

# Python Course Glossary

---

## AI

AI, short for Artificial Intelligence, is the technology that enables machines to perform tasks that typically require human intelligence. In the context of coding, understanding AI allows you to create programs that can learn from data, recognize patterns, and make decisions. Learning AI can enhance your productivity and open up new possibilities in various fields like business, research, and technology.

Introduced in: [Course\\_Introduction.txt](#)

## Jupyter Notebook

A Jupyter Notebook is a coding environment used by programmers and data scientists. It allows you to write and run code in sections called cells. In Lesson 3, the instructor demonstrates running code in a Jupyter Notebook by using the Shift Enter command to execute a line of code, showing "hello world" as an output.

Introduced in: [Lesson\\_3.txt](#)

## Python

Python is a popular programming language used to write instructions for computers to perform tasks. Learning Python can help you automate tasks, analyze data, and develop AI applications. It is widely used in various fields like AI, self-driving cars, and web development. Python has a supportive developer community and is a great language to start learning programming.

Introduced in: [Lesson\\_1.txt](#)

## Shift Enter

In Lesson 3, you learned about Shift Enter in a Jupyter Notebook, a key command in running code. By pressing and holding Shift, then hitting Enter, you can execute a line of code. For instance, running "print('hello world')" using Shift Enter will display "hello world" as the output. Practice using Shift Enter to run code smoothly.

Introduced in: [Lesson\\_3.txt](#)

## automation

Automation in programming refers to using computer programs to perform tasks automatically without human intervention. For example, writing a program to monitor a calendar events and receiving notifications if there are any changes. Automation helps streamline processes, save time, and improve efficiency in various fields like business analysis, AI development, and data analysis.

Introduced in: [Lesson\\_1.txt](#)

## bug

---

A 'bug' in programming is an error or mistake in the code. When a program has a bug, it may not run correctly and can produce error messages. Chatbots can help identify and fix bugs by providing explanations and corrections. It's common for programmers to make mistakes while coding, and finding and fixing bugs is a normal part of programming.

Introduced in: [Lesson 4.txt](#)

## chatbot

---

A chatbot is a computer program that simulates human conversation through text or speech. In Lesson 4, the chatbot was used to help learners understand and practice writing Python code. For example, using a chatbot to modify code to output personalized messages like "Hello, Andrew," and to debug errors in code.

Introduced in: [Lesson 4.txt](#)

## code execution

---

Code execution refers to the process of running and carrying out the instructions written in a program. In Lesson 3, using Jupyter Notebook, code is written and executed by pressing Shift + Enter. For example, typing and running `print("hello world")` in the notebook will display "hello world" as the output.

Introduced in: [Lesson 3.txt](#)

## coding

---

Coding is the practice of writing instructions for computers to follow. In the lesson 'Course Introduction', it was mentioned that learning to code has become more valuable due to the rise of AI tools like ChatGPT. Basic coding skills can enhance productivity in various job roles and open up new possibilities for problem-solving and automation.

Introduced in: [Course Introduction.txt](#)

## coding environment

---

A coding environment is where you write and run your code. In Lesson 3, a Jupyter Notebook was introduced as the coding environment used in the course. It's a common tool among professional programmers and data scientists. You can write code, execute it, and see the output within this environment. An essential command you'll use is Shift Enter to run your code.

Introduced in: [Lesson 3.txt](#)

## comment

---

In programming, a comment is a text in the code that is ignored by the computer when running the program. Comments are used to explain or remind developers about the code. They can be helpful in understanding the purpose of the code or for leaving notes for oneself or others reading the code. An example of a comment in Python is using the pound sign (#).

For instance, in the lesson, the instructor demonstrates how comments are written in a Jupyter Notebook cell by using the pound sign (#) followed by text to describe the code. When the cell is run, the comments are ignored, and the computer does not execute them. Comments are essential for making code more readable and maintainable.

Introduced in: [Lesson 4.txt](#)

## computer programming

---

Computer programming involves writing precise instructions to tell a computer what tasks to perform. By learning how to code, you can command machines to automate tasks, analyze data, and gain new insights. For example, a program can help you monitor schedules, navigate GPS, or develop AI applications. Python is a popular language used for AI, self-driving cars, chatbots, and more.

Introduced in: [Lesson 1.txt](#)

## computer programs

---

Computer programs are detailed sets of instructions that tell a computer how to perform a specific task. Similar to following a cooking recipe to create a dish, programs guide computers through steps to achieve desired outcomes. They enable automation of tasks, data analysis, and insight generation. Learning programming empowers you to make computers work for you efficiently.

Introduced in: [Lesson 1.txt](#)

## copy and paste

---

In Lesson 3, you learned about copying and pasting code in the coding environment called Jupyter Notebook. When you want to run the code you copied, you can paste it by using Command V or Control V, depending on your operating system. Remember, pressing Shift and Enter together is a key command to run the code in Jupyter Notebook.

Introduced in: [Lesson 3.txt](#)

## data

---

Data in programming refers to pieces of information that can be processed by a computer. In Python, data can be text or numbers. Text data is represented as strings, enclosed in double quotation marks. Numbers can be integers (whole numbers) or floats (numbers with decimal points). Python can perform calculations and manipulate different types of data.

Introduced in: [Lesson 5.txt](#)

## f strings

---

F strings, short for formatted strings, allow you to embed expressions like calculations or variables within strings in Python. By placing an "F" before the string and using curly braces {} to wrap the expressions, you can incorporate dynamic content into your output. For example, f strings let you easily display results like temperatures converted between Fahrenheit and Celsius or calculate someone's "dog age" based on their human age. This makes your code more readable and flexible for various outputs.

Introduced in: [Lesson\\_6.txt](#)

## floats

---

Floats in Python are used to represent decimal numbers. They are a data type that can store numbers with a fractional component. For example, if you have a temperature of 23.889 degrees Celsius, the decimal part (.889) is stored as a float in Python. Floats are handy for dealing with calculations that involve numbers beyond whole integers.

Introduced in: [Lesson\\_6.txt](#)

## instructions

---

Instructions in computer programming are precise commands given to a computer to perform tasks. Similar to following a cooking recipe step by step, a computer program guides the computer on what actions to take. For example, writing code to monitor a calendar or automate repetitive tasks are practical uses of instructions in programming.

Introduced in: [Lesson\\_1.txt](#)

## integers

---

Integers are whole numbers in programming that do not have any decimal points. In Python, you can use integers to perform calculations like addition, subtraction, multiplication, and division. For example, in Lesson 6, we saw how to use integers to convert temperatures from Fahrenheit to Celsius or calculate someone's dog age based on their real age.

Introduced in: [Lesson\\_6.txt](#)

## knowledge work

---

Knowledge work refers to tasks that involve using information and creativity to solve problems. In programming, knowing how to write code can make you more effective in various professions such as journalism, marketing, and business research. Learning just a bit of code can enhance your productivity and open up new opportunities in the AI-driven workplace.

Introduced in: [Course\\_Introduction.txt](#)

## order of operations

---

The order of operations in Python determines the sequence in which calculations are performed. Just like in regular math, Python follows the rule of performing multiplication and division before addition and subtraction. To avoid mistakes, you can use parentheses to specify the order of operations. For example, in converting Celsius to Fahrenheit, the correct order is to subtract 32 first, then multiply by five over nine.

Introduced in: [Lesson 5.txt](#)

---

## print

In programming, the 'print' function is used to display information on the screen. In Lesson 2, the chatbot generated code that included the 'print' function to show the message 'Hello World'. For example, using 'print("Hello World")' in Python would output 'Hello World' when the code is run.

Introduced in: [Lesson 2.txt](#)

---

## productivity

Productivity in programming refers to being able to achieve more in less time by utilizing coding skills effectively. By learning just a little bit of code, individuals in various job roles can streamline tasks and boost efficiency. For example, knowing how to write code can greatly enhance productivity in knowledge work and other areas.

Introduced in: [Course Introduction.txt](#)

---

## programming

Programming involves writing instructions for computers to follow using a specific language, like Python. In Lesson 2, we saw how chatbots can help answer programming questions and even generate simple code snippets, like a "Hello, World!" program. Chatbots make learning and writing code more accessible and efficient.

Introduced in: [Lesson 2.txt](#)

---

## programming language

A programming language is a set of rules and symbols that allow us to communicate instructions to a computer. By using a programming language, we can create software, applications, and websites. Learning a programming language enables us to automate tasks, solve problems, and tap into the full potential of technology.

Introduced in: [Course Introduction.txt](#)

---

## prompt

In programming, a 'prompt' refers to a question or a command that is given to a chatbot or a program to generate a response. For example, typing "What is Python?" into a chatbot would be considered a prompt. Chatbots use prompts to provide answers or complete tasks based on the input they receive.

Introduced in: [Lesson 2.txt](#)

## running a program

---

Running a program involves executing the code you've written in Python. In Lesson 4, you were encouraged to run simple programs like printing "Hello world" or customizing messages. Running a program helps you practice coding and learn from mistakes. Comments in the code are notes ignored by the computer, used for explanations or reminders.

Introduced in: [Lesson\\_4.txt](#)

## strings

---

Strings in Python represent text and can store characters, words, or sentences. In Lesson 6, we learned about formatted strings or f-strings, which allow us to combine text with variables or calculations. By placing curly braces {} within an f-string, we can efficiently insert values or results into our strings, making them dynamic and informative.

Introduced in: [Lesson\\_6.txt](#)

## syntax

---

Syntax in programming refers to the set of rules that define the combinations of symbols and words that are considered valid in a programming language. For example, in Python, using proper syntax is important to ensure that your code runs correctly. Python's simple syntax makes it a great choice for beginners.

Introduced in: [Lesson\\_2.txt](#)