

Python Course Glossary

AI tools

AI tools like ChatGPT and generative AI make coding easier by acting as helpful companions, reducing the effort needed to learn code. These tools can assist in various fields, enhancing productivity and making coding more fun and accessible. (From 'Course Introduction')

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

Bug

A bug is an error in your code that causes it to not work as expected. In Lesson 4, we saw an example where a missing quotation mark caused a bug. Such issues can be fixed by identifying the error and correcting the code.

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

Comment

In Lesson 4, a 'comment' in Python starts with a hash sign (#) followed by text. The computer ignores comments, which are used to explain code and make it easier to understand. Example: `# This is a comment`.

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

Jupyter Notebook

A Jupyter Notebook is a coding environment used in Lesson 3 that lets you write and run code interactively. You can paste code, run it by pressing Shift + Enter, and see the output immediately. It's a tool used widely by both programmers and data scientists.

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

Python

Python is a beginner-friendly, high-level programming language with simple syntax. In Lesson 2, we discussed using chatbots to write Python code, like the classic "Hello, World!" program, and even more complex tasks like calculating days between dates.

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

Shift Enter

'Shift Enter' in Lesson 3 is the key combination used in Jupyter Notebooks to run a line of code. Just hold Shift and press Enter to execute the code you've written. For example, pressing Shift Enter after typing

`print("Hello, World!")` runs this command immediately.

Introduced in: [Lesson_3.txt](#)

automation

Automation in programming is about creating scripts to perform repetitive tasks efficiently. In Lesson 1, an example is given where an executive assistant writes a program to monitor a calendar, showing how automation can make tasks easier and more accurate.

Introduced in: [Lesson_1.txt](#)

calculations

In Lesson 6, 'calculations' involve using Python to perform math operations, such as converting temperatures or ages. The lesson demonstrates using f strings (formatted strings) to combine calculations and text in print statements, like converting 75°F to 23.889°C. This makes displaying results straightforward and readable.

Introduced in: [Lesson_6.txt](#)

chatbot

In Lesson 2, we learn how chatbots can be valuable tools for programmers. These AI chatbots can quickly answer questions, write simple code snippets like the "Hello, World!" program, and help modify or generate new code. This improves productivity and learning efficiency.

Introduced in: [Lesson_2.txt](#)

coding environment

A 'coding environment' is a space where you write and run your code. In Lesson 3, the course uses Jupyter Notebook, highlighted for its middle area where you enter code, and the Shift+Enter command to execute it.

Introduced in: [Lesson_3.txt](#)

copy and paste

In Lesson 3, copying and pasting code involves selecting the code, copying it (using a button or Command/Control + C), and then pasting it into the Jupyter Notebook's coding area (using Command/Control + V) to run it. This is essential for executing code examples provided in the course.

Introduced in: [Lesson_3.txt](#)

data

In Lesson 5, data is introduced as any information that can be processed by a computer. Examples include text, numbers, images, and sounds. Python handles these using different data types such as strings for text and integers or floats for numbers.

Introduced in: [Lesson_5.txt](#)

f strings

F strings in Python, introduced in Lesson 6, make it easy to embed calculations and variables directly within strings. By prefixing a string with `f` and using curly braces `{}`, Python evaluates any expressions inside the braces. For example, `f"{75 * 9/5 + 32} F"` would output "167.0 F".

Introduced in: [Lesson 6.txt](#)

float

A 'float' is a data type in Python for numbers with decimal points, like 3.14 or 2.99. In Lesson 5, you learned that floats differ from integers, which are whole numbers without decimals. This allows precise calculations in Python.

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

integer

An integer is a type of number in Python that has no decimal part. Examples from Lesson 5 include 42 and 100. They are distinct from floats, which are numbers with decimal points, such as 3.14.

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

knowledge work

Knowledge Work involves jobs that primarily require handling and processing information, such as journalism, marketing, or business research. In the "Course Introduction," it's highlighted that coding skills can significantly boost productivity in these roles, especially with the aid of AI tools like ChatGPT.

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

order of operations

In Lesson 5, "order of operations" means the sequence in which Python performs math operations, like addition, subtraction, multiplication, and division. It's the same as in regular math. For example, `(75 - 32) * 5/9` correctly converts Fahrenheit to Celsius, because parentheses ensure subtraction happens first.

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

productivity enhancer

A productivity enhancer in coding, introduced in 'Course Introduction', means using just a little bit of code to significantly improve efficiency in various job roles, especially with AI tools like ChatGPT acting as supportive companions in the coding process.

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

programming language

A programming language is a set of instructions used to communicate with computers. In the 'Course Introduction', it's highlighted that learning even a little bit of coding can significantly enhance productivity in various job roles, especially with the support of AI tools like ChatGPT.

Introduced in: [Course_Introduction.txt](#)

prompt

In Lesson 2, a "prompt" is a question or request you type into a chatbot to get information or a response. For example, typing "What is Python?" into the chatbot allows you to receive an explanation about the Python programming language.

Introduced in: [Lesson_2.txt](#)

syntax

In Lesson 2, 'syntax' is introduced as the set of rules that define the structure of a programming language. For instance, Python's simple syntax makes it easier for beginners to learn and use, like using the `print("Hello World")` statement to display text.

Introduced in: [Lesson_2.txt](#)

variables

In Lesson 6, variables are introduced as a fundamental concept in programming. They store data (like numbers or text) so you can use and manipulate it later. For example, instead of repeating "Isabel" and "28" multiple times, you can store them in variables to make your code cleaner and easier to manage.

Introduced in: [Lesson_6.txt](#)