

Tech Dawgs Turtle Workshop for Kids

2/3/25

The Goal of this workshop is to teach kids about some basics of computer programming. We want to show that programming is accessible, easy to learn, and most importantly: fun!

1. Introduction to Python:

a. What is Python?

- i. Python is a **computer programming language**. It is essentially how you tell a computer **what to do**; computers **don't know what to do** unless we tell them!
- ii. The very **same python** programming language we will be using today is used by **professional programmers** all over the world.

1. Examples:

a. Data Science

- i. Commonly used in scientific research
- ii. Used in data recommendations; think Amazon or Netflix recommending movies

b. Automation

- i. Can be used to automate repetitive tasks
- ii. "Imagine if you could create a program to automate all your chores!"

c. Web Development

- i. Used in apps like Spotify, Instagram, and Dropbox

b. What is Turtle?

- i. Turtle is a **Python Library** – almost like a **mod** or **addon** to Python
- ii. It allows users to create drawings and graphics by giving a small "turtle" instructions

2. Starting out with Turtle

a. Website: <https://pythonsandbox.com/turtle>

b. Variables

- i. Variables are placeholders – Think like saving your seat in the cafeteria or on the bus.
- ii. You can assign variables to be anything you like
- iii. In this case the letter 't' has been assigned to the turtle module. This is so the computer knows whenever we put 't.' in front of a command, we want the *turtle* to perform the command.
 1. Make sure you put 't.' in front of anything you want the turtle to do.

3. Basic Turtle Commands

a. Commands:

- i. Turtle.shape - The turtle.shape() command changes the appearance of the turtle, allowing you to choose from predefined shapes like "turtle", "circle", "square", or even custom images.
 - ii. turtle.forward(distance) – Moves the turtle forward by a specified distance (e.g., 100 units).
 - iii. turtle.backward(distance) – Moves the turtle backward by a specified distance.
 - iv. turtle.right(angle) – Turns the turtle to the right by a certain angle (in degrees).
 - v. turtle.left(angle) – Turns the turtle to the left by a certain angle (in degrees).
 - vi. turtle.circle(radius) – Draws a circle with the specified radius. The turtle will follow the path of the circle.
 - vii. turtle.dot() – Draws a dot (small circle) at the turtle's current location.
 - viii. turtle.speed(speed) – Sets the turtle's speed (1 is slowest, 10 is fastest, "fastest" is the maximum speed).
 - ix. turtle.color(color) – Sets the pen color to the specified color (can be a string like "red", "blue", or RGB values).
 - b. Think about a circle. It has 360 degrees. The turtle starts out going to the right side of the screen. If we want it to go straight up it would need to turn left 90°.
 - c. If we wanted to make a square we would need to make 4* 90° turns.
 - d. How many turns would it take for a triangle?
 - e. A equilateral triangle has 3* 120° angles
 - f. Ask the kids to try and make a square or triangle
4. Loops
- a. A big part of computer programming is figuring out all the shortcuts.
 - b. Loops are an easy way to tell a computer to do something over and over again.
 - i. For loops
 - 1. for [var] in range(repetitions)
 - c. Using loops we can make a square or triangle
 - d. Challenge the kids to make a hexagon or octagon
 - i. Dodecagon: 10* 36°
 - ii. Octagon: 8* 45°
 - iii. Hexagon: 6* 60
5. Explain Coordinates
- a. A coordinate grid is a system that uses two numbers (coordinates) to specify a point's location on a flat surface, like a map. The grid consists of two axes: the x-axis (horizontal) and the y-axis (vertical), which intersect at the origin (0, 0). The x and y values describe how far along each axis the point is located, like (3, 2) meaning 3 units right and 2 units up.
 - b. Think of the game battleship, instead of using letters and numbers, its two numbers
6. Intermediate Turtle Commands

a. Commands:

- i. `turtle.penup()` – Lifts the pen off the screen, so it doesn't draw while moving.
- ii. `turtle.pendown()` – Lowers the pen to the screen, allowing the turtle to draw while moving.
- iii. `turtle.setposition(x, y)` – Moves the turtle to a specific position (x, y) on the screen.
- iv. `turtle.goto(x, y)` – Moves the turtle to the given coordinates (x, y).
- v. `turtle.setheading(angle)` – Sets the turtle's heading (direction) to a specific angle.
- vi. `turtle.begin_fill()` – Begins filling the shape that the turtle is about to draw.
- vii. `turtle.end_fill()` – Ends the filling process and fills the shape with the current pen color.
- viii. `turtle.fillcolor(color)` – Sets the fill color for shapes drawn by the turtle.
- ix. `turtle.pensize(size)` – Sets the thickness of the pen line (can be any positive integer).
- x. `turtle.write(text)` – Makes the turtle write text at its current location (you can also change font size, style, etc.).
- xi. `turtle.bgcolor(color)` – Sets the background color of the screen.

7. Advanced Commands

a. Random

- i. `from random import randint`
- ii. `t.setposition(randint(min],[max]),randint(min],[max])`
- iii. `t.color(randint(0,255),randint(0,255),randint(0,255))`
- iv. `t.setheading(randint(-359,359))`