Turtle graphics in Python:

Basic Setup:

To use Turtle, you need to import it first:

import turtle

Then, you create a turtle object which you can use to control the drawing.

t = turtle.Turtle()

This `t` object represents the turtle that will move around and draw on the screen.

Basic Commands:

1. Movement:
    - `forward(x)` or `fd(x)`: Moves the turtle forward by `x` units.
    - `backward(x)` or `bk(x)`: Moves the turtle backward by `x` units.
    - `left(angle)`: Turns the turtle left by the specified angle (in degrees).
    - `right(angle)`: Turns the turtle right by the specified angle.
2. Drawing:
    - `penup()`: Lifts the pen, so the turtle moves without drawing.
    - `pendown()`: Lowers the pen, so the turtle starts drawing again.
3. Colors:
    - `color(color)`: Changes the turtle's pen color.
    - `bgcolor(color)`: Changes the background color of the window.
4. Speed:

- `speed(speed)`: Sets the turtle's speed. It can be a number from `0` (fastest) to `10` (slowest), or `fastest`, `fast`, `normal`, and `slowest`.

5. Other useful methods:

   - `circle(radius)`: Draws a circle with the given radius.
   - `goto(x, y)`: Moves the turtle to the specific coordinates `(x, y)`.
   - `setpos(x, y)`: Moves the turtle to the coordinates `(x, y)` without drawing.
   - `clear()`: Clears the drawing area.

Example:

Here's a simple example of how to draw a square using Turtle:

```
import turtle


# Set up the turtle

t = turtle.Turtle()


# Loop to draw a square

for _ in range(4):

    t.forward(100)  # Move forward by 100 units

    t.left(90)  # Turn left by 90 degrees


# Keep the window open

turtle.done()
```

This code will draw a square with side length 100 units.

Example 2: Drawing a Circle

```
import turtle

# Set up the turtle
t = turtle.Turtle()

# Draw a circle with radius 100
t.circle(100)

# Keep the window open
turtle.done()
```

Turtle Screen:

You can also customize the screen/window using the `Screen()` class.

```
import turtle

# Create a screen object
screen = turtle.Screen()
```

```
# Set the background color of the window
screen.bgcolor("lightblue")

# Set the window title
screen.title("Turtle Graphics Example")

# Create a turtle object
t = turtle.Turtle()

# Draw something
t.forward(100)

# Keep the window open
turtle.done()
```

Event Handling:

Turtle also supports event-driven programming, where you can bind actions to mouse clicks, key presses, etc.

For example, using a click event:

```
import turtle
```

```
# Function to move the turtle to the mouse click position
def move_turtle(x, y):
    t.penup()
    t.goto(x, y)
    t.pendown()

# Set up the turtle
t = turtle.Turtle()

# Bind the click event
turtle.onscreenclick(move_turtle)

# Keep the window open
turtle.done()
```

Additional Tips:

Turtle Reset: If you need to clear the screen and start fresh, you can use `reset()`:

```
t.reset()
```

-

Exit Turtle Graphics: You can use `exitonclick()` to close the Turtle graphics window when clicked:

```
turtle.exitonclick()
```