20 May

Python Basic - 1

* 1. What are keywords in python? Using the keyword library, print all the python keywords.

**Solution 1.**

Keywords in Python are reserved words that have predefined meanings and cannot be used as identifiers (variable names, function names, etc.) because they are part of the language syntax. These keywords are used to define the structure and logic of the code.

To print all the Python keywords, you can use the keyword module in Python. Here's an example

import keyword

all\_keywords = keyword.kwlist

print(all\_keywords)

The keyword.kwlist attribute returns a list of all the keywords in Python. By printing this list, you can see all the keywords supported by the Python language.

* 1. What are the rules to create variables in python?

**Solution 2.**

In Python, there are certain rules and conventions to follow when creating variables. Here are the key rules for creating variables in Python:

1.Variable Names:

* Variable names can consist of letters (a-z, A-Z), digits (0-9), and underscores (\_).
* The first character of a variable name cannot be a digit. It should be a letter or an underscore.
* Variable names are case-sensitive, meaning myVar and myvar are considered different variables.
* Variable names should be descriptive and meaningful, following the naming conventions for better code readability.

2.Reserved Keywords:

* Variable names cannot be the same as reserved keywords in Python. These keywords have predefined meanings and cannot be used as variable names.
* Examples of reserved keywords include if, else, for, while, def, class, import, etc.

3.Naming Conventions:

* It is recommended to follow the Python naming conventions (PEP 8) for variables:
* Use lowercase letters for variable names.
* Separate words in variable names using underscores (snake\_case).
* Choose meaningful and descriptive names that reflect the purpose of the variable.

4.Assignment:

* Variables are created and assigned values using the assignment operator =.
* Example: my\_variable = 10
  1. What are the standards and conventions followed for the nomenclature of variables in python to improve code readability and maintainability?

**Solution 3.**

In Python, there are widely accepted standards and conventions for variable naming to improve code readability and maintainability. The most commonly followed convention is outlined in PEP 8, which is the official style guide for Python code. Here are the key naming conventions:

1.Use descriptive and meaningful names:

Variable names should clearly convey the purpose or meaning of the variable.

Use names that make the code more self-explanatory and easier to understand.

2.Use lowercase letters:

Variable names should be in lowercase letters.

Avoid using uppercase letters for variable names, except for constants (which are typically written in uppercase).

3.Separate words with underscores (snake\_case):

When a variable name consists of multiple words, separate them with underscores.

This convention is called snake\_case.

Example: my\_variable\_name

4.Avoid using single-letter names:

Unless the variable represents a well-known convention like i for an index variable in a loop, avoid using single-letter variable names.

Use more descriptive names that provide context and improve code readability.

5.Avoid using reserved keywords:

Do not use reserved keywords as variable names.

Reserved keywords are predefined words in Python with special meanings and purposes.

6.Follow conventions for specific variable types:

Some conventions are followed for specific types of variables, such as using uppercase for constants and using prefixes or suffixes to indicate variable types (e.g., list\_var, str\_value).

* 1. What will happen if a keyword is used as a variable name?

**Solution 4.**

If a keyword is used as a variable name in Python, it will result in a syntax error. Keywords are reserved words in the Python language and have predefined meanings and functionalities. They cannot be used as identifiers (variable names, function names, etc.) because they are part of the language syntax.

When you attempt to use a keyword as a variable name, Python will raise a syntax error to indicate that the usage is not valid. The specific error message will indicate that the keyword is not allowed as an identifier

* 1. For what purpose def keyword is used?

**Solution 5.**

The def keyword in Python is used to define a function. When you define a function, you use the def keyword followed by the function name and a set of parentheses containing any parameters the function may accept. The function definition is then followed by a colon (:), and the body of the function is indented.

* 1. What is the operation of this special character ‘\’?

**Solution 6**

The special character '' in Python is known as the backslash or escape character. It is used to perform certain special operations and to represent certain characters in string literals. Here are some common uses of the backslash:

1.Escape sequences: The backslash is used to create escape sequences that represent special characters within a string. For example:

'\n' represents a newline character.

'\t' represents a tab character.

'"' represents a double quote character.

'\' represents a literal backslash character.

2.Raw strings: The backslash can be used to create raw strings, which treat backslashes as literal characters rather than escape characters. Raw strings are often used when working with regular expressions or file paths. To create a raw string, prefix the string literal with the letter 'r'. For example:

path = r'C:\Users\Username\Documents'

3.Line continuation: The backslash can be used to continue a long line of code onto the next line. This is useful when you have a long statement or expression that doesn't fit comfortably on a single line. For example:

total = number1 + \ number2 + \ number3

* 1. Give an example of the following conditions:

1. Homogeneous list
2. Heterogeneous set
3. Homogeneous tuple

**Solution 7.**

Homogeneous list: A homogeneous list is a list that contains elements of the same data type. Here's an example of a homogeneous list containing integer values:

numbers = [1, 2, 3, 4, 5]

Heterogeneous set: A heterogeneous set refers to a set that contains elements of different data types. Here's an example of a heterogeneous set:

my\_set = {1, 'hello', 3.14, True}

Homogeneous tuple: A homogeneous tuple is a tuple that contains elements of the same data type. Here's an example of a homogeneous tuple containing string values:

fruits = ('apple', 'banana', 'orange', 'mango')

* 1. Explain the mutable and immutable data types with proper explanation & examples.

**Solution 8.**

In Python, data types are categorized as either mutable or immutable based on whether their values can be changed after they are created.

**Mutable Data Types:**

Mutable data types are those whose values can be modified after they are created. Any changes made to a mutable object will directly affect the object itself. Examples of mutable data types in Python include lists, dictionaries, and sets.

Example of a mutable data type (list):

numbers = [1, 2, 3]

numbers.append(4) # Modifying the list by adding a new element

print(numbers) # Output: [1, 2, 3, 4]

**Immutable Data Types:**

Immutable data types are those whose values cannot be changed once they are created. If any modification is made to an immutable object, a new object is created with the modified value. Examples of immutable data types in Python include strings, numbers (integers, floats), and tuples.

Example of an immutable data type (string):

message = "Hello"

new\_message = message + " World" # Creating a new string by concatenation

print(new\_message) # Output: Hello World

print(message) # Output: Hello

* 1. Write a code to create the given structure using only for loop.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

**Solution 9.**

rows = 5

for i in range(rows):

for j in range(rows - i - 1):

print(" ", end="")

for j in range(2\*i + 1):

print("\*", end="")

print()

* 1. Write a code to create the given structure using while loop.

|||||||||

|||||||

|||||

|||

|

**Solution 10**

rows=5

for i in range(rows-1,-1,-1):

for j in range(rows-i-1):

print(" ",end="")

for j in range(2\*i+1):

print("|",end="")

print()