



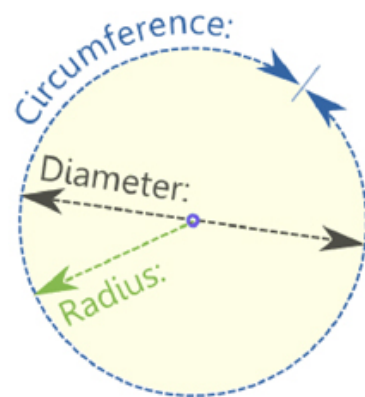
## Lesson 3: Main Circle Commands

### Summary:

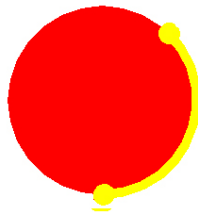
Code Instruction	What it does
<code>t.circle(radius)</code>	Draw a circle with the given radius value. Radius value can be positive or negative, depends on the direction of the turtle motion.
<code>t.circle(radius, arc)</code>	Arc option allows to draw part of the circle. If <code>arc=360</code> turtle draws all 360 degree circle, if <code>arc=180</code> , turtle draws half of the circle, if <code>arc=90</code> → quarter of the circle. Arc value can be positive or negative, depends on the motion direction.
<code>t.circle(radius, arc, steps)</code>	With option <code>steps</code> (integer value) circle is approximated by an inscribed regular polygon, <code>steps</code> determine the number of steps to use. May be used to draw regular polygons. <code>Step=3</code> means that we draw triangle, <code>step=4</code> corresponds square, <code>step=5</code> → pentagon...

### A few tips:

- a. A **circle** is a shape that is made up of a curved line. It's round, and all points on the curved line are an equal distance from the center point. The circumference is the distance around a circle





- b. The arc of a circle is a portion of the circumference of a circle specified by yellow colour.



Arc value is measured in degrees and if arc=360 turtle draws all 360 degree circle, if arc=180, turtle draws half of the circle, if , for example, arc=90→ quarter of the circle. Arc value can be positive or negative, depends on the motion direction.

# Python + Math

## Code & Output for Kids



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('red')
t.circle(100)
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('red','blue')
t.begin_fill()
t.circle(100)
t.end_fill()
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('blue')
t.circle(100,90)
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold','orange')
t.begin_fill()
t.circle(100,270)
t.end_fill()
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold','orange')
t.begin_fill()
t.circle(100,180)
t.end_fill()
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(45)
t.color('gold','green')
t.begin_fill()
t.circle(100,180)
t.end_fill()
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('green','yellow')
t.begin_fill()
t.circle(100,360,7)
t.end_fill()
```


```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(55)
t.color('gold','green')
t.begin_fill()
t.circle(100,360,3)
t.end_fill()
```

```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('green','yellow')
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
t.color('pink','violet')
t.left(180)
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
```

### LESSON 3

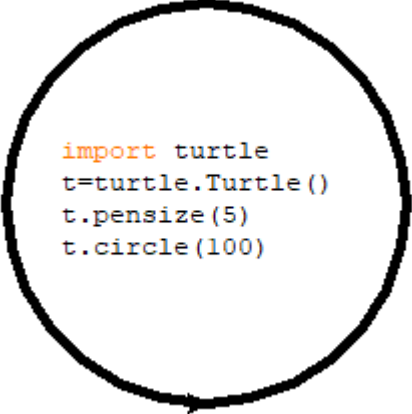
To see examples, images, and challenges

[www.python.kidsgo.ca](http://www.python.kidsgo.ca)

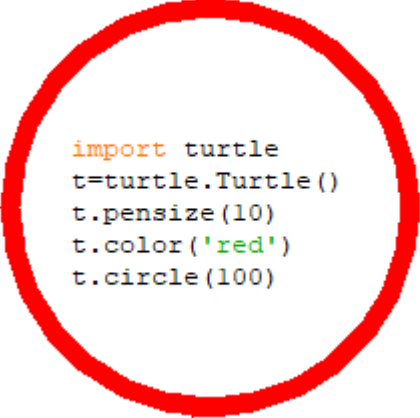


1. **Example #1** (Draw circle)

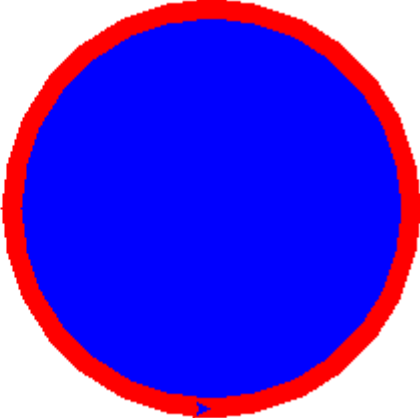
```
import turtle
t=turtle.Turtle()
t.pensize(5)
t.circle(100)
```



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('red')
t.circle(100)
```

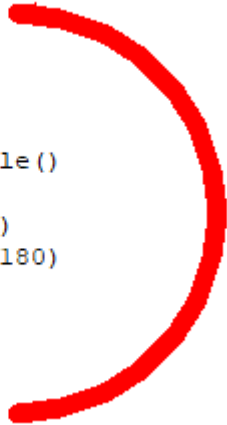


```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('red','blue')
t.begin_fill()
t.circle(100)
t.end_fill()
```

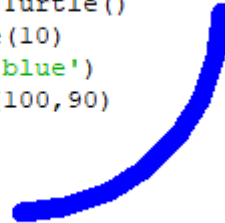


2. **Example #2** (Draw part of the circle). We use arc option

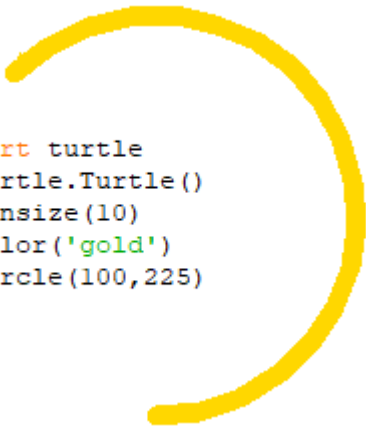
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('red')
t.circle(100,180)
```



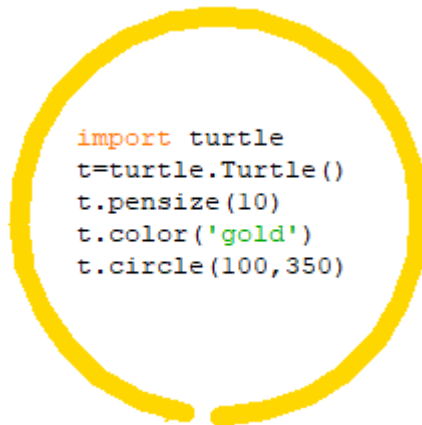
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('blue')
t.circle(100,90)
```



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold')
t.circle(100,225)
```



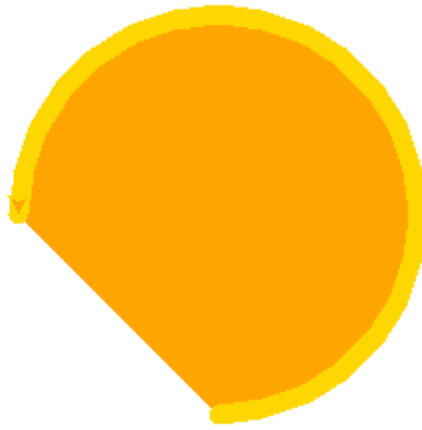
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold')
t.circle(100,350)
```



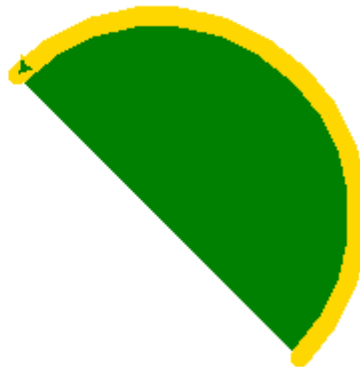
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold','orange')
t.begin_fill()
t.circle(100,180)
t.end_fill()
```



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('gold','orange')
t.begin_fill()
t.circle(100,270)
t.end_fill()
```



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(45)
t.color('gold','green')
t.begin_fill()
t.circle(100,180)
t.end_fill()
```



All images from Example #3 are drawn with **circle** code (turtle can draw different polygons as triangle, square, pentagon, hexagon...). To create it we use an option step, specified in summary. Again code circle(radius, arc, steps) has three options: option 1 → radius value; option 2 → arc value; option 3 → step value. If you use only option 1, turtle draws full circle with radius specified by radius value; if you use two options (option 1 and 2) turtle draws part of the circle; and when you use three options (first radius value, second 360-degree value, and third option 3, or 4, or 5 or 6....) turtle draws polygon instead of circle.

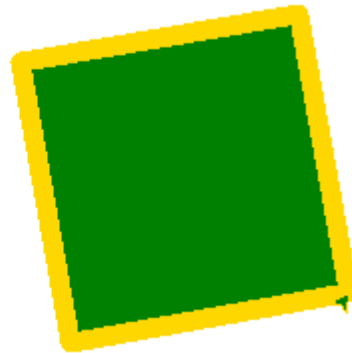
### 3. Example #3(Draw inscribed into the circle regular polygon).

```
import turtle
t=turtle.Turtle()
t.pensize(10)

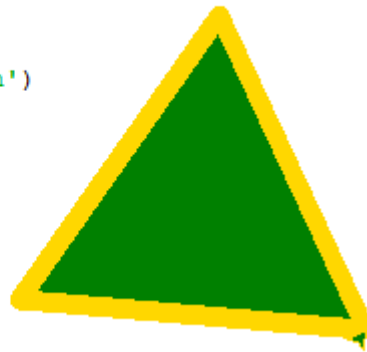
t.color('gold','green')
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
```



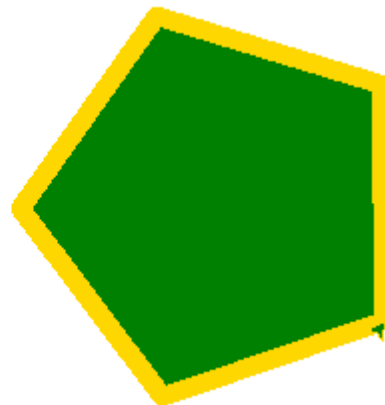
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(55)
t.color('gold','green')
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
```



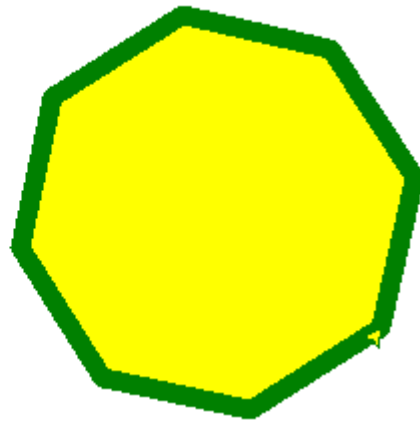
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(55)
t.color('gold','green')
t.begin_fill()
t.circle(100,360,3)
t.end_fill()
```



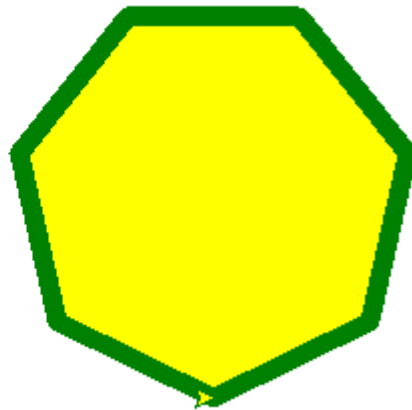
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(55)
t.color('gold','green')
t.begin_fill()
t.circle(100,360,5)
t.end_fill()
```



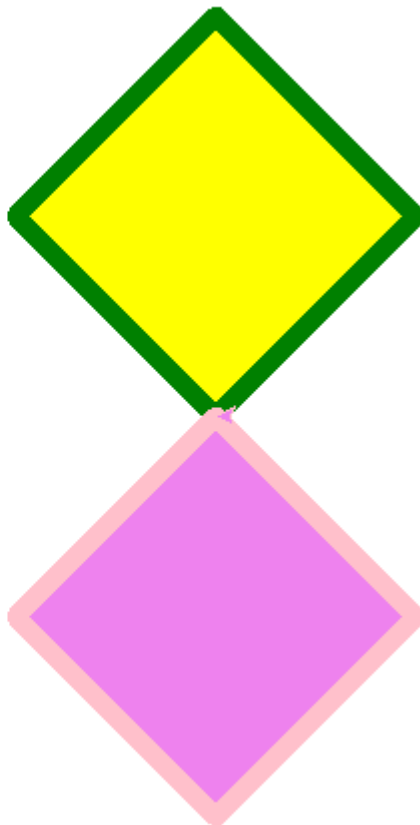
```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.left(55)
t.color('green','yellow')
t.begin_fill()
t.circle(100,360,8)
t.end_fill()
```



```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('green','yellow')
t.begin_fill()
t.circle(100,360,7)
t.end_fill()
```

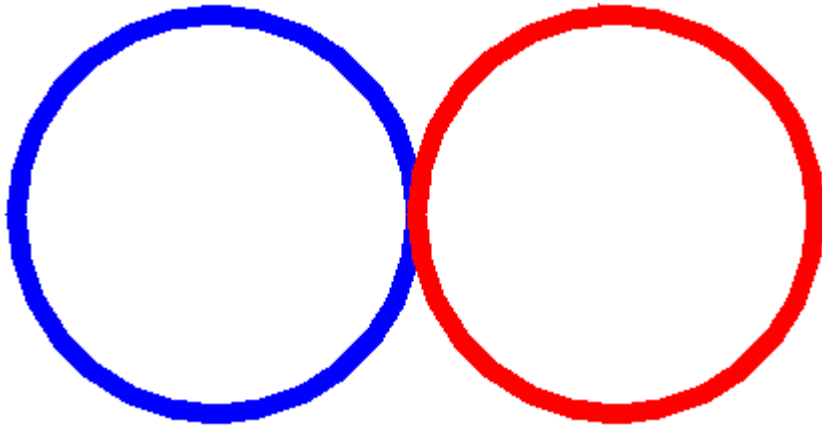


```
import turtle
t=turtle.Turtle()
t.pensize(10)
t.color('green','yellow')
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
t.color('pink','violet')
t.left(180)
t.begin_fill()
t.circle(100,360,4)
t.end_fill()
```

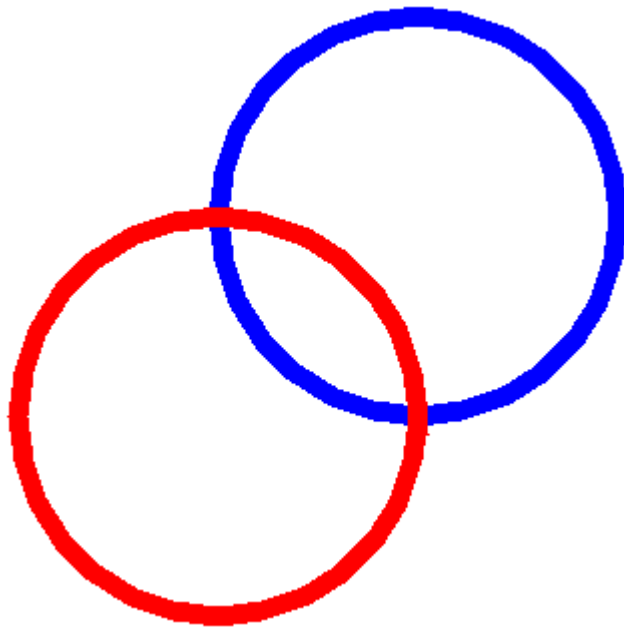


Challenges: write codes to create the following geometry shapes with circle code:

1. Expected output

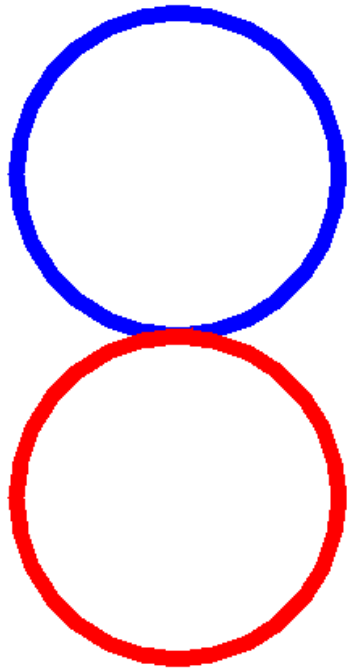


2. Expected output

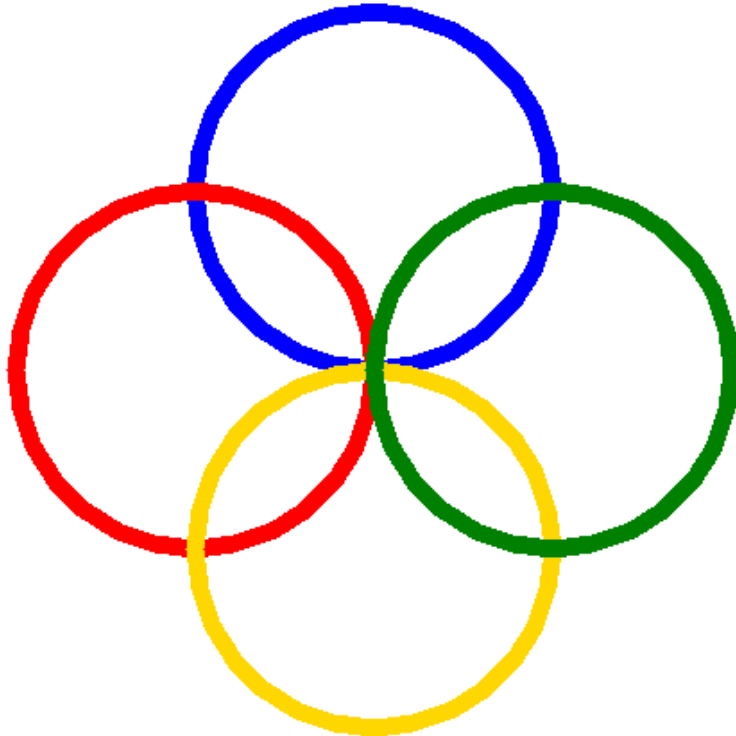




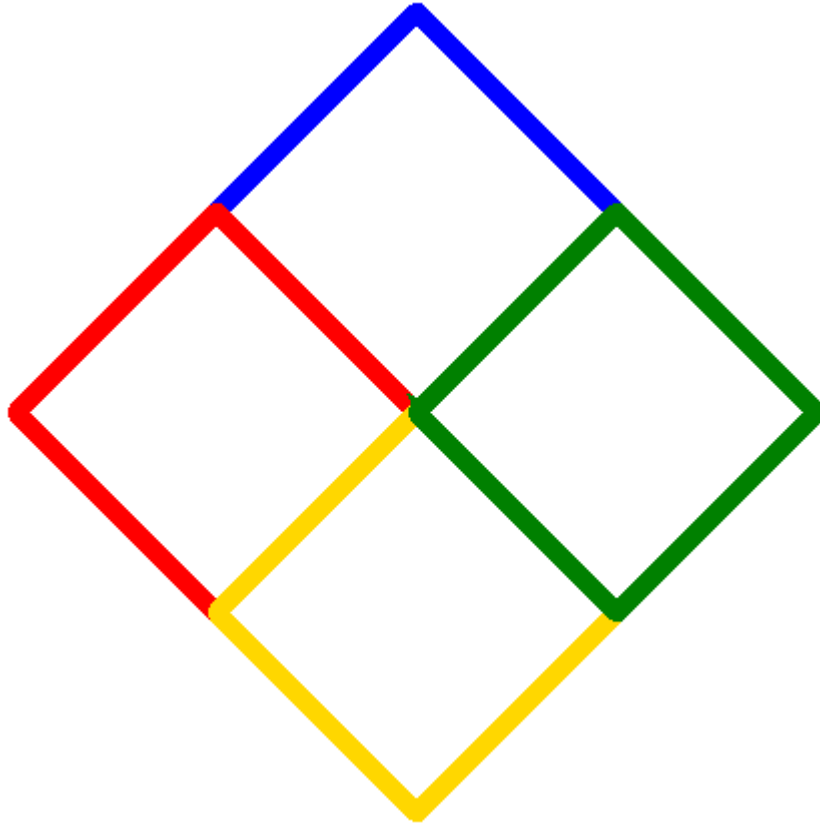
3. Expected output



4. Expected output



5. Expected output



6. Expected output

