Student Guide for Sync Session

<<Week 13: Python Basics and Quick Recap>>

This guide is your roadmap to making the most of our online session. Packed with essential tips and strategies, it's designed to keep you engaged, prepared, and ready to dive into a smooth and productive learning journey. Get ready to participate, learn, and thrive!

Session Overview

<table>
<tr><td><b>Session title</b></td><td>Python Basics and Quick Recap</td></tr>
<tr><td><b>Session duration</b></td><td>3 hours</td></tr>
<tr><td><b>Session type</b></td><td><ul><li><b>Lectures:</b> Introduction to Python, its syntax and features, and core data types.Case Studies:</li><li><b>Case studies</b>: Practical implementation of variables, data types, expressions, string operations, and markdown cells using Jupyter Notebook.</li></ul></td></tr>
<tr><td><b>Scope</b></td><td>This session is a quick but thorough recap of Python fundamentals.<ul><li>It introduces Python's simplicity and versatility, syntax rules, key data types, and operations.</li><li>Learners will also gain familiarity with Jupyter Notebooks through hands-on activities.</li></ul></td></tr>
<tr><td><b>Learning Objectives</b></td><td><table>
<tr><th>Learning objectives</th><th>Core capability</th></tr>
<tr><td>Understand the basics and philosophy of Python</td><td>Conceptual clarity on programming principles</td></tr>
<tr><td>Apply syntax rules and indentation correctly</td><td>Code readability and correctness</td></tr>
<tr><td>Implement core Python data types</td><td>Practical coding ability</td></tr>
<tr><td>Perform arithmetic and comparison expressions</td><td>Logical operations in Python</td></tr>
<tr><td>Execute string manipulations</td><td>Text processing in real-world tasks</td></tr>
<tr><td>Use Jupyter Notebook for coding and markdown</td><td>Tool proficiency for coding workflows</td></tr>
</table></td></tr>
<tr><td><b>Software/Tools</b></td><td><ul><li>Python (via Anaconda distribution)</li><li>IDE: Jupyter Notebook</li><li>Presentation Tool: PowerPoint</li></ul></td></tr>
</table>

Pro tips for success

- **Ask Bold Questions**: No question is too small—curiosity is the key to learning!
- **Be Hands-On:** Coding is your superpower. Tweak, test, and break things (safely) to learn.

- **Collaborate**: Share your ideas in discussions. You might just spark the next big insight!

Session Details

Topic	A glimpse	Insight / Actionable
Introduction to Python	Understand Python's design philosophy of simplicity and readability.	Think about why Python's "focus on problem-solving" makes it popular among developers.
Features of Python	Explore key features like dynamic typing, cross-platform support, and its large library base.	Reflect on which feature might benefit your own coding journey.
Python Syntax	Learn how Python uses indentation and avoids unnecessary syntax clutter.	Practice writing clean, readable Python code without semicolons or braces.
Launching Jupyter Notebook	Discover how to start Jupyter using Anaconda and run your first notebook.	Try creating your first notebook and running a simple print("Hello, Python!") line.
Hands-on: Python Data Types	Get familiar with variables, lists, tuples, dictionaries, and sets.	Create one of each data type and observe how they behave differently.
Activity: Check Data Type	Use type() to identify Python data types.	Try experimenting with new types and check their outputs.
Activity: Create List	Create a list of numbers from 1 to 10.	Try modifying the list, e.g., adding, removing, or slicing elements.
Activity: Create Tuple	Understand the immutability of tuples.	Reflect on when you'd use a tuple vs. a list in real-world scenarios.
Activity: Create Dictionary	Use key-value pairs to organise data.	Build a dictionary representing your profile (e.g., name, country, skill level).
Activity: Create Set	Explore the uniqueness property of sets.	Create a set with duplicates and see how they're removed.
Activity: Markdown Cell	Use Markdown to add headings and structure in Jupyter.	Add formatted notes alongside your code in a notebook.
Hands-on: Expressions	Practice arithmetic and logical expressions in Python.	Write small programs to perform calculations like interest or BMI.
Activity: Simple Interest	Code a formula using variables to compute simple interest.	Reuse the formula for other business or finance applications.
Activity: Comparison	Use relational operators like >, <, ==, !=.	Test how different values evaluate in logical conditions.
Hands-on: String Operations	Use string methods such as upper(), replace(), count() and split().	Manipulate your own strings for practice, like names or favorite quotes.
Activity: Convert Case	Try converting strings to uppercase, lowercase, and title case.	Think of formatting use-cases like customer names or titles in a resume.
Activity: Find and Replace	Locate and modify substrings.	Replace words in a sentence to understand text processing.
Activity: Count Characters	Learn to count character frequency.	Try counting characters in your name or a sentence.

Activity: Split String	Break a string into a list using commas or spaces.	Parse a CSV-style string to understand basic text-to-data conversion.
Async Recap	Refresh your knowledge on what you've learned asynchronously.	Participate in the review quiz or contribute to a group reflection.
Q&A	Ask questions to clear doubts about Python basics.	Prepare 1–2 questions from your practice to ask during the session.
Closing and Thank You	Review key takeaways and next steps.	Reflect on how you'll use Python in your project or job.

Post-Session Activities

Reflection Challenge	What surprised you most about Python's simplicity or functionality? Write a short note or share your thoughts in the forum.
Explore More	• **Read**: Python official documentation to explore data types and functions.. • **Watch**: YouTube tutorials on Python projects using Jupyter Notebook.. • **Try**: Apply your Python skills to automate a daily task—like sorting files or analysing survey results.
Get Inspired	Python powers Instagram, Netflix, and even NASA's projects. You're learning a language that helps drive innovation across industries.
The Journey Ahead	Next, we'll explore Python libraries for data analysis. Stay curious—your coding skills are about to go to the next level!