20 May **Python Basic - 1**

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

**Answer:**

Keywords in Python are reserved words that have predefined meanings and cannot be used as identifiers (such as variable names, function names, etc.) in your code. keywords have specific meanings in the Python language and are used for various programming tasks.

The `keyword` library in Python to get a list of all the keywords.

import keyword

all\_keywords = keyword.kwlist

print(all\_keywords)

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

**Answer:** The name of a variable can consist of letters (both lowercase and uppercase), digits, and underscores.

It must start with a letter (a-z, A-Z) or an underscore (\_). Variables cannot start with a digit or any special characters.

Python is case-sensitive, meaning that variables with different capitalizations are considered different. For example, `myVariable’ and `MYVARIABLE` are all different variables.

While there is no strict requirement, it's a good practice to follow the naming conventions. Use lowercase letters for variable names and separate words with underscores For example: `my\_variable`, `age’.

You cannot use Python keywords (reserved words) as variable names, such as `if`, `else`, etc.

Descriptive Names: It's a good practice to use descriptive variable names that convey the purpose of the variable.

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

**Answer:** 1. Use lowercase letters for variable names.

2. Separate words in variable names using underscores (`snake\_case`) to enhance readability.

3. Avoid using single-character variable names (unless they represent well-known conventions, like `i` for loop counters).

4. For constants, use uppercase letters and underscores (`ALL\_CAPS`) to indicate that they should not be modified.

5. Use descriptive and meaningful names that convey the purpose of the variable.

6. Choose consistency in naming conventions throughout codebase.

7. For class names, use `CamelCase` .

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

**Answer:** If you use a keyword as a variable name in Python, you will encounter a syntax error.

For example,

if = 10 # This will result in a syntax error

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

**Answer:** The `def` keyword in Python is used to define a function. Functions are blocks of reusable code that perform a specific task.

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

def function\_name(parameters):

# Function body

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

**Answer:** The special character `'\'` is called the backslash in Python. It is used to escape characters that have a special meaning in strings or to create special characters. For example:

- `\'`: Represents a single quote within a string.

- `\"`: Represents a double quote within a string.

- `\\`: Represents a literal backslash within a string.

- `\n`: Represents a newline character.

- `\t`: Represents a tab character.

- `\r`: Represents a carriage return.

- `\b`: Represents a backspace.

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

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

**Answer:** (i) Homogeneous List: A homogeneous list contains elements of the same data type. In this example, all elements are integers.

homogeneous\_list = [1, 2, 3, 4, 5]

(ii) Heterogeneous Set: A heterogeneous set contains elements of different data types. In this example, the set contains a mix of integers, strings, and a boolean.

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

(iii) Homogeneous Tuple: A homogeneous tuple contains elements of the same data type. In this example, all elements are strings.

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

.

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

**Answer:** **Mutable Data Types:** Mutable data types are those whose values can be modified after they are created. This means you can change the contents of the object without creating a new instance of it. When you modify a mutable object, you are actually modifying the same object in memory.Examples of mutable data types:

1. Lists 2. Dictionaries 3. Sets

Example: mutable\_list = [1, 2, 3]

mutable\_list[1] = 4 # Modify an element

mutable\_list.append(5) # Add an element

mutable\_list.remove(3) # Remove an element

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

**Immutable Data Types:** Immutable data types are those whose values cannot be changed after they are created. If you want to make any changes to an immutable object, you need to create a new instance with the desired modifications. Examples of immutable data types:

1. Integers 2. Floats 3. Strings 4. Tuples

example: immutable\_string = "Hello"

# immutable\_string += " World" # This will cause an error

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

**\***

**\*\*\***

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

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

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

**Answer:**

rows = 5

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

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

print(" ", end="")

for k in range(1, 2 \* i):

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

print()

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

**|||||||||**

**|||||||**

**|||||**

**|||**

**|**

**Answer:**rows = 5

i = 1

while i <= rows:

j = 1

while j < i:

print(" ", end="")

j += 1

k = 1

while k <= 2 \* (rows - i) + 1:

print("|", end="")

k += 1

print()

i += 1