**HashMap in python**

In Python, a hash map is implemented using the built-in dict class, which represents a dictionary. The dict class provides an efficient way to store and retrieve key-value pairs. Here's how you can use a hash map (dictionary) in Python:

1. Creating a Hash Map:  
   You can create an empty hash map or initialize it with key-value pairs using curly braces {} or the dict() constructor. Here are a few examples:

python

# Empty hash map

my\_dict = {}

# Initializing with key-value pairs

my\_dict = {'key1': 'value1', 'key2': 'value2'}

# Using dict() constructor

my\_dict = dict({'key1': 'value1', 'key2': 'value2'})

```

1. Adding and Updating Elements:  
   To add or update elements in the hash map, you can assign a value to a specific key or use the update() method. If the key already exists, the value will be updated; otherwise, a new key-value pair will be added. Here's an example:

python

my\_dict = {}

my\_dict['key1'] = 'value1' # Adding a new key-value pair

my\_dict['key2'] = 'value2' # Adding another key-value pair

my\_dict['key1'] = 'new value' # Updating the value for 'key1'

# Using update() method

my\_dict.update({'key3': 'value3', 'key4': 'value4'}) # Adding multiple key-value pairs

```

1. Accessing Values:  
   You can access the values in the hash map by providing the corresponding key. If the key is not found, it will raise a KeyError. Here's an example:

python

my\_dict = {'key1': 'value1', 'key2': 'value2'}

value1 = my\_dict['key1'] # Accessing value using key

value2 = my\_dict.get('key2') # Accessing value using get() method

```

1. Removing Elements:  
   You can remove elements from the hash map using the del keyword or the pop() method. The del keyword deletes the key-value pair, while pop() removes the pair and returns the corresponding value. Here's an example:

python

my\_dict = {'key1': 'value1', 'key2': 'value2'}

del my\_dict['key1'] # Removing a specific key-value pair

value2 = my\_dict.pop('key2') # Removing and returning value using pop() method

```

1. Iterating over a Hash Map:  
   You can iterate over a hash map using a for loop to access its keys, values, or key-value pairs. Here's an example:

python

my\_dict = {'key1': 'value1', 'key2': 'value2'}

for key in my\_dict:

print(key) # Accessing keys

for value in my\_dict.values():

print(value) # Accessing values

for key, value in my\_dict.items():

print(key, value) # Accessing key-value pairs