

## COURSE GLOSSARY

# Introduction to Functions in Python

**Argument:** The actual value passed to a function parameter when the function is called

**Built-in scope:** The namespace provided by Python that contains pre-defined names like `print` and `sum` which are available everywhere unless shadowed

**Closure:** A function object that remembers values from its enclosing lexical scope even when that enclosing function has finished executing

**def keyword:** The Python keyword used to begin a function definition, followed by the function name, parameters in parentheses, and a colon

**Default argument:** A parameter value provided in the function header that is used if the caller does not supply that argument

**Docstring:** A short, triple-quoted string placed immediately after a function header that documents what the function does, its parameters, and its return values

**Exception:** An error detected during program execution that interrupts normal flow and can be handled using error-handling constructs like `try-except`

**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 header:** The first line of a function definition that includes the `def` keyword, the function name, and any parameters in parentheses

**Function:** A named block of reusable code that performs a specific task when called and may accept inputs and return outputs

**global keyword:** A statement used inside a function to indicate that a name refers to the module-level global variable, allowing assignments to change that global value

**Global scope:** The top-level scope of a script or module where names defined outside functions are accessible throughout the module

**Immutable:** A property of an object that prevents modification of its contents after creation, as seen with tuples and strings in Python

**Keyword arguments (`**kwargs`):** A convention using a double asterisk before a parameter name to collect extra named arguments into a dictionary inside the function

**Lambda function:** A small anonymous function defined with the `lambda` keyword that can take arguments and return the result of a single expression

**LEGB rule:** The order Python follows to resolve name references: Local, Enclosing (for nested functions), Global, then Built-in

**List:** An ordered collection type in Python defined with square brackets that can hold multiple values and is mutable, meaning its contents can be modified

**Local scope:** The scope that exists inside a function where names defined within that function are accessible only during its execution

**Nested function:** A function defined inside another function, which can access names in its enclosing function's scope

**nonlocal keyword:** A statement used inside a nested function to indicate that a name refers to a variable in an enclosing (but non-global) scope, allowing that variable to be reassigned

**Parameter:** A named variable listed in a function's definition that specifies the input the function expects

**return:** A statement inside a function that sends a value (or values) back to the caller and ends the function's execution

**Scope:** The region of a program where a name (such as a variable or function) is visible and can be accessed

**Tuple:** An ordered collection type in Python defined with parentheses that can hold multiple values and is immutable, meaning its contents cannot be changed after creation