Introduction to Programming

Introduction to Turtle Graphics

Run this:

```python
from turtle import *

color('red', 'yellow')
begin_fill()
while True:
    forward(200)
    left(170)
    if abs(pos()) < 1:
        break
end_fill()
done()
```

Turtle commands

The commonly used commands available in `turtle` are given below. Click on any command to learn more about it.

- `degrees()`
- `radians()`
- `reset()`
- `clear()`
- `tracer(flag)`
- `forward(distance)`
- `backward(distance)`
- `left(angle)`
- `right(angle)`
- `up()`
- `down()`
- `width(width)`
- `color(*args)`
- `begin_fill()`
- `end_fill()`
- `setheading(angle)`
- `window_width()`
- `window_height()`
- `position()`
- `setx(xpos)`
- `sety(ypos)`
- `goto(x,y)`
- `heading()`

### degrees()

Sets the angle input method to degrees. All following angle inputs are assumed to be degree measures. This is the default setting.

### radians()

Sets the angle input method to radians. All following angle inputs are assumed to be radian measures.

### reset()

Resets everything to the default values, and clears the canvas. After a call to `reset`, the canvas will be in exactly the same state as it was when the import command was called: you will have a blank canvas with the turtle (colored black with fill set to unfilled) pointing to the right at the center (heading = 0.0).
clear()

Erases the entire canvas and redraws the turtle. Does not move the turtle.

tracer(n=None, delay=None)

Turns turtle animation on/off and set delay for update drawings.

If non-negative integer n is given, only each n-th regular screen update is performed. Can be used to accelerate the drawing of complex graphics. When called without arguments, returns the currently stored value of n. Second argument sets delay value (see delay()).

Turning the turtle off makes the turtle disappear and makes drawing MUCH faster. Drawing commands are still executed without the turtle, and lines are still drawn when the turtle is moved. Use up and down to turn drawing on and off, or just use the setx, sety, or goto functions to move without drawing.

forward(distance)

Moves the turtle forward distance, drawing a line behind the turtle. The line will be drawn even if the turtle is turned off.

backward(distance)

Moves the turtle backward distance, drawing a line along the path taken. The line will be drawn even if the turtle is turned off.

left(angle)

Turns the turtle left by angle. If degrees has been called (the default), angle will be used as a degree measure; if radians has been called, angle will be interpreted as a measure in radians.

right(angle)

Turns the turtle right by angle. If degrees has been called (the default), angle will be used as a degree measure; if radians has been called, angle will be interpreted as a measure in radians.

up()

Stops all drawing. Until down is called, nothing will be drawn to the screen. Cursor movement will still take effect, however.

down()

Resumes drawing after a call to up. Commands between the up and down statements will not be drawn, but commands after the down statement will appear as normal.

width(width)

Sets the width of the line drawn using the forward and backward commands.
**color(*args)**

Changes the current color. The current color is used for drawing lines using `forward` and `backward`, as well as for filling shapes when `end_fill()` is called after `begin_fill()`. The color can be given as a single color string (as in `color("blue")`, `color("chocolate")`, `color("peru")`, `color("#a0df00")`, or `color("#1dead1")`). A three-tuple of rgb float values (as in `color((0.1,0.5,0.9))` or `color((95/255., 12/255., 9/255.))`) can be used.

**begin_fill()**

Used to fill shapes. First, call `begin_fill()`, then proceed to draw the outline of the shape to be filled. After the shape is done, call `end_fill()`. A line will be drawn from the current turtle position to the position of the turtle when the `begin_fill()` command was called, and the resulting polygon will be filled with the current color (the color of exterior lines will also be changed). If any interior angle in the resulting polygon is greater than 180°, however, the resulting filled polygon will only include the first two line segments after the `begin_fill()` statement, forming a triangle.

**end_fill()**

Used to fill shapes. First, call `begin_fill()`, then proceed to draw the outline of the shape to be filled. After the shape is done, call `end_fill()`. A line will be drawn from the current turtle position to the position of the turtle when the `begin_fill()` command was called, and the resulting polygon will be filled with the current color (the color of exterior lines will also be changed). If any interior angle in the resulting polygon is greater than 180°, however, the resulting filled polygon will only include the first two line segments after the `begin_fill()` statement, forming a triangle.

**heading()**

Returns the current heading of the turtle, in degrees counterclockwise from horizontal right. If `radians` has been called, the measure will be in radians.

**setheading(angle)**

Sets the heading of the turtle to `angle`. Whether `angle` is interpreted as degrees or radians depends on whether `radians` or `degrees` has been called most recently.

**window_width()**

Returns the width of the current window, in pixels.

**window_height()**

Returns the height of the current window in pixels.

**position()**

Returns the position of the turtle as a 2-element list. Coordinates are relative to the origin, which by default is in the middle of the window.
**setx(xpos)**

Changes the x-coordinate of the turtle to xpos. This moves the turtle horizontally and draws a line from the beginning to the end of the movement. The movement is relative to the coordinate axis, not the current position of the turtle. The turtle's heading is unchanged.

**sety(ypos)**

Changes the y-coordinate of the turtle to ypos. This moves the turtle vertically and draws a line from the beginning to the end of the movement. The movement is relative to the coordinate axis, not the current position of the turtle. The turtle's heading is unchanged.

**goto(x, y)**

Moves the turtle from the current position to the location x, y along the shortest linear path between the two locations (i.e. a direct line between the current position and (x,y)). It draws a line behind the turtle along the path taken. The movement is relative to the coordinate axis, not the current position of the turtle. The turtle's heading is unchanged.