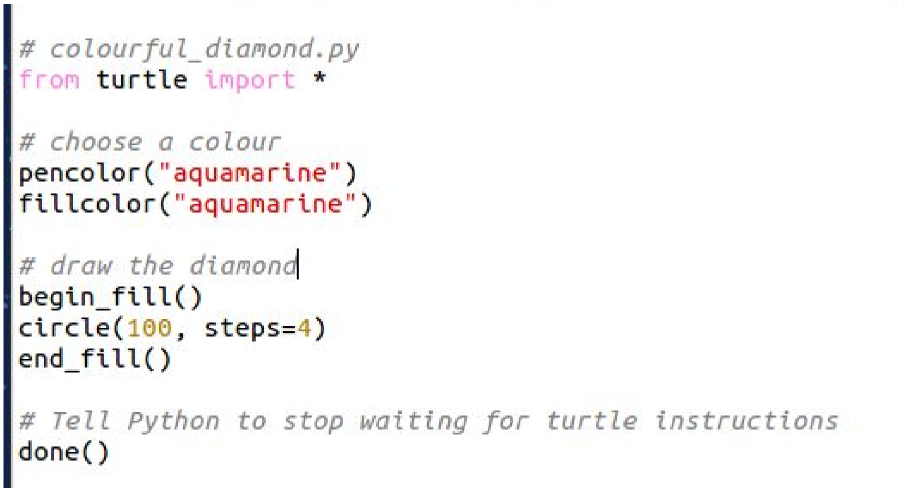
- Change the color - Hint: Look for where it says pencolor and fillcolor.
- Change the size - Hint: Look where it draws the “circle”.
- Change the number of steps - Hint: Look where it draws the “circle”.

Diamond:

The **circle()** command is a turtle function that treats a circle as a polygon with many sides. The **steps** argument tells how accurate to be - how many straight lines to make the circle from. So the diamond is a “circle” with 4 steps.



Code to get it

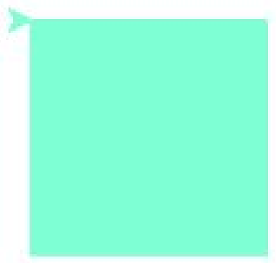


```
# colourful_diamond.py
from turtle import *

# choose a colour
pencolor("aquamarine")
fillcolor("aquamarine")

# draw the diamond
begin_fill()
circle(100, steps=4)
end_fill()

# Tell Python to stop waiting for turtle instructions
done()
```

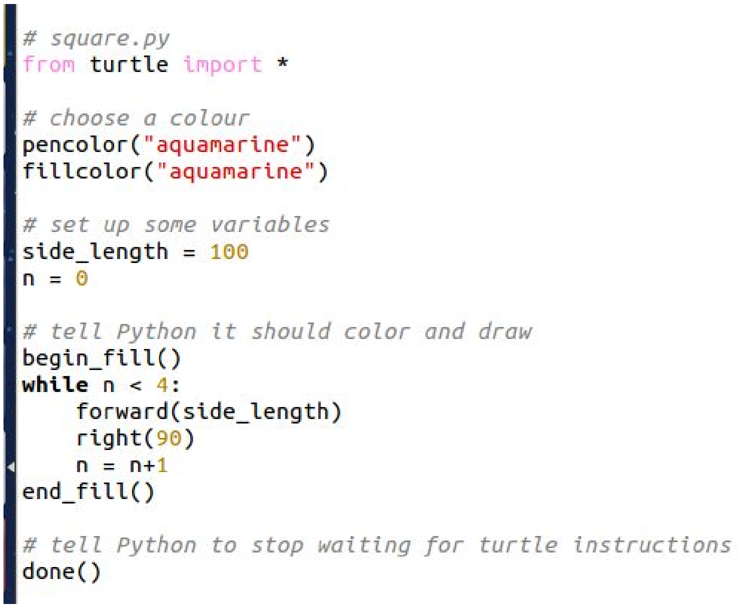
Well done! :) Things you can try now:

- Change the color - Hint: Look for where it says pencolor and fillcolor.
- Change the size - Hint: Look where it draws the “circle”.
- Change the number of steps - Hint: Look where it draws the “circle”.

Square:



Code to get it



```
# square.py
from turtle import *

# choose a colour
pencolor("aquamarine")
fillcolor("aquamarine")

# set up some variables
side_length = 100
n = 0

# tell Python it should color and draw
begin_fill()
while n < 4:
    forward(side_length)
    right(90)
    n = n+1
end_fill()

# tell Python to stop waiting for turtle instructions
done()
```

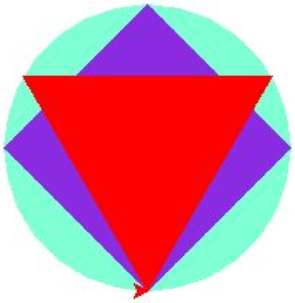
Well done! :) Things you can try now:

- Change the color - Hint: Look for where it says pencolor and fillcolor.
- Change the size - Hint: Look where it sets the variable side_length.
- What does it happen if you change 90 for another number?
- What does it happen if you change 4 for another number?

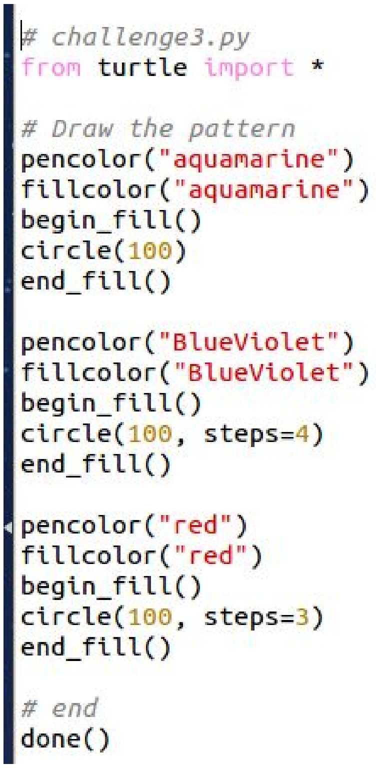
n + 1

- The step is **very** important, can you tell why?

Awesome stuff #1:



Code to get it



```
# challenge3.py
from turtle import *

# Draw the pattern
pencolor("aquamarine")
fillcolor("aquamarine")
begin_fill()
circle(100)
end_fill()

pencolor("BlueViolet")
fillcolor("BlueViolet")
begin_fill()
circle(100, steps=4)
end_fill()

pencolor("red")
fillcolor("red")
begin_fill()
circle(100, steps=3)
end_fill()

# end
done()
```

Well done! :) Things you can try now:

- Change the colors/sizes/steps
- Add more figures!