📚 Python Arithmetic Basics

Brief Overview

This note covers **Python arithmetic basics** and was created from the Lecture 1: Integers, Floats & Math Operations in Python | #python #datasciencecourse #2025 YouTube video. It provides a concise walkthrough of elementary math operations, data type distinctions, and practical tips for working in VS Code and Jupyter Notebooks.

Key Points

- Basic arithmetic: addition, subtraction, multiplication, division, floor division, and exponentiation.
- Numeric data types: integers vs. floats, type inspection, and floating-point behavior.
- VS Code and Jupyter workflow: file handling, cell types, and execution shortcuts.
- Variable basics and printing techniques, including formatting options.

🧮 Basic Mathematical Operations

Addition ➕

> **Addition**: The process of combining two numbers to get their sum.

```
3 + 4   # returns 7
```

Subtraction ➖

> **Subtraction**: The process of finding the difference between two numbers.

```
4 - 5   # returns -1
```

Multiplication & Exponentiation ✖️

> **Multiplication**: Repeated addition, using the * operator.
> **Exponentiation**: Raising a base to a power, using the ** operator.

```
3 \times 4      # 12
3 ** 4     # 81 (3 to the power of 4)
```

Division vs. Floor Division ➗

Operator	Description	Example	Result
/	True division – always returns a **float**	42 / 2	21.0
//	Floor division – returns the **integer part** of the quotient	42 // 2	21
	Handles negative numbers by rounding **down** to the next lower integer	-45 // 2	-23
	Works with floats, discarding the fractional part	56.7 // 2	28.0

Key points

- / yields a float even when the division is exact.
- // discards the fractional part; for negative results it rounds **down** (e.g., -22.5 // 1 → -23).

📂 Working in VS Code & Jupyter Notebooks

- **Creating a new file**:
  - macOS: click *File → New File* or use the top menu.
  - Save with **Cmd + S** (macOS) or **Ctrl + S** (Windows).
- **Naming**: Any name is acceptable; example used – *Python lecture one*.

- **Running code in a notebook**: Press **Shift + Enter** to execute the current cell and move to the next.
- **Choosing a kernel**: Select the Python version you have installed (e.g., 3.10.12 or 3.14.0).
- **Cell types**:
  - **Code cell** – executable Python code.
  - **Markdown cell** – non-executable text for comments, headings, etc. Create with the **down arrow** button or by pressing **Shift + Enter** on a selected cell and choosing *Markdown*.
- **In-cell comments**:
  - macOS shortcut **Shift + 3** inserts the # symbol to start a comment line.

📊 Data Types: Integers vs. Floats

> **Integer (int)**: Whole numbers without a decimal point.
> **Float (float)**: Numbers that include a decimal point, even if the fractional part is zero.

```
type(90)    #
type(90.0)  #
```

- The presence of a **period** (.) determines whether a numeric literal is treated as a float.
- Operations between two floats (addition, subtraction, multiplication, division) always produce a float result.

🧩 Variable Notion & Printing (brief)

> **Variable**: A named reference that stores a value, allowing reuse throughout the program.

- **Printing**: Use print() to display values.
- **Special formatting**: Python's print() can be customized with separators, end characters, and string interpolation (e.g., f-strings).

(Further string operations, slicing, and advanced formatting were mentioned but not detailed in the transcript.)