CHAPTER -10

Tuples and Dictionaries

2MARK QUESTIONS

Q1: Define a tuple in Python.

Answer:

A tuple is an ordered and immutable collection of elements, enclosed within parentheses, and can contain elements of different data types.

Q2: How is a tuple different from a list in Python?

Answer:

Tuples are immutable, meaning their elements cannot be modified after creation. Lists, on the other hand, are mutable and allow for modifications.

Q3: Explain the concept of tuple unpacking in Python.

Answer:

Tuple unpacking allows the values of a tuple to be assigned to multiple variables simultaneously. For example: x, y = (1, 2).

Q4: Write a Python code snippet to create an empty tuple.

Answer:

```
empty_tuple = ()
```

Q5: Discuss the significance of parentheses in distinguishing between tuples and expressions in Python.

Answer:

Parentheses are crucial for creating tuples. The presence of parentheses distinguishes tuples from expressions, ensuring proper tuple interpretation.

Dictionaries:

Q6: Define a dictionary in Python.

Answer:

A dictionary is an unordered collection of key-value pairs, where each key must be unique, and the values can be of different data types.

Q7: How can you access the value associated with a specific key in a dictionary?

Answer:

You can access the value associated with a specific key using square brackets and the key, like this: value = my_dict['key'].

Q8: Explain the purpose of the keys() and values() methods in Python dictionaries.

Answer:

The keys() method returns a view of all keys in a dictionary, while the values() method returns a view of all values. These methods are useful for iterating over keys or values.

Q9: Write a Python code snippet to add a new key-value pair to an existing dictionary.

Answer:

my_dict['new_key'] = 'new_value'

Q10: Discuss the concept of dictionary comprehension in Python.

Answer:

Dictionary comprehension is a concise way to create dictionaries in Python. It allows the definition of dictionaries in a single line using a compact syntax.

These questions cover fundamental concepts related to tuples and dictionaries in Python, providing a solid foundation for understanding their properties and usage.

4MARK QUESTIONS

Q1: Explain the concept of immutability in tuples and its significance.

Answer:

Tuples are immutable, meaning once a tuple is created, its elements cannot be modified or changed. This immutability ensures data integrity and stability. It allows tuples to serve as keys in dictionaries and elements in sets, as their values won't change during their lifetime.

Q2: Compare and contrast the syntax for creating a tuple with one element and the syntax for creating an empty tuple. Provide examples.

Answer:

To create a tuple with one element, a trailing comma is required: single_element_tuple = (1,).

To create an empty tuple, simply use empty parentheses: empty_tuple = ().

Q3: Discuss the advantages of using tuple unpacking in Python. Provide examples.

Answer:

Tuple unpacking allows multiple variables to be assigned values from a tuple simultaneously. It simplifies the assignment process and is often used in functions that return multiple values. For example:

coordinates = (3, 4)

x, y = coordinates

Q4: Write a Python program to concatenate two tuples.

Answer:

tuple1 = (1, 2, 3)

tuple2 = ('a', 'b', 'c')

concatenated_tuple = tuple1 + tuple2

Dictionaries:

Q5: Explain the purpose of the get() method in Python dictionaries and how it differs from direct key access.

Answer:

The get() method retrieves the value associated with a specified key in a dictionary. Unlike direct key access, if the key is not present, get() returns None instead of raising a KeyError. This helps in handling cases where the key may not exist.

Q6: Discuss the advantages of using dictionaries over lists for certain scenarios. Provide examples.

Answer:

Dictionaries are advantageous when data needs to be accessed by a key rather than an index. They provide faster lookups, and their key-value pairs allow for more meaningful and self-explanatory data representation. For example:

```
student_info = {'name': 'John', 'age': 18, 'grade': 'A'}
```

Q7: Write a Python program to iterate over the keys and values of a dictionary.

Answer:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}

for key, value in my_dict.items():
    print(f"Key: {key}, Value: {value}")
```

Q8: Explain the purpose of the pop() method in Python dictionaries. Provide an example.

Answer:

The pop() method removes and returns the value associated with a specified key in a dictionary. Example:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}

removed_value = my_dict.pop('b')
```

Q9: Discuss the role of dictionary comprehension in Python and provide an example.

Answer:

Dictionary comprehension is a concise way to create dictionaries. It allows the definition of dictionaries in a single line using a compact syntax. Example:

squared_dict = {x: x**2 for x in range(1, 6)}

Q10: Write a Python program to merge two dictionaries.

Answer:

dict1 = {'a': 1, 'b': 2}

dict2 = {'b': 3, 'c': 4}

merged_dict = {**dict1, **dict2}

These questions and answers cover various aspects of tuples and dictionaries in Python, providing a comprehensive understanding of their properties and usage.

7MARK QUESTIONS

Q1: Discuss the concept of immutability in tuples and provide an example of how immutability is maintained.

Answer:

Tuples are immutable, meaning their elements cannot be modified after creation. Immutability ensures data integrity. For example:

```
my_tuple = (1, 2, 3)

# Attempting to modify the tuple will result in an error

my_tuple[0] = 4
```

Q2: Explain the term "packing" and "unpacking" in the context of tuples. Provide examples.

Answer:

Packing: Combining values into a tuple.

```
packed_tuple = 1, 'hello', 3.14
```

Unpacking: Assigning values from a tuple to multiple variables.

```
x, y, z = packed_tuple
```

Q3: Write a Python program to find the sum of elements in a tuple.

Answer:

```
my_tuple = (2, 4, 6, 8, 10)

sum_of_elements = sum(my_tuple)
```

Q4: Discuss the advantages and disadvantages of using tuples as keys in dictionaries.

Answer:

Advantages: Tuples are immutable, ensuring keys won't change. They can represent composite keys. Example:

```
coordinates = {(1, 2): 'point'}
```

Disadvantages: Limited flexibility compared to using mutable keys.

Dictionaries:

Q5: Explain the concept of dictionary comprehension in Python. Provide an example.

Answer:

Dictionary comprehension is a concise way to create dictionaries. Example:

squared_dict = {x: x**2 for x in range(1, 6)}

Q6: Discuss the time complexity of common dictionary operations such as insertion, deletion, and retrieval.

Answer:

Insertion: O(1) average, O(n) worst case.

Deletion: O(1) average, O(n) worst case.

Retrieval: O(1) average, O(n) worst case.

Q7: Write a Python program to merge two dictionaries, ensuring that duplicate keys are handled properly.

Answer:

dict1 = {'a': 1, 'b': 2}

dict2 = {'b': 3, 'c': 4}

merged_dict = {**dict1, **dict2}

Q8: Explain the purpose and usage of the update() method in Python dictionaries. Provide an example.

Answer:

The update() method merges the keys and values of one dictionary into another. Example:

dict1 = {'a': 1, 'b': 2}

dict2 = {'b': 3, 'c': 4}

dict1.update(dict2)

Q9: Write a Python program to iterate over keys, values, and items in a dictionary.

Answer:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
for key in my_dict:
    print("Key:", key)
for value in my_dict.values():
    print("Value:", value)
for key, value in my_dict.items():
    print("Key:", key, "Value:", value)
```

Q10: Discuss the advantages and use cases of dictionaries over lists for certain scenarios.

Answer:

Advantages:

Fast and efficient lookups using keys.

Meaningful representation of data with key-value pairs.

No fixed size, dynamic structure.

Use Cases:

Storing settings/configuration.

Representing relationships between entities.

Efficient data retrieval based on keys.

These questions and answers cover a range of topics within the "Tuples and Dictionaries" chapter, providing a comprehensive understanding of their properties, usage, and operations in Python.

Multiple-Choice Questions (MCQs):

1Q: What is a key feature of tuples in Python?

A. Mutability

B. Unordered

C. Immutability

D. Dynamic Size

Answer: C

2Q: Which of the following is used to create an empty tuple in Python?

A. empty_tuple = {}

B. empty_tuple = ()

C. empty_tuple = []

D. empty_tuple = (,)

Answer: B

3Q: What is the purpose of tuple unpacking in Python?

A. Combining values into a tuple

B. Modifying existing tuples

C. Assigning values from a tuple to multiple variables

D. Creating a tuple from a list

Answer: C

4Q: Which method is used to find the sum of elements in a tuple?

A. sum()

B. total()

C. add()

D. calculate_sum()

Answer: A

5Q: What does the term "immutability" mean in the context of tuples?

A. Elements cannot be modified after creation

B. Elements can be modified at any time

C. Elements are mutable only during creation

D. Elements are fixed-size

Answer: A

6Q: Which of the following is a disadvantage of using tuples as keys in dictionaries?

A. Immutability

B. Limited flexibility

C. Dynamic size

D. Faster lookups

Answer: B

7Q: What is the purpose of the get() method in Python dictionaries?

A. Add a new key-value pair

B. Retrieve the value associated with a key

C. Delete a key-value pair

D. Sort the dictionary

Answer: B

8Q: Which method merges the keys and values of one dictionary into another in Python?

A. concat()

B. merge()

C. combine()

D. update()

Answer: D

9Q: What is the time complexity of common dictionary operations like insertion and retrieval?

A. O(1) average, O(n) worst case

B. O(log n)

C. O(n) average, O(1) worst case

D. O(n log n)

Answer: A

10Q: In dictionary comprehension, what is the purpose of the expression key: value?

A. Define a new key

B. Assign a value to a key

C. Specify a condition for a key

D. Create a new dictionary

Answer: B

Fill in the Blanks :

Q1: Tuples are _____, meaning their elements cannot be modified after creation.

Answer: immutable

Q2: The syntax for creating an empty tuple is _____.

Answer: empty_tuple = ()

Q3: Tuple unpacking is useful for assigning values from a tuple to _____ variables.

Answer: multiple

Q4: The update() method is used to _____ the keys and values of one dictionary into another.

Answer: merge

Q5: In Python dictionaries, the get() method returns None if the key is not _____.

Answer: present

Q6: Dictionary comprehension is a concise way to create dictionaries using _____ syntax.

Answer: compact

Q7: The time complexity of common dictionary operations like insertion and retrieval is O(_____).

Answer: 1

Q8: The expression key: value in dictionary comprehension specifies the _____ for a key.

Answer: value

Q9: Dictionary keys must be _____, meaning each key must be unique.

Answer: unique

Q10: The purpose of the pop() method in Python dictionaries is to _____ and return the value associated with a specified key.

Answer: remove