MODULE 1: CORE PYTHON & DATA

WEEK: 1 LECTURE: 1

DATE: 15/08/2025

INSTRUCTOR: ORANGZAIB RAJPOOT

A LAUNCHPAD INTO PYTHON AND AI: YOUR FIRST STEPS

- **Welcome to the world of Python and Artificial Intelligence!** This presentation is designed to be your launchpad, whether you're taking your very first steps into programming or you're a seasoned developer from another language. Today, we'll journey through the fundamentals of Python, set up your coding environment, and write our first program. We'll also explore why Python is the undisputed language of choice for AI and get you started with some hands-on practice.

TODAY'S AGENDA

- **Introduction to Python:** A brief history and its modern-day applications.
- **Why Python is the Language of AI:** Exploring the reasons for its dominance.
- **Real-World Motivation:** Seeing how Python and AI are shaping our world.
- **Setting Up Your Lab:** A step-by-step guide to installing Python, VS Code, and setting up a virtual environment with Conda.
- **"Hello, World!": Your First Python Script:** We'll write and execute a simple program.
- **Hands-On Experience:** Time to get your hands dirty with coding exercises.

A BRIEF HISTORY OF PYTHON

- Python was conceived in the late 1980s by Guido van Rossum in the Netherlands. His goal was to create a language that was both powerful and easy to read, emphasizing code readability and simplicity. The name "Python" was inspired by the British comedy series "Monty Python's Flying Circus," reflecting the language's philosophy of making programming enjoyable.
- The first version, Python 0.9.0, was released in 1991. Over the years, Python has evolved through major versions, with Python 2.0 released in 2000 and the backward-incompatible Python 3.0 in 2008. Today, Python is maintained by the Python Software Foundation and has a massive global community of developers.

PYTHON'S MODERN USE CASES

Python's versatility has made it a powerhouse in numerous fields:

- **Web Development:** Frameworks like Django, Flask and Frappe allow for the creation of robust and scalable web applications.
- **Data Science and Analytics:** Libraries such as Pandas, NumPy, and Matplotlib are the go-to tools for data manipulation, analysis, and visualization.
- **Automation and Scripting:** Python is perfect for automating repetitive tasks, from file management to sending emails.
- **Cybersecurity:** Professionals use Python for tasks like penetration testing and vulnerability scanning.
- **Game Development:** While not the primary language for high-end games, libraries like Pygame are popular for creating simpler games and prototypes.

WHY PYTHON IS THE LANGUAGE OF AI

Python's dominance in the world of Artificial Intelligence is no accident. Here's why it has become the preferred language for AI and Machine Learning:

- **Simplicity and Readability:** Python's clean and intuitive syntax resembles plain English, allowing developers to focus on complex AI algorithms rather than getting bogged down by complicated code. This readability also makes it easier for teams to collaborate on projects.

- **A Rich Ecosystem of Libraries and Frameworks:** Python boasts a vast collection of specialized libraries for AI and machine learning. Key libraries include:
  - **TensorFlow, PyTorch, and Keras:** For building and training deep learning models.
  - **Scikit-learn:** For various machine learning tasks like classification, regression, and clustering.
  - **NumPy and Pandas:** For efficient data manipulation and analysis, which is a crucial first step in any AI project.
- **Strong Community Support:** Python has a massive and active global community. This means that if you encounter a problem, there's a high chance that someone has already solved it and shared the solution.
- **Open-Source Nature:** Python and its libraries are free to use, which is a significant advantage for everyone from students to startups.
- **Versatility:** Python's ability to be used for a wide range of tasks, from web scraping for data collection to deploying AI models in web applications, makes it an incredibly versatile tool.

REAL-LIFE MOTIVATION: PYTHON AND AI IN ACTION

Now that you know *why* Python is the language of choice for AI, let's look at some inspiring real-world examples of what you can eventually build. These applications are powered by the very tools and concepts you are about to learn.

- **Netflix's Recommendation Engine:** Netflix uses machine learning algorithms, largely built with Python, to analyze your viewing habits and recommend movies and TV shows you might like.
- **Social Media Content Moderation:** Platforms like Instagram and Twitter use Python-based AI to automatically detect and filter out spam and inappropriate content.
- **Personalized Finance:** Financial institutions use Python to build models that can predict stock market trends and help manage investments.
- **Robotics:** Companies like Boston Dynamics use Python to test the functionality and movements of their advanced robots.
- **Smart Assistants:** Voice assistants like Siri and Alexa use Natural Language Processing (NLP), a field of AI where Python is a key tool, to understand and respond to your voice commands.

SETTING UP YOUR DEVELOPMENT ENVIRONMENT

- Let's get your tools ready. We'll be using Visual Studio Code (VS Code), a popular and powerful code editor, along with Python itself.
- **Step 1: Install Python**
- Head over to the official Python website: https://www.python.org/downloads/
- Download the latest stable version for your operating system (Windows, macOS, or Linux).
- During installation on Windows, **make sure to check the box that says "Add Python to PATH."** This is a crucial step.

- To verify your installation, open a terminal (or Command Prompt on Windows) and type python3 --version (for macOS/Linux) or python --version (for Windows). You should see the Python version number displayed.
- **Step 2: Install Visual Studio Code**
- Go to the official VS Code website: https://code.visualstudio.com/download
- Download and install the version for your operating system.

- **Step 3: Install the Python Extension for VS Code**
- Open VS Code.
- Click on the Extensions icon in the Activity Bar on the side of the window (it looks like four squares).
- Search for "Python" and install the extension provided by **Microsoft**. This will provide you with features like code highlighting, autocompletion (IntelliSense), and debugging.

VIRTUAL ENVIRONMENTS WITH CONDA

Why Do We Need Virtual Environments?

- Imagine you are working on two different projects. Project A needs version 1.0 of a specific library, but Project B needs version 2.0 of the *same* library. If you install these system-wide, you'll have a conflict.
- A **virtual environment** is an isolated, self-contained directory that holds a specific version of Python plus all the necessary packages for a particular project. This prevents conflicts and keeps your projects organized.
- **Conda** is an open-source package and environment management system. It's extremely popular in the data science and AI community. **Miniconda** is a smaller, bootstrap version of Anaconda that includes just Conda, Python, and the packages they depend on. We'll use Miniconda to keep our setup lightweight.

HOW TO INSTALL AND USE MINICONDA

- **Download Miniconda:**
  - Go to the Miniconda documentation: https://docs.conda.io/en/latest/miniconda.html
  - Download the latest Python 3 installer for your operating system (Windows, macOS, or Linux).
- **Install Miniconda:**
  - Run the installer you downloaded. It's recommended to accept the default settings.
  - After installation, close and reopen your terminal for the changes to take effect.
- **Create Your First Environment:**
  - Open your terminal (or "Anaconda Prompt" if you're on Windows).
  - Let's create an environment for this course called ai-course. We will specify the Python version we want to use (e.g., 3.9). Type the following command and press Enter:
  - `conda create --name ai-course python=3.9`
  - Conda will show you what will be installed and ask you to proceed (y/n). Type y and press Enter.

- **Activate Your Environment:**
  - To start using the environment, you need to "activate" it. In the same terminal, run:
  - `conda activate ai-course`
  - You'll notice your terminal prompt changes to show (ai-course) at the beginning. This indicates that the environment is active. Any Python packages you install now will be placed inside this environment, separate from your main system.
- Deactivate an Environment:
  - When you're done working, you can deactivate the environment with:
  - `conda deactivate`

YOUR FIRST PYTHON SCRIPT: "HELLO, WORLD!"

It's a tradition in programming to start with a "Hello, World!" program.

- **Create a Project Folder:** On your computer, create a new folder for your Python projects.
- **Open the Folder in VS Code:** In VS Code, go to File > Open Folder and select the folder you just created.
- **Create a New File:** In the VS Code Explorer on the left, click the "New File" icon and name your file hello.py. The .py extension is important.
- **Write the Code:** In the hello.py file, type the following single line of code:
  - `print("Hello, World!")`
- **Run Your Script:**
  - Make sure your ai-course environment is active in the VS Code terminal.
  - Click the "Run Python File" play button in the top-right corner of the editor.

EXERCISE 1: PERSONALIZED GREETING

- Modify your hello.py script. Create a variable to store your name.
- Print a personalized greeting using that variable, like "Hello, [Your Name]!".

EXERCISE 2: FUN WITH VARIABLES

- Create a new file named variables.py.
- Create a variable for your age (an integer) and a variable for your favorite hobby (a string).
- Print a sentence that combines these, for example: "I am 25 years old and I love to go hiking."

EXERCISE 3: SIMPLE CALCULATOR

- Create a new file called calculator.py.
- Create two variables, num1 and num2, and assign them numeric values.
- Write code to add, subtract, multiply, and divide these two numbers.
- Print the result of each operation with a descriptive label, like "Sum: 15".

EXERCISE 4: USER INPUT

- Create a new file user_greeting.py.
- Use the input() function to ask the user for their name and store it in a variable.
- Use the input() function again to ask for their city.
- Print a friendly message that includes both their name and city.

EXERCISE 5 (CHALLENGE): AREA OF A RECTANGLE

- Create a file named rectangle_area.py.
- Ask the user to input the width of a rectangle.
- Ask the user to input the height of a rectangle.
- **Hint:** The input() function returns a string. You will need to convert the user's input into a number (e.g., an integer) using int() before you can do math with it.
- Calculate the area (width * height) and print the result.