Introduction to Programming Concepts

- Programming
- Compilers vs Interpreters
- Using an Input, Process, Output programming method
- Using Python
  - Variables and Simple Data Types
  - Strings
  - Numbers
  - Constants and Comments
  - Input function
  - Modulo Operator

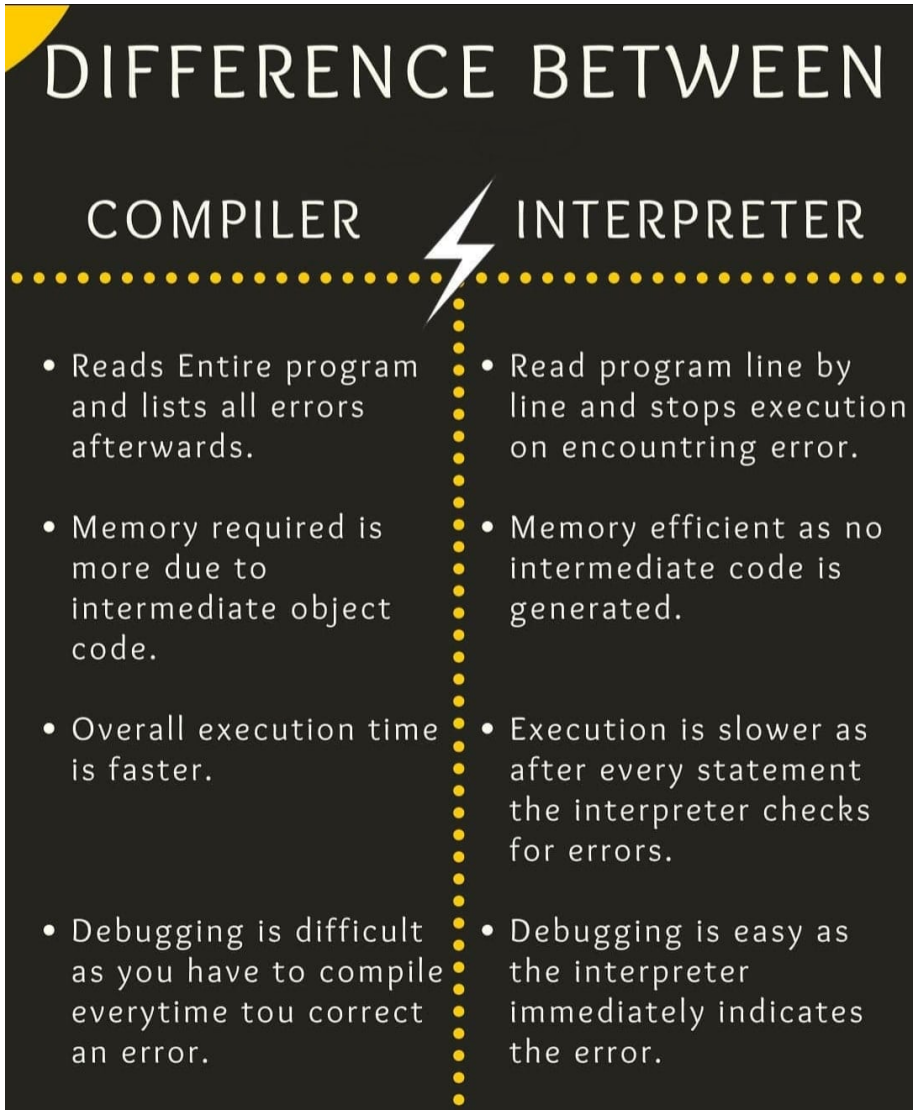
Programming

- Computers must be programmed to be of any value
- A Program is a sequence of stored instructions that a computer follows to perform a task
  - Commonly referred to as *Software or Algorithm*
- A Programmer is a person who can design, create, debug, and test computer programs
- Sequential or Procedural code is when the code block executes from top to bottom.
- Conditional code is when questions are asked of the program to determine flow in the form of if-else statements.
- Repetitive code is when the code block executes over and over until it is finished with the use of loops.

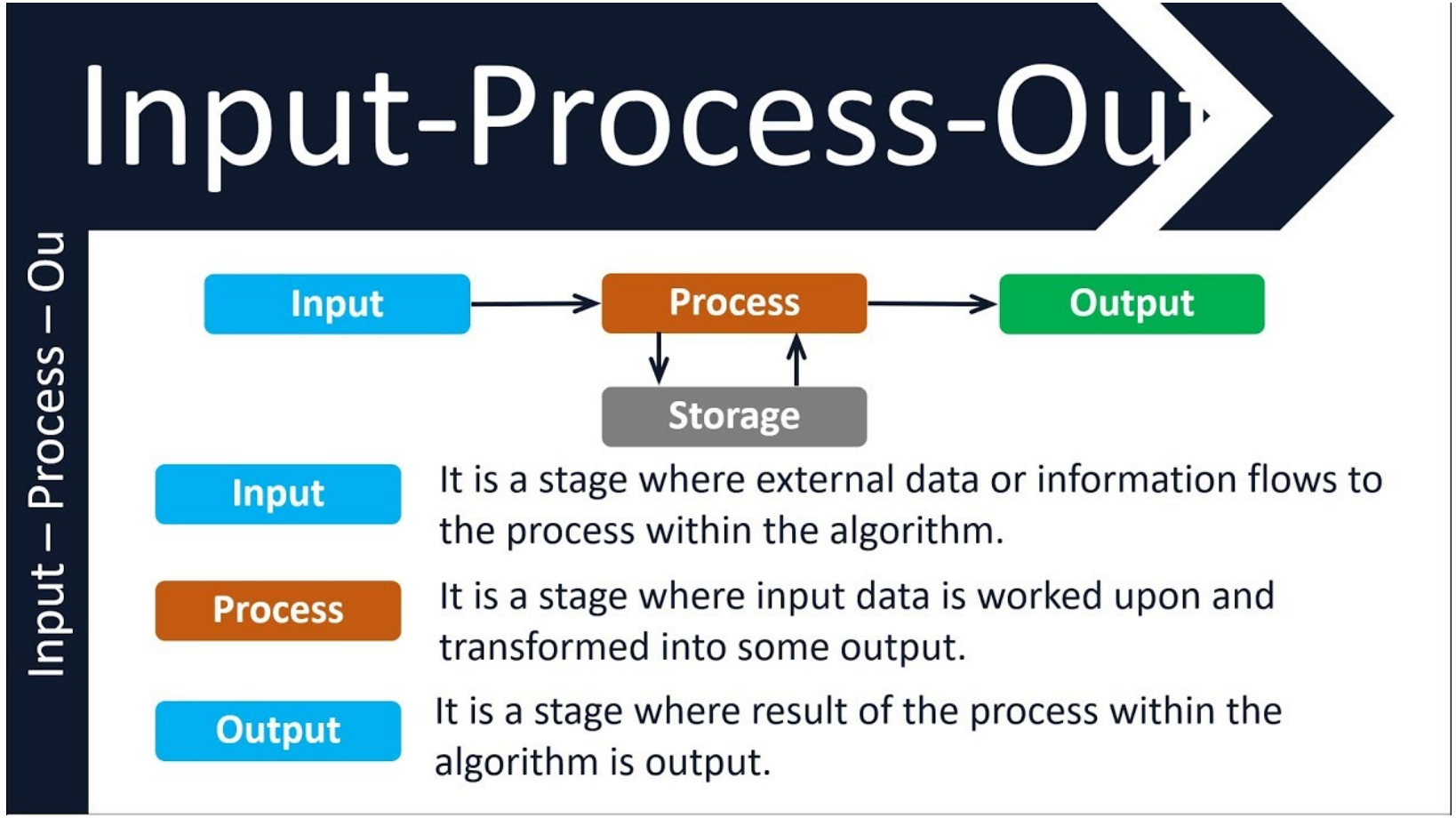
Compilers and Interpreters

- Programs written in high-level languages must be translated into machine language to be executed
- A Compiler translates high-level language program into separate machine language program ie C, C++, C#, Java
  - Machine language program can be executed at any time
  - Generates executable file (or in some cases DLL (dynamically linked library file))
- An Interpreter translates and executes instructions in high-level language program like Python
- Source code is the statements that programmers write and pass to the compiler or interpreter
- Syntax errors occur when Python doesn’t recognize a section of your program as valid Python code.

Compiler vs Interpreter



Input, Process, Output Method of Programming

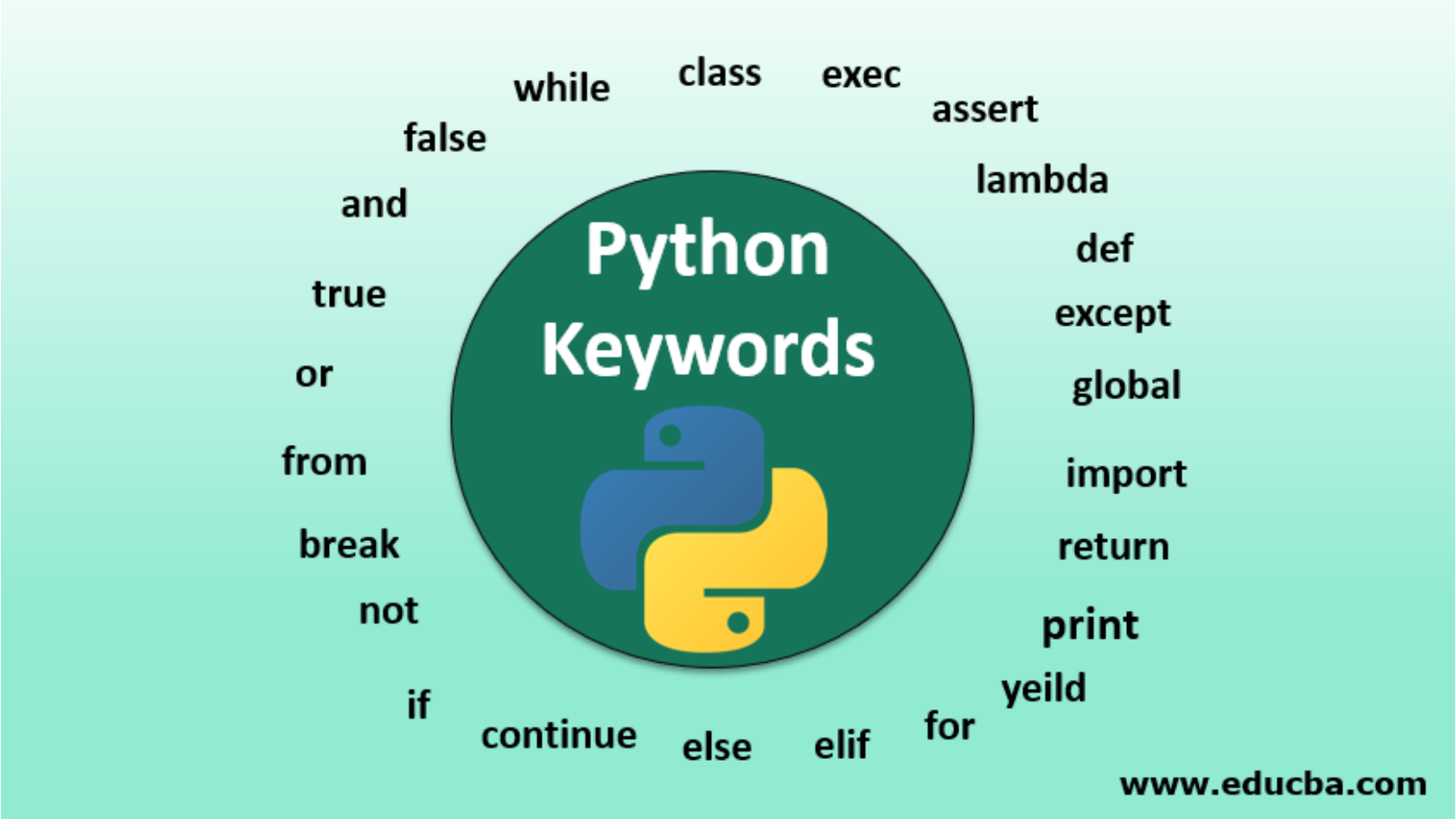


Using Python

- Python must be installed and configured prior to use
  - One of the items installed is the Python interpreter
- Python interpreter can be used in two modes:
  - Interactive mode: enter statements on keyboard
    - enter commands and source code at the chevron >> (or prompt)
  - Script mode: save statements in Python script
    - Source code files with a .py extension
    - Run in IDE (integrated development environment
    - Can also be ran at chevron >> (prompt)
- Python was written in C

Python Features

Python Reserve Words



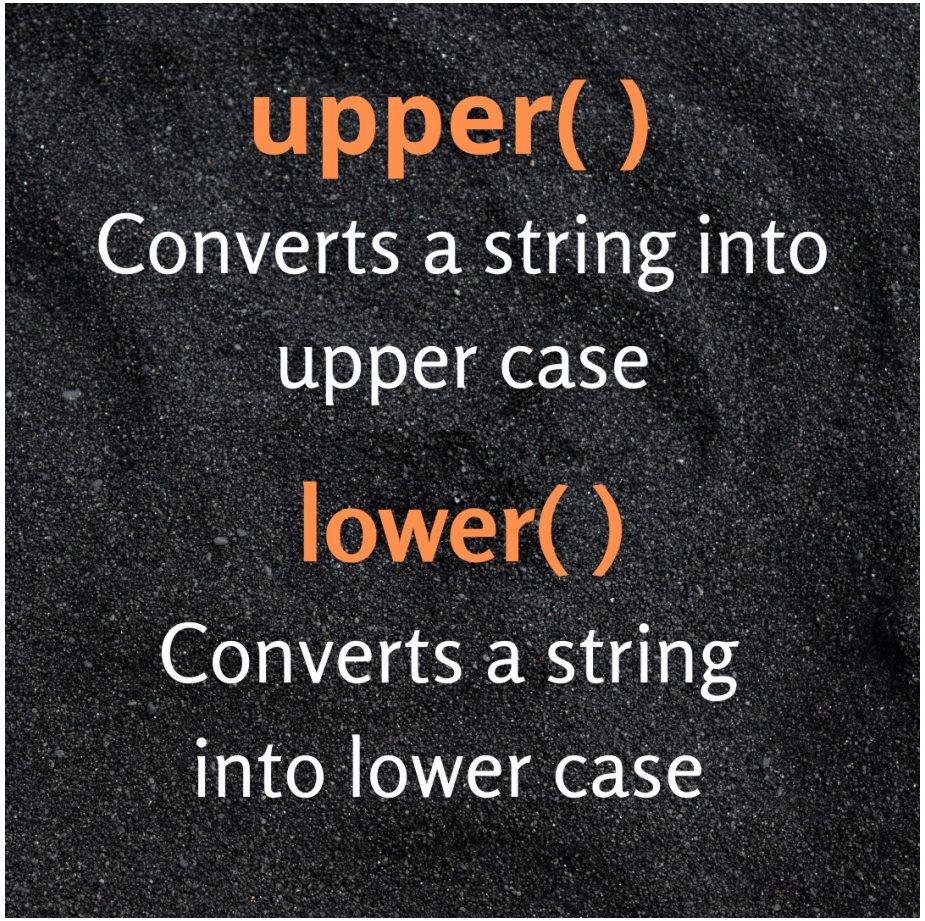
Variables and Simple Data Types

- Variables – temporary storage in memory (label to an address)
  - Naming rules and conventions
    - Variable names can contain only letters, numbers, and underscores
    - Names can start with an underscore, but not with a number.
    - Spaces are not allowed in variable names, but underscores can be used to separate words in variable names.
      - Variables should be lower case with _ between words
    - Avoid using Python keywords and function names as variable names.
    - Variable names should be short but descriptive.
      - Can use x,y,z but best to be descriptive ie hours, wages, first_name
    - Be consistent in naming and USAGE, ie Hours is not the same as hours.
    - Variable assignment uses = (a single equal sign)
    - Comparison of variables uses == (two equal signs)
  - Python variables are dynamically typed
    - Can be declared and assigned in one step
    - A variables type can be changed during execution (but not a good idea)
- Simple Data Types
  - Strings
  - Numbers
    - Integer
    - Float
  - Boolean

Strings

- A string is a series of characters.
- String items can be referenced with an index (indexes start at 0)
  - name = “Dale” name[0] returns “D” name[3] returns “e”
- Strings are immutable (can’t change the items in a string)
  - name[0] = “B” throws a traceback (python error)
  - Must make a copy to change the string
  - Create a new variable to work with string variable
    - name_upper = name.upper() because name remains unchanged

String Methods



String Methods

String Methods

Method	Description
`isalnum()`	Returns true if the string contains only alphabetic letters or digits and is at least one character in length. Returns false otherwise.
`isalpha()`	Returns true if the string contains only alphabetic letters and is at least one character in length. Returns false otherwise.
`isdigit()`	Returns true if the string contains only numeric digits and is at least one character in length. Returns false otherwise.
`islower()`	Returns true if all of the alphabetic letters in the string are lowercase, and the string contains at least one alphabetic letter. Returns false otherwise.
`isspace()`	Returns true if the string contains only whitespace characters and is at least one character in length. Returns false otherwise. (Whitespace characters are spaces, newlines (`\n`), and tabs (`\t`).
`isupper()`	Returns true if all of the alphabetic letters in the string are uppercase, and the string contains at least one alphabetic letter. Returns false otherwise.

String Methods

- Some methods return a copy of the string, to which modifications have been made
  - This is to simulate strings as mutable objects but immutable
  - To keep the changes, must use a new variable or overlay the old variable
- String comparisons are case-sensitive
  - Uppercase characters are distinguished from lowercase characters
  - `lower` and `upper` methods should be used for making case-insensitive string comparisons

String Methods

Method	Description
`lower()`	Returns a copy of the string with all alphabetic letters converted to lowercase. Any character that is already lowercase, or is not an alphabetic letter, is unchanged.
`lstrip()`	Returns a copy of the string with all leading whitespace characters removed. Leading whitespace characters are spaces, newlines (`\n`), and tabs (`\t`) that appear at the beginning of the string.
`lstrip(char)`	The *`char`* argument is a string containing a character. Returns a copy of the string with all instances of *`char`* that appear at the beginning of the string removed.
`rstrip()`	Returns a copy of the string with all trailing whitespace characters removed. Trailing whitespace characters are spaces, newlines (`\n`), and tabs (`\t`) that appear at the end of the string.
`rstrip(char)`	The *`char`* argument is a string containing a character. The method returns a copy of the string with all instances of *`char`* that appear at the end of the string removed.
`strip()`	Returns a copy of the string with all leading and trailing whitespace characters removed.
`strip(char)`	Returns a copy of the string with all instances of *`char`* that appear at the beginning and the end of the string removed.
`upper()`	Returns a copy of the string with all alphabetic letters converted to uppercase. Any character that is already uppercase, or is not an alphabetic letter, is unchanged.

String Methods

Method	Description
`endswith(substring)`	The *substring* argument is a string. The method returns true if the string ends with *substring*.
`find(substring)`	The *substring* argument is a string. The method returns the lowest index in the string where *substring* is found. If *substring* is not found, the method returns –1.
`replace(old, new)`	The *old* and *new* arguments are both strings. The method returns a copy of the string with all instances of *old* replaced by *new*.
`startswith(substring)`	The *substring* argument is a string. The method returns true if the string starts with *substring*.

Joining strings

- f-strings (old format function)
  - full_name = f"{first_name} {last_name}"
  - The f is for format, because Python formats the string by replacing the name of any variable in braces with its value.
- Concatenation
  - full_name = first_name + “ “ + last_name
  - This looks like text addition but only joins the strings (unlike the f string, you must add spacing)
- Multiplication
  - name = “Dale” name * 3 results in DaleDaleDale

String Manipulation

- To access an individual character in a string:
  - Use a `for` loop
    - Format: `for` *`character`* `in` *`string`*`:`
    - Useful when need to iterate over the whole string, such as to count the occurrences of a specific character
  - Use indexing
    - Each character has an index specifying its position in the string, starting at 0
    - Format: *`character = my_string`*`[`*`i`*`]`

String Slicing

- A Slice is a span of items taken from a sequence, known as *substring*
  - Slicing format: `string[start : end]`
    - Expression will return a string containing a copy of the characters from `start` up to, but not including, `end`
    - If `start` not specified, `0` is used for start index
    - If `end` not specified, `len(string)` is used for end index
  - Slicing expressions can include a step value and negative indexes relative to end of string

Numbers

- Integers (whole numbers no decimals) 1, 10
- Float (numbers with decimal values) 1.0, 10.50
- Python supports order of operations PEMDAS
- (), **, *, / (//), +, -
- Division operation depends on number type
  - Divide float by float use / returns float
  - Divide float by int use / returns float
  - Divide int by int using / returns float
  - Divide int by int using // returns int
- Multiple Assignment
  - x, y, z = 1,2 3 results in x == 1, y == 2, z == 3

Constants and Comments

- Constants (do not change during execution of a program)
  - A constant is like a variable whose value stays the same throughout the life of a program.
  - Use all capital letters to indicate a variable should be treated as a constant and never be changed ie TAX_RATE = 6.25
- Comments
  - A comment allows you to write notes in English within your programs.
  - In Python, the hash mark (#) indicates a comment.
  - Anything following a hash mark in your code is ignored by the Python interpreter.
  - Comments are for humans, not computers

Input function

- The input() functio… aits for the user to enter some text.
  - name = input("Ente…
- The input() functio… mpt, or instructions.
- User data goes into …
- All data passed via t… ring
- Any numeric data fr… e converted for use in math functions ie… age") then use int(my_age) to conv…

What will the following program print out: >>> x = 15; >>> x = x + 5; >>> print(x)?
Python scripts (files) have names that end with what type of file extension?
Which of these words are reserved words in Python ?
_____ errors occur when Python doesn't recognize a section of your program as valid Python code.
Programs that are compiled are easier to debug.
A sequence of instructions in a programming language is called an Algorithm.
When the computer does not understand the statement that you entered, it is a _____ error.
Python is a strongly typed language in which you must delcare your variables before use.
Python and C# are both compliers
A program is a sequence of stored instructions
When you are typing code directly at the chevron >>> you are coding interactively.
Sequential code is when questions are asked of the program to determine flow.
Conditional code is when the code block executes from top to bottom.
Repeative code is when the code block executes over and over until it is finished.
Which of one of the following functions is concerned with getting information from the user?
An _____ reads the source code of the program as written by the programmer, parses the source code, and interprets the instructions on the fly.
A compiler will process the source code and generate a(n) _____ file.
The python interpeter was written in the _____ programming language.
In the following code, x = 42. What is "x"?
Which of the following variables is the "most mnemonic"?
Which of the following is not a Python reserved word?
Assume the variable x has been initialized to an integer value (e.g., x = 3). What does the following statement do? x = x + 2
Which of the following elements of a mathematical expression in Python is evaluated first?
What will be the value of x after the following statement executes: x = 1 + 2 * 3 – 8 / 4
What will be the value of x when the following statement is executed: x = int(98.6)
Variable names can be arbitrarily long. They can contain both letters and numbers and can begin with a number.
Variable names can contain an underscore _ and can start with one.
In Python, a variable must be declared before it is assigned a value
In Python, a variable may be assigned a value of one type, and then later assigned a value of a different type:
What is the output of the following code? print(100/50)
What is the output of the following code? print(7//3)
What is the output of the following code? print(7%3)
What is the output of the following code? print(3**2)
What is the value of the expression 1 + 2 ** 2 * 3 ?
What is the output of the following code? k=10 k -= 2 print(k)
The + operator works with strings, but it is not addition in the mathematical sense. This is called _____.
Python provides a built-in function called _____ that gets input from the keyboard.
Every _____ is connected to a _____, which is the information associated with it.
Variable names can contain only letters, numbers and _____.
Spaces are allowed in variable names, but underscores can not be used to separate words in variable names.
Avoid using Python keywords and function names as variable names.
Variable names should be short but descriptive.
The variables Message and message are the same in Python.
A _____ is a series of characters.
Strings are mutable (they can be changed or altered).
A _____ is an action that Python can perform on a piece of data.
Every method is followed by a set of parentheses, because methods often need additional information to do their work.
The_____ function test to see what type an object is and returns true or false
The _____ will remove whitespace from both ends of a string and returns a copy of the new string.
A character in a string has an index specifying its position, starting at 1

Modulo Operator

- Modulo operator (%) divides one number by another number and returns the remainder only
- Useful tool for working with numerical information
- Great for finding even or odd numbers