Turtles are like cursors used to indicate the position for drawing to the screen. Turtles are used for drawing on the screen. Most of the methods for controlling turtles are below.

Turtle methods

t = turtle.Turtle() create a new Turtle object and open its window

Position

t.home()	move the turtle to (0, 0), pointing east (depending on mode)
t.position()	return the current position (x,y)
t.pos()	
t.setposition(x,y)	move to coordinates (x,y)
t.teleport(x,y=None,fill_gap=False)	move to coordinates (x,y) without drawing a line. If fill_gap is True, it fills the gap from the old position to the new one.
t.setpos(x,y)	
t.goto(x,y)	
t.xcor()	Return the turtle's x coordinate.
t.ycor()	Return the turtle's y coordinate.
t.setx(x)	set the turtle's x coordinate, leave y unchanged
t.sety(y)	set the turtle's y coordinate, leave x unchanged
t.distance(x, y)	Return the distance from the turtle to coordinates (x, y)

Direction

t.left(angle)	turn left, anticlockwise, angle degrees
t.lt(angle)	
t.right(angle)	turn right, clockwise, angle degrees
t.rt(angle)	
t.setheading(to_angle)	change heading to direction angle degrees
t.seth(to_angle)	
t.heading()	return the current direction angle
t.towards(x, y)	return the angle between the line from turtle position to position specified by (x,y), the vector or the other turtle.

Movement

t.speed(n)	how fast the turtle moves (n = 0 to 10); speed=0 means no animation
t.forward(distance)	move in the current direction distance pixels
t.fd(distance)	
t.backward(distance)	move in the reverse direction distance pixels
t.back(distance)	
t.bk(distance)	

Drawing

t.penup()	raise the pen, stoping drawing
t.up()	
t.pu()	
t.pendown()	lower the pen for drawing
t.down()	
t.pd()	
t.isdown()	return True if the pen is down, False if up.
t.pensize(width),	return or set linewidth to width pixels
t.width(width)	
t.pen()	return or set turtle characteristics
t.circle(radius, extent, steps)	draw a circle with given radius, of arc angle extent. in a number of steps (for a poylgon).
t.dot(size, color)	draw a dot at the current position.
t.stamp()	draw the turtle shape at the current position and return a stamp_id for that stamp.
t.clearstamp(stampid)	delete stamp given by stamp_id.
t.clearstamps(n)	delete n stamps.
t.shape(name)	Return or set turtle shape; "arrow", "turtle", "circle", "square", "triangle", "classic".
t.write(str, move, align, font)	write a message to the screen

Drawing colour

t.color(pencolor, fillcolor)	change the pencolor and fillcolor
t.pencolor(color)	change the pen colour
t.fillcolor(color)	change the fill colorur

Filling

t.begin_fill()	The starting point is remembered for a filled polygon.
t.end_fill()	Current fill color is filled after closing the polygon.
t.filling()	Return True if begin_fill has been called or False if not, or if end_fill has been called.

Clear drawing

t.clear()	Clear the turtle's drawing.
t.reset()	Clear the turtle's drawing, place it at home and return it to it's default settings.
t.undo()	Undo repeatedly the last turtle actions

Turtle Visibility

t.hideturtle()	Make the turtle invisible to speed up the drawing.
t.ht()	
t.showturtle()	Make the turtle visible.
t.st()	
t.isvisible()	Return True if the Turtle is shown, False if it's hidden.