Python Basic Assignment – 5

1. **What does an empty dictionary's code look like?**

In Python, an empty dictionary's code looks like this:

empty\_dict = {}

print(empty\_dict)

An empty dictionary is represented by curly braces {} with nothing inside, indicating that it contains no key-value pairs. It is a fundamental data structure in Python and can be used to store and organize data in a key-value format. You can add elements to the dictionary by assigning values to specific keys, like this:

1. **What is the value of a dictionary value with the key 'foo' and the value 42?**

In the dictionary, 'foo' is the key, and 42 is the associated value. Dictionaries in Python are used to store data in a key-value format, where each key is unique, and it maps to a corresponding value. You can access the value associated with the key 'foo' using square brackets:

my\_dict = {'foo': 42}

value\_of\_foo = my\_dict['foo']

print(value\_of\_foo) # Output: 42

This will retrieve the value 42 from the dictionary my\_dict using the key 'foo'.

1. **What is the most significant distinction between a dictionary and a list?**

The following are the most significant distinction between a dictionary and a list

* **Data Structure:** A dictionary is an unordered collection of key-value pairs. Each key in the dictionary must be unique, and it maps to a corresponding value. A list is an ordered collection of elements, and each element is identified by an index. The elements in a list can be of any data type, and they are stored in a specific order.
* **Accessing Elements:** Elements in a dictionary are accessed using their keys. You use the keys as an index to retrieve the associated values. This allows for fast data retrieval based on known keys. Elements in a list are accessed using numerical indices. The first element has an index of 0, the second element has an index of 1, and so on. This means that you access elements in a list based on their position in the list.
* **Mutability:** Dictionaries are mutable, which means you can add, modify, or delete key-value pairs after the dictionary is created. Lists are also mutable. You can change elements, add new elements, or remove elements from the list.
* **Syntax:** Dictionaries are represented using curly braces {} with key-value pairs inside, separated by colons. Lists are represented using square brackets [], and elements are separated by commas

1. **What happens if you try to access spam['foo'] if spam is {'bar': 100}?**

If you try to access spam['foo'] and spam is {'bar': 100}, it will raise a KeyError. In Python, accessing a dictionary with a key that does not exist in the dictionary will result in a KeyError exception. In this case, the key 'foo' does not exist in the spam dictionary, so attempting to access spam['foo'] will raise the error.

1. **If a dictionary is stored in spam, what is the difference between the expression’s 'cat' in spam and 'cat' in spam.keys()?**

In Python, if a dictionary is stored in the variable spam, there is a subtle difference between the expression’s 'cat' in spam and 'cat' in spam.keys():

* **'cat' in spam:** This expression checks whether the key 'cat' exists in the dictionary spam. It returns a Boolean value (True or False) indicating whether the key 'cat' is present as a key in the dictionary.
* **'cat' in spam.keys():** This expression explicitly checks whether the key 'cat' exists in the set of keys of the dictionary spam. The keys() method returns a view object containing all the keys of the dictionary. So, 'cat' in spam.keys() checks if the key 'cat' is present in this view object.

Both expressions will give the same result since 'cat' is being used as a key, and keys in a dictionary are always unique.

1. **If a dictionary is stored in spam, what is the difference between the expression’s 'cat' in spam and 'cat' in spam.values()?**

In Python, if a dictionary is stored in the variable spam, there is a subtle difference between the expression’s 'cat' in spam and 'cat' in spam.keys():

* **'cat' in spam:** This expression checks whether the key 'cat' exists in the dictionary spam. It returns a Boolean value (True or False) indicating whether the key 'cat' is present as a key in the dictionary.
* **'cat' in spam.keys():** This expression explicitly checks whether the key 'cat' exists in the set of keys of the dictionary spam. The keys() method returns a view object containing all the keys of the dictionary. So, 'cat' in spam.keys() checks if the key 'cat' is present in this view object.

Both expressions will give the same result since 'cat' is being used as a key, and keys in a dictionary are always unique.

1. **What is a shortcut for the following code?**

**if 'color' not in spam:**

**spam['color'] = 'black'**

A shortcut for the given code can be achieved using the dict.setdefault() method. The setdefault() method allows you to set a default value for a key if the key is not already present in the dictionary. If the key exists, it returns the value associated with the key. If the key does not exist, it sets the key with the provided default value and returns the default value.

spam.setdefault('color', 'black')

This line of code checks if the key 'color' is in the spam dictionary. If it is not present, it sets the key 'color' with the value 'black'. If the key is already in the dictionary, it does nothing, and the existing value remains unchanged.

1. **How do you "pretty print" dictionary values using which module and function?**

To "pretty print" dictionary values in Python, you can use the pprint module, which stands for "pretty-print." The pprint module provides a function called pprint() that formats the dictionary in a visually more readable and organized way when printed.

import pprint

my\_dict = {'name': 'John', 'age': 30, 'city': 'New York'}

pprint.pprint(my\_dict)

The above code snippet displays each key-value pair on a separate line, and the elements are aligned for better readability. The pprint() function is especially useful when dealing with large and complex dictionaries, as it makes the output much more manageable and easier to understand.