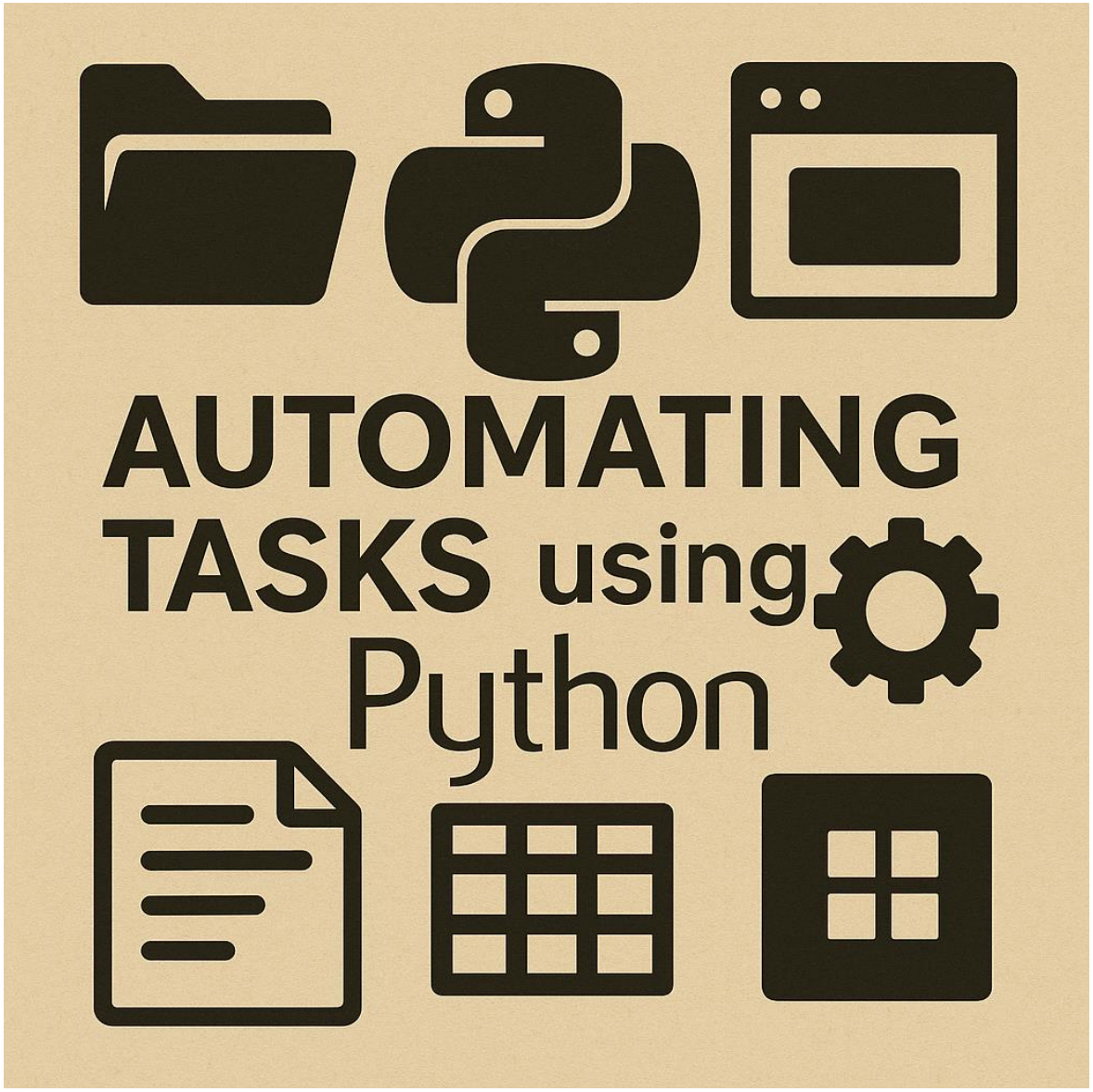

AUTOMATING
TASKS using
Python

- Valdis Saulespurens
- Lecturer, Riga Technical University
- Faculty of Computer Science, Information Technology and Energy

RIGA TECHNICAL UNIVERSITY

Course Materials: **https://github.com/ValRCS/RTU_Automating_Tasks_With_Python**

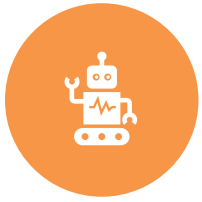
Course Overview

1. Getting Started with Python + VS Code

2. Working with Text Files & String Processing

3. Automating Excel with openpyxl

4. Data Processing with pandas



5. PDF Automation, Image Processing

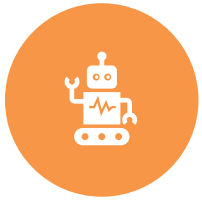
Course Overview

6. Web Scraping (requests, BeautifulSoup)

7. Browser Automation with Selenium

8. Desktop GUI Automation with pyautogui

9. Email & File System Automation



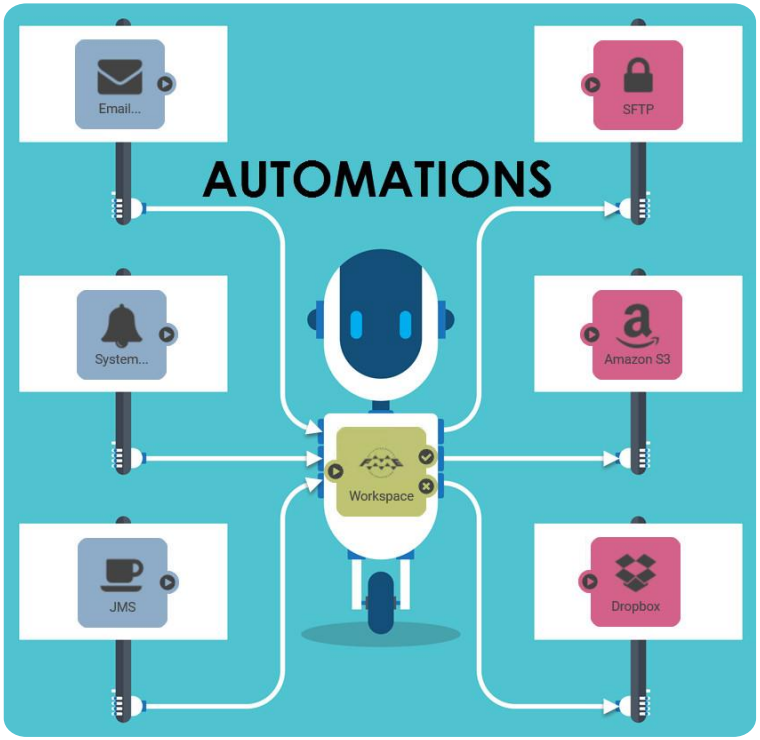
10. Final End-to-End Automation Project

python™

What We'll Do Today

- Why Python and short History
- Install Python
- Verify installation
- Create .py files in Notepad
- Install & configure VS Code
- Run scripts
- Basic Python
- Mini automation teaser

What Do We Mean by Automation?



- Using software to perform tasks automatically instead of manually
- Reducing repetitive work and minimizing human error
- Saving time by letting the computer handle routine tasks
- Standardizing processes for consistent and reliable results
- Connecting systems together (files, apps, websites, APIs)
- Triggering tasks automatically based on schedules or events
- Scaling up tasks beyond what is feasible by hand

Why Python for Automation?

- Simple & readable syntax → faster development and fewer bugs
- Extremely popular with a huge community and long-term stability
- Great 'glue language' for connecting files, APIs, tools, and services
- Rich standard library ('batteries included') for common automation tasks
- Powerful ecosystem: Excel, PDF, web scraping, GUI automation, and more
- Cross-platform: works on Windows, macOS, and Linux
- Excellent tooling: VS Code, Jupyter, virtual environments, pip
- Fast prototyping: write useful scripts in minutes
- Ideal for both beginners and experts

A Brief History of Python & Guido van Rossum

- Created by Guido van Rossum in late 1980s at CWI (Netherlands)
- First public release in 1991 (Python 0.9.0)
- Designed for readability, simplicity, and a rich standard library
- Python 2 released in 2000; Python 3 released in 2008
- Python 2 officially retired in 2020 after long transition
- Used today for automation, data science, AI/ML, web development, and more - #1 programming Language
- Guido served as Python's BDFL until 2018
- Now guided by the Python Steering Council (Guido still influential)
- Guido worked at Google, then Dropbox then Microsoft

Google

Dropbox

Microsoft

Installing Python

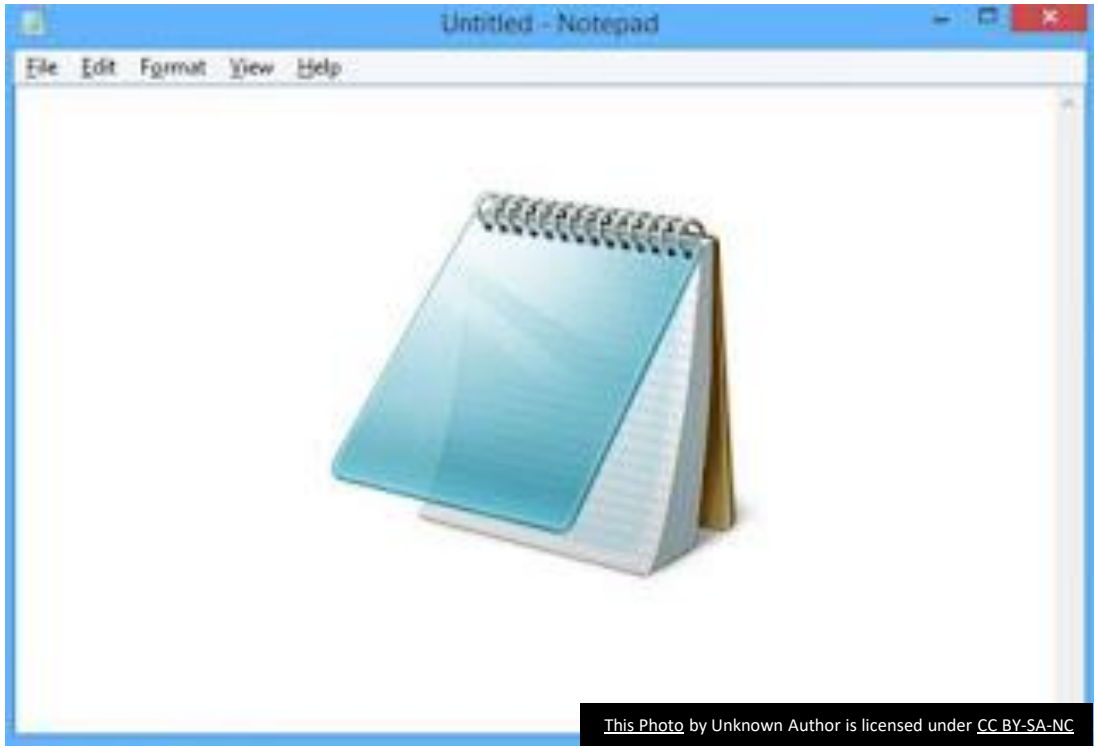
- Download from https://**python.org**
- Choose Windows Installer (64-bit) (unless on MacOs or ARM tablet)
- **Get Last Years Latest Version (**3.13.9 in 2025)
- Check 'Add Python to PATH'
- Use recommended settings

Verify Python Installation

- Run in CMD:
- python --version
- python

Python is Just Text

- Use any editor (even Notepad)
- Example:
- print("Welcome to Python automation")
- Save as hello.py
- Run: python hello.py



Installing Visual Studio Code

- Download from code.visualstudio.com
- Standard install
- Cross-platform

Essential VS Code Extensions

- Python – REQUIRED https://marketplace.visualstudio.com/items?itemName=ms-python.python
- Pylance
- Jupyter
- Black
- isort
- Optional: PowerShell, WSL

Setting Up Your Workspace

Create

- Create folder: python_automation_day1

Open

- Open folder in VS Code

Select

- Select Python interpreter

VS Code Shortcuts

- Ctrl+Shift+P – Command Palette
- Ctrl+` – Terminal
- Ctrl+F5 – Run
- Ctrl+P – File search
- Ctrl+Shift+F – Search Text in project
- Many more – can customize!

https://code.visualstudio.com/docs/reference/default-keybindings

First Script in VS Code

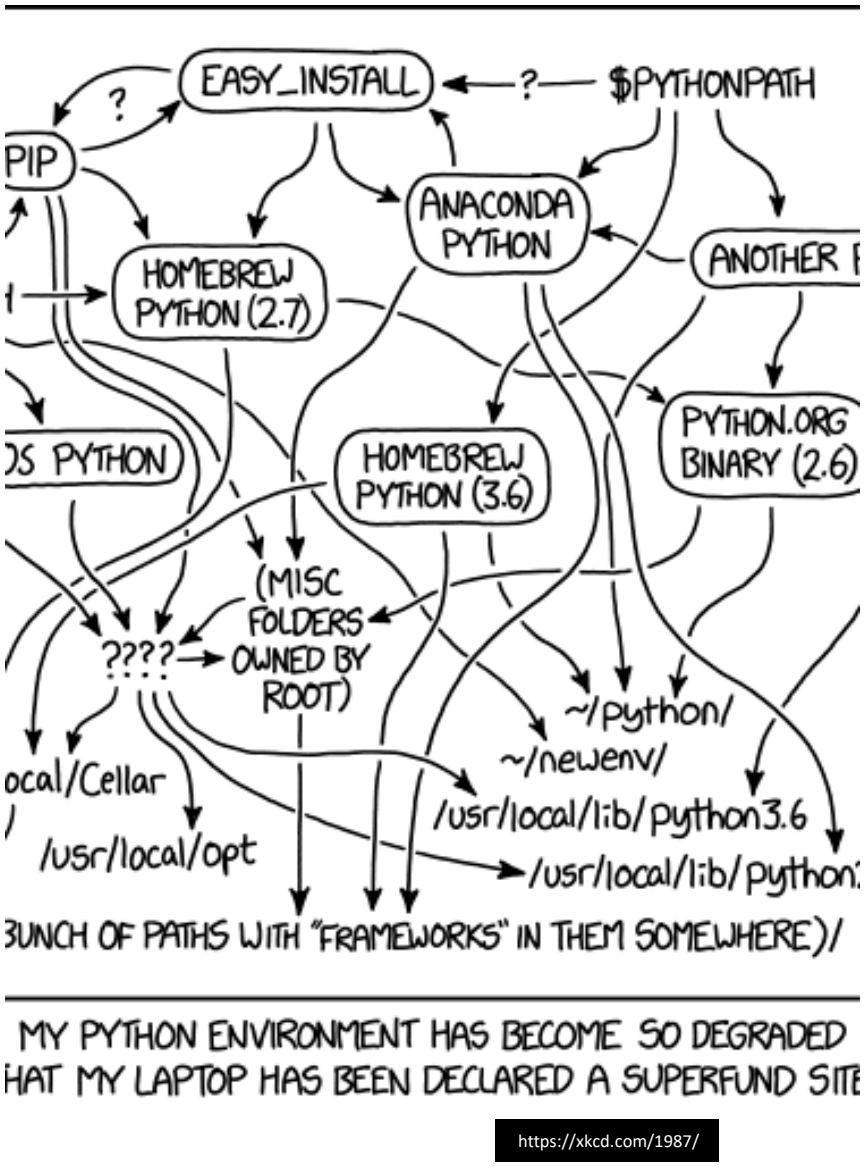
CREATE HELLO.PY

PRINT("HELLO, WORLD!")

RUN WITH RUN BUTTON OR TERMINAL

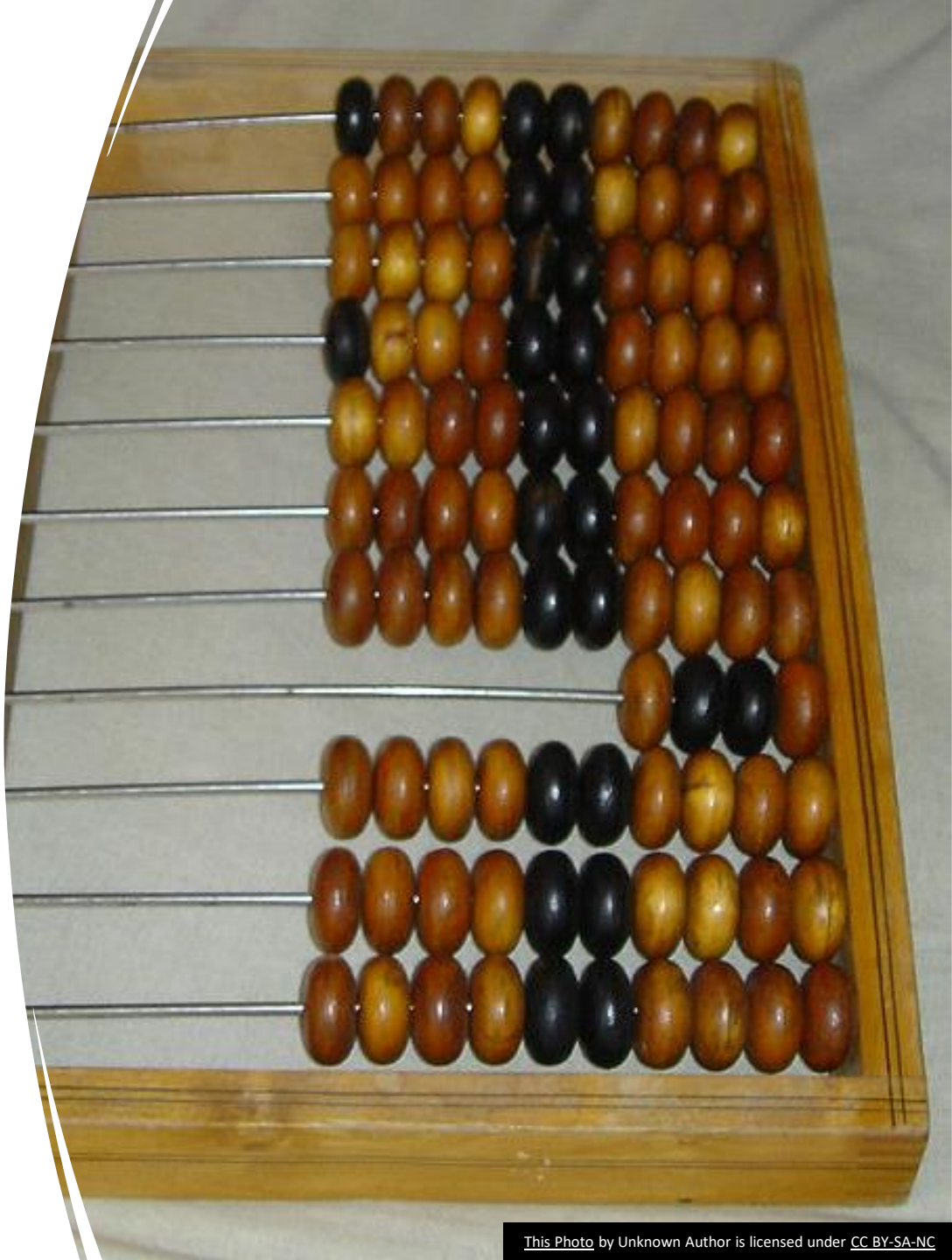
Virtual Environments (Optional)

- python -m venv venv
- venv\Scripts\activate
- Official Python venv tutorial: https://docs.python.org/3/tutorial/venv.html
- Real Python – Virtual Environments Primer: https://realpython.com/python-virtual-environments-a-primer/
- VS Code – Working with Python Environments: https://code.visualstudio.com/docs/python/environments

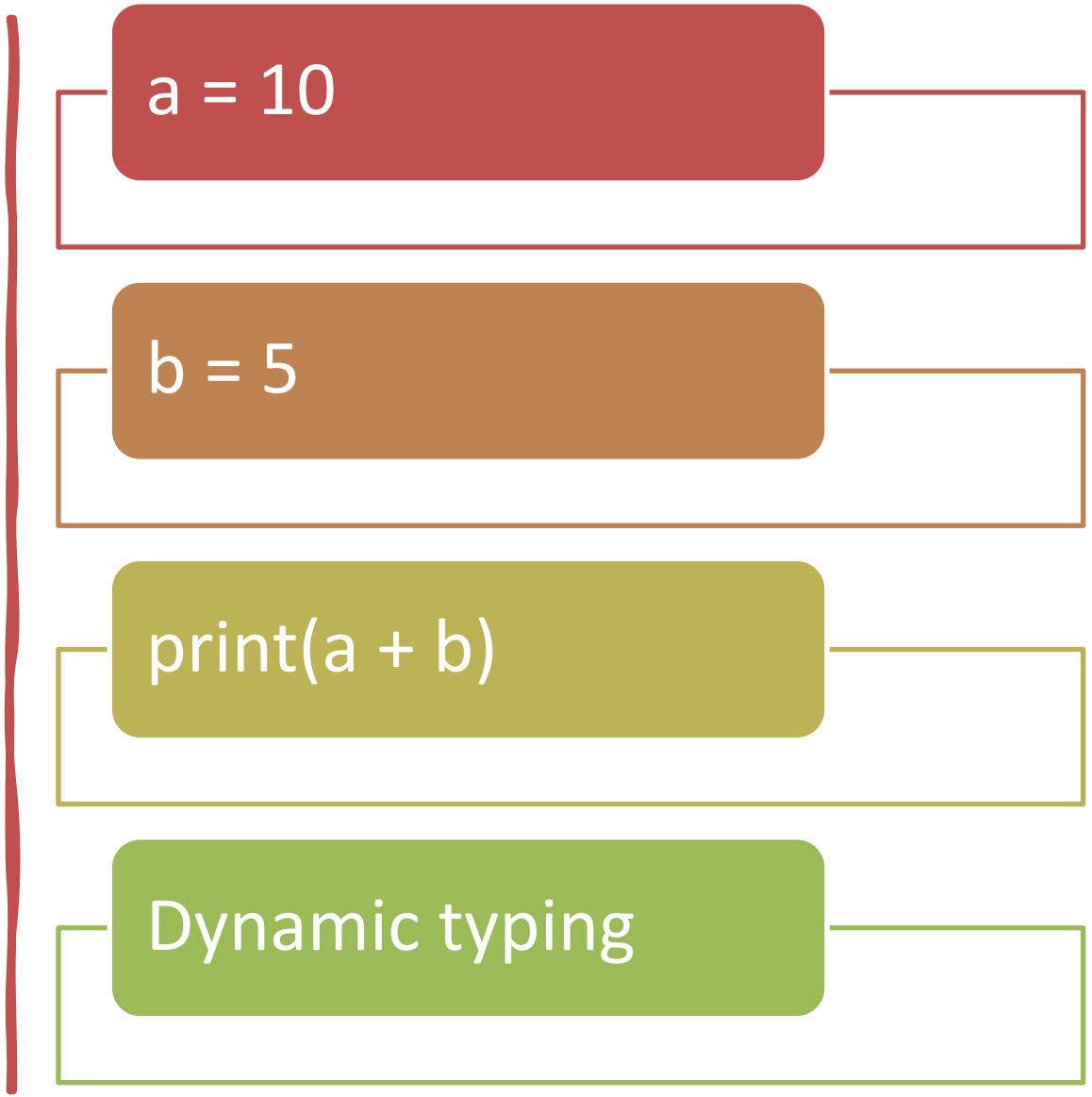


Python Arithmetic

- 3 + 2
- 7 - 3
- 4 * 5
- 8 / 2
- 20 % 7
- 2 ** 3
- BIG NUMBER SUPPORT

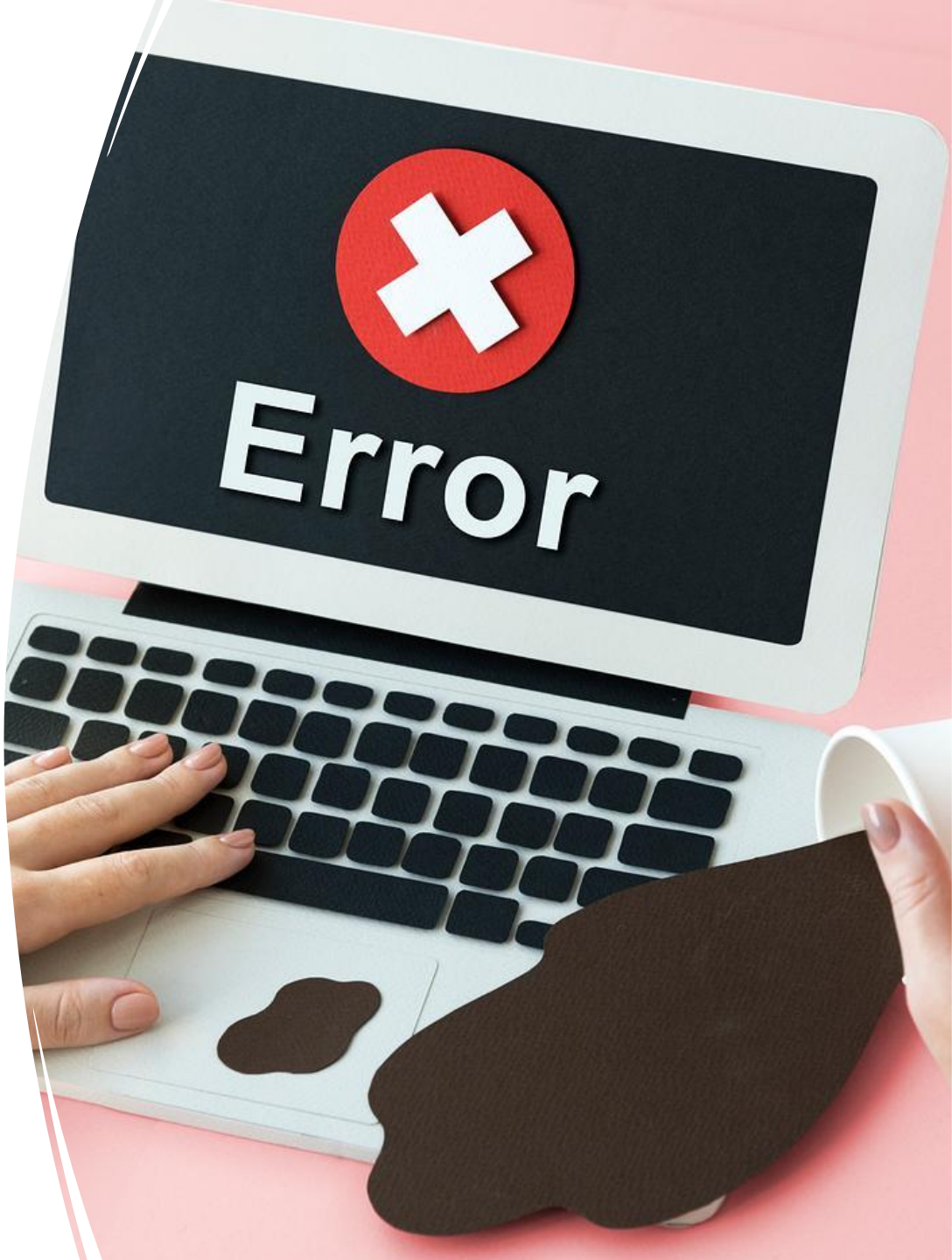


Python Variables



Python Errors

- SyntaxError
- IndentationError
- NameError
- Errors = Feedback



Mini Automation Teaser

```
NAME = INPUT('ENTER
YOUR NAME: ')
```

```
PRINT('HELLO,', NAME)
```

Potato Program

We want for user to input their favorite food

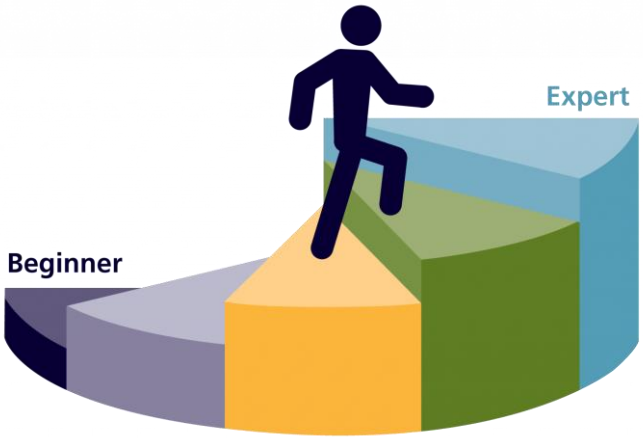
We want to ask for its price

We want to find out how much to buy

We want to calculate total

What to improve?

Beginner-Friendly Python Resources



- Official Python Tutorial: https://docs.python.org/3/tutorial/
- Automate the Boring Stuff (free online): https://automatetheboringstuff.com/
- Real Python Tutorials: https://realpython.com/
- W3Schools Python Basics: https://www.w3schools.com/python/
- Google Python Class (free): https://developers.google.com/edu/python/
- Python Cheatsheet: https://www.pythoncheatsheet.org/
- VS Code Python Tutorial: https://code.visualstudio.com/docs/python/python-tutorial

Not Covered in this Course but Very Useful: Git & GitHub

- Git = version control for tracking changes in code
- GitHub = platform for storing and collaborating on code
- Helps back up scripts and sync work between computers
- Track history, undo mistakes, compare versions
- Essential for teamwork and professional development

- Beginner Resources:
- • Download Git: https://git-scm.com/downloads
- • GitHub Getting Started Guide: https://docs.github.com/en/get-started
- • VS Code + Git Basics: https://code.visualstudio.com/docs/sourcecontrol/overview

What We Learned Today (Day 1)

- What automation means and why it's useful
- What Python is and how it came to be
- Python installation on Windows (with PATH)
- Installing and configuring VS Code
- VS Code essentials: Python extensions
- Creating and running Python scripts (.py)
- Running simple code: print, variables, basic arithmetic
- Overview of the full 10-part course structure
- Beginner resources (virtual environments, Git/GitHub)

What's Next

TEXT PROCESSING

READING &
WRITING FILES

HOMEWORK:
WRITE TINY SCRIPTS