

Python for Kids: Comprehensive Coding Curriculum

A structured, interactive 20-day path from computer science foundations to graphic programming

WEEK 1: FUNDAMENTALS & ENVIRONMENT

Day 1: Introduction to Coding & Setup

Core mechanism of Input → Process → Output. Step-by-step installation of Python 3.7 on Windows. Exploring IDLE and writing your first program: `print("hi python")`.

Day 2: File Management & Workflow Shortcuts

Creating, saving, and executing scripts using F5. Core keyboard shortcuts (Ctrl+S, Ctrl+N, Ctrl+C/V/Z). Windows Command Prompt basics (CD, DIR, MD, RD).

Day 3: Print Function, Strings & Escape Characters

Mastering the `print()` function and string text data. Solving string syntax errors using double quotes and escape characters (`'`, `\"`). Conceptual overview of variables.

Day 4: Storing Variables & Formatting Output

Differentiating data types (strings vs. integers) via `type()`. Variable naming rules and conventions (snake_case). Sophisticated output with f-strings and multi-line triple quotes.

Day 5: Interactive Cloud Coding via Google Colab

Introduction to browser-based programming in Google Colab notebooks. Learning the interface using code blocks and text blocks. Interactive exercises: "Introduce Yourself" and "Quote a Quote".

WEEK 2: MATH & DATA COLLECTIONS

Day 6: Arithmetic Operators & Boolean Logic

Using numeric data types (integers and floats). Python arithmetic operators (`+`, `-`, `*`, `/`, `%`, `//`, `**`) and PEMDAS rules. Introducing Comparison Operators resulting in True/False.

Day 7: Chapter 3 Lab Activities

Hands-on applications in Google Colab. "How Old Are You?" variable exercises and the complex "Operation PEMDAS" challenge to compute a target variable value through creative math operations.

Day 8: String Manipulation & Intro to Lists

String concatenation and multiplication repetition. Introducing List collections for multi-object grouping. Mastering ordered, zero-based list indexing and data type flexibilities.

Day 9: Advanced List Operations & Mutability

Extracting sublists with list slicing [`start:end`]. Mutating data using addition assignment (`+=`), membership operators (`in`, `not in`), and built-in methods (`.append()`, `.remove()`, `del`).

WEEK 3: CONTROL FLOW & AUTOMATION

Days 10-11: Conditionals & Chapter 4 Labs

Introduction to Tuples (immutable datasets) and `if/elif/else` statements for code paths. Practicing the "Random Factory" lab activity focusing on index retrieval and concatenation scenarios.

Day 12: Loops, Iteration & Defending Against Bugs

Repeating execution blocks using `for` loops (with single/multi-parameter `range()`) and `while` loops under condition counters controlled dynamically by `len()`. Halting infinite loops using `Ctrl+C`.

Day 13: Chapter 5 Lab & Python's History

Origin of Python's name (Monty Python). Automating greetings using loops, conditional list sorting ("Categorizing Cat Friends" matrix), and character string extraction ("Looping Letters").

WEEK 4: TURTLE GRAPHICS & MODULES

Day 14: Turtle Module & Object-Oriented Logic

Built-in graphic rendering. Basic Object-Oriented Programming (OOP) paradigms: instances, properties, and methods. Initializing screen and canvas targets. Core RGB color channel constraints (0-255).

Day 15: Hexadecimal Systems & Graphic Customization

Deep-dive into base-16 number systems (0-9, a-f) for web colors. Setting environment properties with `colormode(255)` and resizing object boundaries with `turtlesize()` parameters.

Day 16: Geometry Rendering & Script Optimization

Vector control vectors (`forward()`, `back()`, `left()`, `right()`). Structuring geometric graphics to draw a perfect square. Optimizing manual lines with clean 4-count `for` loops.

Day 17: Multi-Parameter Shapes & Typography

Filling vector loops with flat colors. Modifying the `circle()` function with radius, extent, and angular step parameters. Generating canvas text styling using `write()` configurations.

Day 18: Chapter 6 Capstone Graphic Animation

Building vector math games and animations. Scripting a five-point automated star generator and creating an interactive coordinate-based "Fortune Teller Spinner" with the `random` library.

WEEK 5: ADVANCED FUNCTIONS & REVIEW

Day 19: Time Calculations & Chapter 7 Activities

Defining parameters inside custom code scopes. Building real-world applications using `datetime` and `timedelta` to process regional time zone differences simultaneously across global cities.

Day 20: Program Culmination & Synthesis

Comprehensive structural review of chapters 1 through 6. Consolidating logic paths, collection behaviors, graphic architectures, and troubleshooting workflows to ensure self-sufficiency.