20 May

Python Basic - 1

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

**ANSWER-**

Python keywords are special reserved words that have specific meaning and purposes and can’t be used for anything but those specific purposes.

These keywords are always available you don’t need to import them into your code.

Python keywords are different from python’s built-in functions and types.

**Example**-

import keyword

print(keyword.kwlist)

**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:**

Rules for Python variables:

* A variable name must start with a letter or the underscore character.
* A variable name cannot start with a number.
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
* Variable names are case-sensitive (age, Age and AGE are three different variables)
  1. What are the standards and conventions followed for the nomenclature of variables in python to improve code readability and maintainability?

**ANSWER:**

Use grammatically correct variable and class name should start with an uppercase and must follow camelCase convention, If more than two words are to be used.

In the same way, a function name should be joined with an underscore and it must be lowercase.

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

**ANSWER:**

If keywords were allowed as variable names, it would be very hard to tell (for developers and compilers) whether something was a variable or a keyword.

For example-

if (x ==10)

is it an if- statement or a calling function called if ??

Neither the developer nor the compiler would be able to tell.

* 1. For what purpose def keyword is used?

**ANSWER**:

Python def keyword is used to define a function, it is placed before a function name that is provided by the user to create a user defined function.

**Example**-

def python\_def\_keyword():

    print("Hello")

python\_def\_keyword()

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

**ANSWER:**

In python strings, the backslash “\” is a special character also called the “escape“ character.It is used in representing certain whitespace characters:

“\t” is tab

“\n” is newline

“\r” is a carriage return

**For example-**

data = "We have enrolled in \"iNeuron's\" Full Stack Data Science class."

print(data)

|  |  |
| --- | --- |
| Code | Result |
| \' | Single Quote |
| \\ | Backslash |
| \n | New Line |
| \r | Carriage Return |
| \t | Tab |
| \b | Backspace |
| \f | Form Feed |
| \ooo | Octal value |
| \xhh | Hex value |

* 1. Give an example of the following conditions:

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

**ANSWER-**

**Homogeneous list-**

Homogeneous means that it only contains a single type of data meaning only one type of data can be stored in a array.

**Heterogeneous set-**

A heterogeneous set are made up of any type of numeric, string, Boolean, objects, etc.

This means that a list/set can have any type of data and we can iterate over this list using any type of loop.

**Homogeneous tuple-**

A tuple can used to group values of any type and tuple have heterogeneous aggregation mechanism.

A tuple cannot be homogeneous in nature because we can not manipulate the items from it.

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

**ANSWER-**

Mutable objects or data type allows you to change their value or data in place without affecting the object’s identity/type.

Immutable objects or data type don’t allows you to change their value instead you will just have an option of creating new objects of the dame type with different values.

In short the mutable objects can changed after it is created and an immutable object can not.

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

\*

\*\*\*

\*\*\*\*\*

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

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

**ANSWER-**

n = 5

for i in range(n):

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

print(" ", end="")

for k in range(i+1):

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

print()

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

|||||||||

|||||||

|||||

|||

|

**ANSWER-**

for i in range(5):

for J in range(i):

print(" ", end="")

for K in range(i, 5):

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

print()