**Name: Abhas**

**Email: abhas2495@gmail.com**

20 May **Python Basic - 1**

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

**Answers:**

Keywords in Python are reserved words that have predefined meanings and serve specific purposes in the language. These keywords cannot be used as variable names or identifiers because they are part of the Python syntax. They are used to define control structures, loop statements, data types, and other fundamental aspects of the language.

**Code:**

**import keyword**

**all\_keywords = keyword.kwlist**

**print(all\_keywords)**

Q.2. What are the rules to create variables in python?

**Answers:**

**Valid Characters**: Variable names can consist of letters (a-z, A-Z), digits (0-9), and underscores (\_). However, they cannot start with a digit.

**Case-Sensitivity:** Python is case-sensitive, so variables like myVariable and myvariable are considered different.

**Reserved Keywords:** You cannot use reserved keywords (e.g., if, while, for, def, etc.) as variable names, as they have special meanings in the Python language.

**Meaningful Names:** It is recommended to use meaningful and descriptive names for variables to enhance code readability.

**No Spaces or Special Characters:** Variable names cannot contain spaces or special characters like !, @, $, %, etc. Instead, use underscores to represent multiple words, like my\_variable or total\_sum.

**Length Limit:** Variable names can be of any length, but it's advisable to keep them concise and meaningful.

Q.3. What are the standards and conventions followed for the nomenclature of variables in python to improve code readability and maintainability?

**Answers:**

Adhering to certain standards and conventions for variable naming can significantly improve code readability and maintainability. Here are some commonly followed standards and conventions:

**Snake Case:** Use lowercase letters with underscores (\_) to separate words in variable names. This convention is known as snake case. For example: my\_variable, total\_sum, user\_name.

**Descriptive Names:** Choose meaningful and descriptive names for variables that accurately represent their purpose or contents. Avoid using single-character names or vague abbreviations that can make the code harder to understand.

**Avoid Reserved Keywords:** Do not use reserved keywords as variable names. Reserved keywords have special meanings in Python and cannot be used as identifiers. For example, avoid using if, while, for, def, class, etc., as variable names.

**Use Lowercase for Most Variables:** Typically, variable names in Python are written in all lowercase letters. This convention helps distinguish variables from classes and other objects.

**Constants:** If a variable represents a constant value that should not be modified, use uppercase letters and underscores to separate words. For example: MAX\_COUNT, PI, DEFAULT\_VALUE.

**Use Plural for Collections**: When a variable represents a collection or a sequence of items, it is a good practice to use a plural name. For example: students, fruits, books.

**Be Consistent:** Maintain consistency in your variable naming conventions throughout your codebase. Use similar styles and naming conventions to make the code more cohesive and easier to understand.

**Follow PEP 8:** PEP 8 is the official Python style guide, which provides guidelines for writing readable Python code. It covers various aspects, including variable naming conventions. Adhering to PEP 8 can greatly improve code consistency and maintainability.

By following these standards and conventions, you can make your Python code more readable, self-explanatory, and consistent, which in turn improves code maintainability and collaboration with other developers.

Top of Form

Q.4. What will happen if a keyword is used as a variable name?

**Answers:**

If a keyword is used as a variable name in Python, it will result in a **SyntaxError.** Keywords in Python have predefined meanings and are reserved for specific purposes in the language's syntax. Using a keyword as a variable name violates this rule, as it conflicts with the intended usage of the keyword.

**Example:**

for = 10

**Output:**

SyntaxError: invalid syntax

Q.5. For what purpose def keyword is used?

**Answers:**

The def keyword in Python is used to define a function. It is an essential keyword for creating reusable blocks of code that can be called and executed at different points in a program.

Here's the basic syntax for defining a function using the def keyword:

**def function\_name(parameters):**

**# Function body**

**# Code to be executed**

**# Optionally, return a value**

Q.6. What is the operation of this special character ‘\’?

**I. Escape Sequences:** The backslash \ is used as an escape character to represent special sequences or characters that have a specific meaning in Python. For example:

1. '\n' represents a newline character.
2. '\t' represents a tab character.
3. '\'' represents a single quote character.
4. '\"' represents a double quote character.
5. '\\' represents a literal backslash character.

II. Line Continuation: The backslash \ can be used to continue a long line of code onto the next line. This is useful when a line of code becomes too long and hampers readability

**Code:**

**my\_long\_variable = 10 + \**

**20 + \**

**30**

III. Unicode Characters: The backslash \ is used to represent Unicode characters by their hexadecimal value. For example, '\uXXXX' represents a Unicode character with the hexadecimal code XXXX.

Q.7. Give an example of the following conditions:

1. Homogeneous list
2. Homogeneous List: A homogeneous list is a list where all elements have the same data type. Here's an example of a homogeneous list containing integers:

**Example:**

**my\_list = [1, 2, 3, 4, 5]**

In the above example, all the elements in the list my\_list are integers, making it a homogeneous list.

1. Heterogeneous set
2. A heterogeneous set is a set where the elements can have different data types. Here's an example of a heterogeneous set containing elements of different types

**Example:**

**my\_set = {1, 'apple', 3.14, True}**

1. Homogeneous tuple
2. A homogeneous tuple is a tuple where all elements have the same data type. Here's an example of a homogeneous tuple containing strings:

**Example:**

my\_tuple = ('apple', 'banana', 'cherry', 'date')

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

**Answers:**

* 1. **Immutable Data Types:** Immutable data types are those whose values cannot be modified after they are created. If you want to modify an immutable object, you need to create a new object with the desired changes. Examples of immutable data types in Python include:

1. int (integer)
2. float (floating-point number)
3. str (string)
4. tuple

Here's an example to illustrate the immutability of an integer:

**x = 5**

**print(x) # Output: 5**

**x = x + 1**

**print(x) # Output: 6**

**Mutable Data Types:** Mutable data types are those that allow in-place modifications to their values. This means you can modify their elements, add or remove elements, or change their state directly. Examples of mutable data types in Python include:

1. list

j. dict (dictionary)

k. set

Here's an example to demonstrate the mutability of a list:

my\_list = [1, 2, 3]

print(my\_list) # Output: [1, 2, 3]

my\_list.append(4)

print(my\_list) # Output: [1, 2, 3, 4]

my\_list[1] = 5

print(my\_list) # Output: [1, 5, 3, 4]

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

\*

\*\*\*

\*\*\*\*\*

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

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

**Answer:**

**Code:**

**rows = 5**

**for i in range(rows):**

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

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

**print()**

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

|||||||||

|||||||

|||||

|||

|

**Answers:**

**Code:**

**rows = 5**

**while rows >= 1:**

**print("|" \* rows)**

**rows -= 1**