Junior Python Module Plan (15 Sessions)

Inspired by our JavaScript module structure, this roadmap is split into **three levels**, each containing **five sessions** that build foundational programming skills for total beginners (teens). Every session follows the same pattern:

1. **Review & Introduction** – recap previous lesson, set objectives
2. **Demo** – instructor-led code walkthroughs
3. **Apply** – guided hands-on exercises in the online IDE
4. **Homework** – small take-home practice task

Learners will code exclusively in a web-based IDE (e.g., Replit or CodeSandbox) so no local setup is required.

Prerequisites

- Comfortable using a web browser and keyboard
- Completed HTML & CSS modules (optional but recommended for context)

Level 1 – Programming Fundamentals (Sessions 1-5)

Session 1: Getting Started with Python

Demo

- Navigating the online IDE
- `print()` statements & comments
- Basic data types: integers, floats, strings
  Apply – Create a *Hello, Python!* program displaying name & age.
  Homework – Add decorative ASCII art using multi-line strings.

Session 2: Variables, Input & Arithmetic

Demo – `input()`, variables, arithmetic operators, order of operations.
Apply – Simple calculator.
Homework – Temperature converter (°C ↔ °F).

Session 3: Conditionals

Demo – `if`, `elif`, `else`, comparison & logical operators.
Apply – Age-based movie rating checker.
Homework – Even/odd number detector.

Session 4: Loops

Demo – `while`, `for`, `range()`, `break`, `continue`.
Apply – Number guessing game.
Homework – Multiplication table generator.

Session 5: Mini Project & Review

Combine topics 1-4 to build a *Simple Quiz* (questions stored in lists).
Homework – Refactor quiz for at least 5 questions.

Level 2 – Data Structures & Functions (Sessions 6-10)

Session 6: Lists & Basic List Methods

Create, index, slice, mutate, `append`, `pop`, `len`.
Homework – Grocery list manager.

Session 7: Dictionaries & Tuples

Key-value pairs, nested dicts, tuples for constants.
Homework – Student gradebook using a dictionary.

Session 8: Functions

Defining functions, parameters, return values, scope.
Homework – Function that checks prime numbers.

Session 9: Combining Data Structures & Functions

Build a text-based *To-Do* app using functions + lists/dicts.
Homework – Add “mark as done” feature.

Session 10: Mid-Course Project

Capstone for Levels 1-2: *Text Adventure* game utilising loops, conditionals & functions.
Homework – Add at least two extra rooms.

Level 3 – Fun Projects & Real-World Uses (Sessions 11-15)

Session 11: Cool Built-in Tricks

What you’ll learn – How to *import* built-in helpers like `random`, `time`, and `math`.
Mini-make – Build a digital dice roller that shows random numbers.
Homework – Make a *coin flip* program that counts heads/tails.

Session 12: Drawing with Turtle

What you’ll learn – Use the friendly `turtle` module to move a pen on screen.
Mini-make – Draw a colourful house using simple shapes.
Homework – Draw your initials in turtle lines.

Session 13: Turtle Race Game

What you'll learn – Loops + turtle to animate movement.
Mini-make – Code a turtle race where players guess the winner.
Homework – Add a countdown timer before the race starts.

Session 14: Saving & Loading Text

What you'll learn – Write and read plain text files for scores or notes.
Mini-make – Save the fastest turtle race time to a file.
Homework – Make a *journal.txt* where kids can write a daily note.

Session 15: Final Project & Celebration

Goal – Combine your favourite ideas into one small game or art piece.
Examples: mini quiz that saves high scores, a multi-colour turtle drawing, or a story generator.
We'll share projects, give friendly feedback, and wrap up the course.

Assessment & Progression

- **Homework** after every session (short, auto-test friendly).
- End-level mini projects (Sessions 5, 10, 15) showcase mastery and unlock the next level.

Next Steps

1. Review this outline and suggest changes.
2. Flesh out each session with timed agenda, demo code snippets, and starter templates.
3. Prepare homework auto-grading or rubrics in the IDE.