20 May

Python Basic - 1

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

**Answer –**

keywords are reserved words that have special meanings and purposes within the language. These keywords cannot be used as variable names or identifiers because they are already predefined for specific functionalities. They are an essential part of Python's syntax and structure.

**Input -**

import keyword

keywords = keyword.kwlist

for i in keywords:

print(i)

**Output –**

False

None

True

and

as

assert

async

await

break

class

continue

def

del

elif

else

except

finally

for

from

global

if

import

in

is

lambda

nonlocal

not

or

pass

raise

return

try

while

with

yield

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

**Answer –**

To create variables in Python these are the following rules:

1. Variable names can contain letters (a-z, A-Z), digits (0-9), and underscores (\_).
2. The first character of a variable name cannot be a digit.
3. Variable names are case-sensitive.
4. Avoid using reserved keywords as variable names (e.g., "if", "for", "while").
5. Variable names should be descriptive and meaningful.

variable should be a combination of letters, digits, and underscores, starting with a letter and not being a reserved keyword.

* 1. **What are the standards and conventions followed for the nomenclature of variables in python to improve code readability and maintainability?**

**Answer –**

To improve code readability and maintainability these standards and conventions for variable naming shoul be followed -

1. Use descriptive names that reflect the purpose or content of the variable.
2. Use lowercase letters and underscores for multi-word variable names.
3. Avoid using single letters or generic names that don't provide meaningful information.
4. Be consistent in naming style and casing throughout your code.
   1. **What will happen if a keyword is used as a variable name?**

**Answer –**

If a keyword is used as a variable name in Python, it will result in a syntax error.

* 1. **For what purpose def keyword is used?**

**Answer –**

The **def** keyword is used to define a function.

When you use the def keyword followed by a function name, parentheses, and a colon, you are creating a function definition. This allows you to define a block of reusable code that can be called and executed later in the program.

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

**Answer –**

The backslash **(‘\’)** serves as an escape character, allowing you to include special characters or create special character sequences within strings.

* 1. **Give an example of the following conditions:**

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

**Answer –**

1. **Homogeneous list:** An example of a homogeneous list would be a list of integers, where all the elements in the list are of the same data type. For instance, [1, 2, 3, 4, 5] is a homogeneous list of integers because all the elements are integers.
2. **Heterogeneous set:** A heterogeneous set refers to a set that contains elements of different data types. For example, {1, 'apple', 3.14, True} is a heterogeneous set because it includes an integer, a string, a floating-point number, and a boolean value.
3. **Homogeneous tuple:** A homogeneous tuple is a tuple in which all the elements have the same data type. An example would be ('cat', 'dog', 'rabbit'), where all the elements are strings.
   1. **Explain the mutable and immutable data types with proper explanation & examples.**

**Answer –**

Mutable data types can be modified after they are created, while immutable data types cannot be changed once they are created.

**Examples of mutable data types:**

**Lists**: You can modify the elements of a list by adding, removing, or modifying items. For example:

my\_list = [1, 2, 3]

my\_list[0] = 4

print(my\_list)

# Output: [4, 2, 3]

**Examples of immutable data types:**

**Strings**: Once a string is created, you cannot change its individual characters. You can only create a new string. For example:

my\_string = "Hello"

new\_string = my\_string + " World"

print(new\_string)

# Output: "Hello World"

**Tuples:** Tuples are similar to lists, but they are immutable. Once a tuple is created, you cannot modify its elements. For example:

my\_tuple = (1, 2, 3)

my\_tuple[0] = 4

#This would raise an **error**

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

\*

\*\*\*

\*\*\*\*\*

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

**Answer –**

x = 4

for i in range(x + 1):

print("\*" \* (2\*i + 1))

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

|||||||||

|||||||

|||||

|||

|

**Answer –**

x = 5

while x >= 1:

print("|" \* x)

x -= 1