COURSE GLOSSARY

Intermediate Python for Developers

**kwargs (arbitrary keyword arguments): A convention where two asterisks before a parameter collect extra keyword arguments into a dictionary, enabling flexible named inputs to a function

*args (arbitrary positional arguments): A convention where a single asterisk before a parameter collects any extra positional arguments into a tuple, allowing functions to accept a variable number of inputs

Built-in function: A function provided by Python's standard library that you can call directly without importing a module, such as len(), sum(), or print(), to perform common tasks

DataFrame: A two-dimensional, labeled data structure provided by pandas that stores tabular data in rows and columns, similar to a spreadsheet or SQL table

Default argument: A parameter in a function definition that has a predefined value used when the caller does not supply that argument

Docstring: A short string placed at the start of a function, module, or class that documents what it does and how to use it, accessible via help() or the __doc__ attribute

DRY (Don't Repeat Yourself) principle: A software design guideline encouraging reuse of code and abstraction to avoid duplication and reduce maintenance effort

Hallucination: When a model produces confident but incorrect or fabricated information, often due to gaps or biases in its training data or reasoning process

Hallucination: When a model produces confident but incorrect or fabricated information, often due to gaps or biases in its training data or reasoning process

Function: A named block of reusable code defined with def that performs a specific task and can accept inputs (arguments) and return outputs

import: A Python statement that brings a module or package into your current script so you can use its functions, classes, and attributes

KeyError: A built-in exception raised when attempting to access a mapping (like a dictionary or DataFrame column) using a key that does not exist

Keyword argument: An argument passed to a function by explicitly naming the parameter (name=value), which improves clarity and allows arguments in any order

Lambda function: A short anonymous function defined with the lambda keyword that contains arguments, a colon, and a single expression and is useful for simple, inline operations

map(): A built-in function that applies a provided function to every item of an iterable and returns an iterator of the results

Method: A function that is associated with a specific data type or object and is called using dot notation, such as DataFrame.head()

Module: A single Python file (.py) that contains reusable code like functions, classes, and variables which can be imported into other scripts

Package: A collection of related Python modules organized in a directory (often distributed as a library) that can be installed and imported to add functionality to projects

pandas: A popular third-party Python package for data manipulation and analysis that provides powerful data structures like DataFrame for working with tabular data

pip: The standard package installer for Python used from the command line to download and install packages from PyPI into your environment

Positional argument: An argument passed to a function based on its position in the call, where the order of values determines which parameter they match

PyPI (Python Package Index): The central public repository of Python packages where developers publish libraries that others can install and use

raise: A statement used to intentionally trigger an exception, optionally with a custom message, to signal an error condition or enforce input validation

Traceback: The detailed report Python prints when an exception occurs that shows the call stack and the location(s) in code where the error propagated

try-except block: A control-flow structure that attempts to run code in the try section and lets you handle specific exceptions in one or more except clauses to prevent program termination

TypeError: A built-in exception raised when an operation or function is applied to an object of inappropriate type, like adding a string to an integer

ValueError: A built-in exception raised when a function receives an argument of the right type but an inappropriate value, such as converting "hello" to float