
🐍 Python for Beginners – Week 1 Curriculum

📅 **Duration:** 5 Days
🎯 **Goal:** Build a solid foundation in core Python concepts with hands-on practice and daily assignments.

🔹 Day 1: Welcome to Python

🔧 Concepts Introduced:

- What is Python? Why is it popular?
- Installing Python and VS Code
- Writing your first Python script (`print()`)
- Code structure, indentation, and comments

📝 Assignment:

"Getting to Know You"
Create a script that prints:

- Your name
- Your age
- Your favorite hobby
- A one-line motivational quote

💡 Mini Challenge:

Print the sentence:
`Python is fun!`
…five times using only one line of code.

🔹 Day 2: Variables & Data Types

🔧 Concepts Introduced:

- Variables and assignment
- Data types: `int`, `float`, `str`, `bool`
- Checking types with `type()`
- Type conversion (`str()`, `int()`, `float()`)

📝 Assignment:

"My Digital ID Card"
Ask the user for:

- Full name
- Age
- Favorite color
- Student ID (numbers only)

Then print it in a formatted multi-line string like a badge.

💡 Mini Challenge:

Convert an age (e.g., 23) into months, days, and hours (approx.).

🔹 Day 3: Operators & Expressions

🔧 Concepts Introduced:

- Arithmetic operators: `+`, `-`, `*`, `/`, `//`, `%`, `**`
- Comparison operators: `==`, `!=`, `>`, `<`
- Logical operators: `and`, `or`, `not`

📝 Assignment:

"Simple Math Bot"
Build a calculator that:

- Asks the user for two numbers
- Performs all 5 basic operations
- Displays the results clearly

💡 Mini Challenge:

Let the user input a number. Print:

- If it's odd or even
- If it's divisible by 3

🔹 Day 4: Conditional Statements (Control Flow)

🔧 Concepts Introduced:

- `if`, `elif`, and `else` statements
- Nesting conditions
- Boolean expressions

📝 Assignment:

"Grading Assistant"
Ask the user for a numeric test score. Based on it:

- Print "Excellent" for 90+
- "Good" for 70–89
- "Pass" for 50–69
- "Fail" below 50

Also, let them know if the score is an exact multiple of 5.

💡 Mini Challenge:

Write a program that checks if a given year is a leap year.

🔹 Day 5: Loops – Repetition Made Easy

🔧 Concepts Introduced:

- `for` loops with `range()`
- `while` loops
- Loop control: `break`, `continue`, `pass`

📝 Assignment:

"Number Fun Zone"

- Print all odd numbers between 10 and 50
- Ask the user to guess a number between 1 and 10 using a `while` loop until correct

💡 Mini Challenge:

Create a loop that prints every third number from 3 to 30.

🐍 Python for Beginners – Week 2 Curriculum

📅 **Duration:** 5 Days
🎯 **Goal:** Strengthen programming logic, master key Python structures (functions, lists, dictionaries), and apply them in real scenarios.

🔹 Day 6: Functions – Writing Reusable Code

🔧 Concepts Introduced:

- Defining functions with `def`
- Parameters and return values
- Function scope (local vs global variables)
- Calling functions from within other functions

📝 Assignment:

"Smart Calculator (v2)"

- Create a calculator where each operation is a function
- Functions: `add()`, `subtract()`, `multiply()`, `divide()`
- The main function lets users choose an operation

💡 Mini Challenge:

Create a `greet_user(name)` function that formats a custom message depending on the time of day (simulate with a variable).

🔹 Day 7: Lists & Tuples – Organizing Collections

🔧 Concepts Introduced:

- Creating lists and tuples
- Indexing, slicing, and modifying lists
- Common list methods: `append()`, `remove()`, `sort()`, `pop()`
- Looping through lists with `for`
- Tuples: immutability and use cases

📝 Assignment:

"Classroom List Manager"

- Add students to a list
- Allow removal of a student
- Display the list sorted alphabetically

Add validation: only allow unique names.

💡 Mini Challenge:

Create a list of numbers. Print the sum of all even numbers in the list.

🔹 Day 8: Dictionaries & Sets – Key-Value Magic

🔧 Concepts Introduced:

- Creating dictionaries
- Accessing, adding, and updating values
- Iterating over keys and values

- Using `.get()` safely
- Intro to `set` and uniqueness

📝 Assignment:

"Student Grades Tracker"

- Store names as keys and grades as values
- Allow updating a grade
- Print a report showing names and status: Pass/Fail

Use a loop and `if` statement to determine pass/fail based on grade.

💡 Mini Challenge:

Write a program that counts word frequency in a sentence using a dictionary.

🔹 Day 9: Practice & Quiz 2 – Logic Builder Day

🧠 Activities:

- Problem-solving using functions, lists, and dictionaries
- Review nested data (list of dictionaries, dict of lists)

📋 Quiz 2 Topics:

- Functions
- Lists and Tuples
- Dictionaries
- Logic in loops and conditions

📝 **Assignment:**
"Mini Library System"

- Store book titles and availability (`True/False`) in a dictionary
- Add features to:
    - View all books
    - Mark as borrowed or returned

Use functions for each action.

🔹 Day 10: String Manipulation – Playing with Text

🔧 Concepts Introduced:

- String methods: `.lower()`, `.upper()`, `.strip()`, `.replace()`, `.split()`
- Joining and splitting strings
- `in` keyword and membership tests
- f-strings for formatted output

📝 Assignment:

"Sentence Styler"

- Ask the user to input a sentence
- Convert it to title case
- Replace any bad words from a predefined list with `***`

💡 Mini Challenge:

Ask for a name and format it:
`"Hello, JANE DOE"` (all uppercase)

🐍 Python for Beginners – Week 3 Curriculum

📅 **Duration:** 2 Days + Final Project
🎯 **Goal:** Empower students to build full Python scripts using external files, handle errors gracefully, and complete a working project.

🔹 Day 11: File Handling – Saving & Reading Data

🔧 Concepts Introduced:

- Opening files with `open()`: modes `"r"`, `"w"`, `"a"`
- Reading from files: `.read()`, `.readlines()`
- Writing to files with `.write()`
- Using `with` blocks to manage files safely

📝 Assignment:

"Notes App (v1)"

- Prompt user to write a short note
- Save it to a file (append mode)
- Display all saved notes on command

Organize using functions: `write_note()`, `read_notes()`

💡 Mini Challenge:

Write a script that copies all lines from one file to another.

🔹 Day 12: Error Handling & Modules

🔧 Concepts Introduced:

- Try-except blocks for catching runtime errors
- Catching specific errors (`ValueError`, `ZeroDivisionError`, `FileNotFoundError`)
- `finally` block
- Using standard Python modules: `random`, `math`, `datetime`

📝 Assignment:

"Resilient Calculator"

- Add error handling to the calculator
- Catch division by zero, invalid inputs, and more
- Use `math` for extra functions: square root, power

💡 Mini Challenge:

Use `random.choice()` to simulate a dice roll or coin toss.

🔹 Day 13: Final Project – Your First Python App

🎯 Project Goal:

Apply all major concepts (functions, conditionals, loops, lists/dictionaries, file I/O, error handling) to build a complete working script.

🛠️ Project Options (Choose One):

---

1. To-Do List Manager

- Add, delete, and mark tasks as done
- Store tasks in a text file
- Show tasks on startup

---

2. Expense Tracker

- Add and categorize expenses
- Save to file with date and amount
- Show total and category summary

---

3. Student Result Portal

- Input student scores for subjects
- Calculate and store average
- Show pass/fail and grade

✅ **Project Requirements:**

- Use at least 2 functions
- Include user input and validation
- Handle file saving/loading
- Handle at least 1 type of exception
- Have clean output and formatting

🌟 Wrap-Up: Course Review + Next Steps

- Quick review of:
    - Variables & types
    - Loops, functions, collections
    - File handling & error management
- Encourage students to:
    - Expand their final project
    - Explore basic Python modules (`os`, `csv`, `json`)
    - Start learning **Django** (in a future separate course)