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

**Ans:** An empty dictionary in Python is represented by a pair of curly braces {} with no key-value pairs inside. Here's what it looks like:

Python code

empty\_dict = {}

This creates an empty dictionary named empty\_dict. we can later add key-value pairs to this dictionary using the dictionary's methods or by directly assigning values to keys.

**2. What is the value of a dictionary value with the key ‘foo’ and the value 42?**

**Ans:** The value of a dictionary value with the key 'foo' and the value ‘42’ is simply ‘42’. In a dictionary, values are associated with keys, and you can retrieve a value using its corresponding key.

Python Code

my\_dict={"foo":42}

value=my\_dict["foo"]

print(value)

42

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

**Ans:** The most significant distinction between a dictionary and a list in Python lies in how they store and access their elements:

1. **Storage and Access:**
   * Dictionary: A dictionary stores elements as key-value pairs, where each value is associated with a unique key. Keys are used to access the corresponding values, making dictionary lookups very efficient even for large collections of data. Keys can be of any immutable type (such as strings, numbers, or tuples), and they must be unique within a dictionary.
   * **List:** A list stores elements in an ordered sequence using integer indices. Each element is associated with an index starting from 0. Lists are indexed using these integer indices, and elements are accessed sequentially. List lookups can be slower for larger lists, as they require iterating through the list to find an element based on its index.

**Example:**

my\_dict={"Name":"Krithvik","city":"Bangalore","Age":2,}

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

print(my\_dict["Age"])

2

print(my\_list[1])

2

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

**Ans:** If we try to access spam['foo'] when spam is {'bar': 100}, we will encounter a KeyError. This error occurs because the key 'foo' does not exist in the dictionary spam.

Python Code

spam={"bar":100}

print(spam["foo"])

Traceback (most recent call last):

File "<pyshell#10>", line 1, in <module>

print(spam["foo"])

KeyError: 'foo'

value=spam["foo"]

Traceback (most recent call last):

File "<pyshell#11>", line 1, in <module>

value=spam["foo"]

KeyError: 'foo'

KeyboardInterrupt

**5. If a dictionary is stored in spam, what is the difference between the expressions ‘cat’ in spam and**

**‘cat’ in spam.keys()?**

**Ans:** If a dictionary is stored in spam, there is no practical difference between the expressions 'cat' in spam and 'cat' in spam.keys(). Both expressions are used to check whether a given key exists in the dictionary. The result of both expressions will be the same.

Python Code

spam={"cat":5,"colour":"yellow"}

"cat" in spam

True

"cat" in spam.keys()

True

**6. If a dictionary is stored in spam, what is the difference between the expressions ‘cat’ in spam and ‘cat’ in spam.values()?**

**Ans:** If a dictionary is stored in spam, there is a difference between the expressions 'cat' in spam and 'cat' in spam.values():

1. 'cat' in spam: This expression checks whether the string 'cat' is a key in the dictionary spam. If 'cat' is one of the keys in the dictionary, the expression will evaluate to True; otherwise, it will evaluate to False.
2. 'cat' in spam.values(): This expression checks whether the string 'cat' appears as a value in any of the dictionary's values. If 'cat' is present as a value in any of the values, the expression will evaluate to True; otherwise, it will evaluate to False.

Python Code

spam={"cat":5,"colour":"yellow"}

"cat" in spam

True

"cat" in spam.values()

False

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

**if ‘color’ not in spam:**

**spam[‘color’] = ‘black’**

**Ans:** A shortcut for the given code can be achieved using the setdefault() method of dictionaries. The setdefault() method checks if a key exists in the dictionary, and if it doesn't, it adds the key with the specified default value. If the key already exists, it returns the existing value without making any changes.

Python Code

spam={"cat":5,"colour":"yellow"}

if 'color' not in spam:

spam['color'] = 'black'

print(spam)

{'cat': 5, 'colour': 'yellow', 'color': 'black'}

del spam["color"]

print(spam)

{'cat': 5, 'colour': 'yellow'}

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

'black'

print(spam)

{'cat': 5, 'colour': 'yellow', 'color': 'black'}

**8. How do you “pretty print” dictionary values using which module and function?**

**Ans:** You can use the pprint module in Python to "pretty print" dictionary values. The pprint module provides a function called pprint() that formats and prints dictionary (and other data structure) values in a more human-readable and visually organized way.

Python Code

import pprint

pprint.pprint(my\_dict)

{'Age': 2, 'Name': 'Krithvik', 'city': 'Bangalore'}

pprint.pprint(spam)

{'cat': 5, 'color': 'black', 'colour': 'yellow'}

print(my\_dict)

{'Name': 'Krithvik', 'city': 'Bangalore', 'Age': 2}

print(spam)

{'cat': 5, 'colour': 'yellow', 'color': 'black'}