

Must be in the program:

- `import turtle` – **first line of program** – allows you to use the turtle commands that have been written for you (those turtle commands are below)
- `wn=turtle.Screen` – **second line of program** - gives the turtle a place to draw
- `wn.exitonclick()` – **last line of program** – causes your drawing to stay open until you click the mouse in the window (otherwise it goes away too fast for you to see what you have done)
- `t = turtle.Turtle()` - creates a turtle with the name indicated (t in this example). You can have more than one turtle in your program. Each of the commands below used by saying the that name and a dot before the command so it knows which turtle should do that action

Positioning:

- `t.setposition(x,y)` - move the turtle named t to position (x,y) in the Cartesian coordinate system The center of the screen is the default location and is (0,0)
- `t.forward(distance)` - how many pixels to move the turtle named t forward
- `t.backward(distance)` - how many pixels to move the turtle named t backward

Direction:

- `t.left(angle)` - turn turtle t to the left an amount given by the angle
- `t.right(angle)` - turn turtle t to the right
- `t.setheading(angle)` - sets the turtle t's direction to be the angle given
  - (0=east, 90=north, 180=west, 270=south)

The pen:

- `t.penup()` - pick up the pen so future movements do not draw anything
- `t.pendown()` - put the pen down so that future movements show results (the pen is down by default)
- `t.pencolor(colourname)` - change the color of the pen to the color whose name is given as a string. EG: "red" or "blue"
- `t.pensize(number)` - change the size of the pen to a number:1-10

black	cyan	green	pink	violet
blue	gold	magenta	purple	white
brown	gray	orange	red	yellow

Predefined Figure:

- `t.circle(radius)` - draw a circle to the left of the turtle with the given radius. The turtle ends up in the same location, pointing in the same direction.

Filling:

- `t.begin_fill()` - what we are about to draw should be filled in
- `t.end_fill()` - do not fill in after this (when paired with `begin_fill`)
- `t.fillcolor(colourname)` – change the color of the filling to the color whose name is given as a string. EG: "red" or "blue" (same colors as above)

The Turtle

- `t.shape(shapename)` - what we are about to draw should be filled in "arrow", "turtle", "circle", "square", "triangle", "classic"
- `t.shapesize(number)` - what we are about to draw should be this size number:1-10
- `t.stamp()` - stamps the shape of the turtle at the current location
- `t.color(colourname)` – change the color of the stamp to the color whose name is given as a string. EG: "red" or "blue" (same colors as above)

Random Numbers

- `include random`- must be included at the top right after the `include turtle`
- `variable = random.randrange(low,high)`- gives the variable named on the left a random integer that is between the low and the high values indicated (any value previously held in the variable named on the left of the equals sign will be lost)

Looping

```
for variable in range (low, high):  
    do these tasks
```

The variable named will take on all values from low to high-1 in sequence.  
For each of those values, the list of tasks indented below will be done

### Example showing Lines

```
import turtle
wn=turtle.Screen()
t = turtle.Turtle()

# moves to the correct place
t.penup()
t.setposition(-200,100)
t.pencolor("black")
t.pensize(3)
t.pendown()
#makes a few lines
t.forward(30)
t.right(50)
t.forward(70)
t.left(300)
t.forward(100)
t.left(40)
t.forward(50)
t.right(20)
t.backward(60)

wn.exitonclick()
```



### Example showing Stamping

```
import turtle
wn=turtle.Screen()
t = turtle.Turtle()

t.shape("square")
t.color("black")
t.shapesize(30)
t.setposition(0,0)
t.setheading(0)
t.stamp()

t.color("yellow")
t.shapesize(10)
t.stamp()

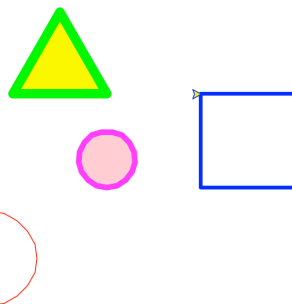
t.shape("turtle")
t.penup()

t.color("red")
t.shapesize(4)
t.setposition(100,150)
t.setheading(90)
t.stamp()

t.color("blue")
t.shapesize(7)
t.setposition(-100,150)
t.setheading(270)
t.stamp()

t.color("magenta")
t.shapesize(2)
t.setposition(0,-20)
t.setheading(0)
t.stamp()

wn.exitonclick()
```



### Example showing Several shapes

```
import turtle
t = turtle.Turtle()
wn=turtle.Screen()

# moves to the correct place
t.penup()
t.setposition(-125,-125)
t.pencolor("red")
t.pensize(1)
t.pendown()
#makes a red circle at that place
t.circle(50)

# moves to the correct place
t.penup()
t.setposition(0,0)
t.pencolor("magenta")
t.pensize(6)
t.pendown()
#makes a magenta circle filled
with pink at that place
t.begin_fill()
t.fillcolor("pink")
t.circle(30)
t.end_fill()

# moves to the correct place
t.penup()
t.setposition(-100,100)
t.setheading(0)
t.pencolor("green")
t.pensize(10)
# makes a green triangle filled
with yellow at that place
t.begin_fill()
t.fillcolor("yellow")
t.pendown()
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
t.forward(100)
t.end_fill()

# moves to the correct place
t.penup()
t.setposition(100,100)
t.setheading(0)
t.pencolor("blue")
t.pensize(4)
# makes a blue square at that
place
t.pendown()
for side in range(0,4):
    t.forward(100)
    t.right(90)

wn.exitonclick()
```

