字典的一些基本用法：

1. 創建字典：

可以使用大括號或 dict() 函數來創建字典。大括號用於創建空字典或非空字典。dict() 函數可以用來創建空字典或由鍵值對構成的字典。

例如：

empty\_dict = {}

non\_empty\_dict = {'a': 1, 'b': 2, 'c': 3}

another\_dict = dict(d=4, e=5, f=6)

1. 訪問字典元素：

可以通過字典中的鍵來訪問字典中的值。使用中括號和鍵來訪問值。如果該鍵不存在於字典中，則會引發 KeyError。

例如：

my\_dict = {'a': 1, 'b': 2, 'c': 3}

print(my\_dict['a']) # 輸出 1

print(my\_dict['d']) # 引發 KeyError

1. 添加或修改字典元素：

可以通過使用中括號和鍵來添加或修改字典中的值。如果鍵已經存在於字典中，則該值將被更新。否則，新鍵值對將被添加到字典中。

例如：

my\_dict = {'a': 1, 'b': 2, 'c': 3}

my\_dict['d'] = 4 # 添加新鍵值對

my\_dict['a'] = 5 # 修改值

print(my\_dict) # 輸出 {'a': 5, 'b': 2, 'c': 3, 'd': 4}

1. 刪除字典元素：

可以使用 del 關鍵字來刪除字典中的鍵值對。

例如：

my\_dict = {'a': 1, 'b': 2, 'c': 3}

del my\_dict['b'] # 刪除 'b' 鍵值對

print(my\_dict) # 輸出 {'a': 1, 'c': 3}

1. 字典方法：

字典支持許多有用的方法，例如 keys()、values() 和 items()。這些方法分別返回字典中的所有鍵、所有值和所有鍵值對。

例如：

my\_dict = {'a': 1, 'b': 2, 'c': 3}

print(my\_dict.keys()) # 輸出 dict\_keys(['a', 'b', 'c'])

print(my\_dict.values()) # 輸出 dict\_values([1, 2, 3])

print(my\_dict.items()) # 輸出 dict\_items([('a', 1), ('b', 2), ('c', 3)])

1. 檢查鍵是否存在：

可以使用 in 關鍵字來檢查

更進階的字典用法：

1. 判斷鍵是否存在

使用 **in** 關鍵字可以判斷一個鍵是否存在於字典中，例如：

my\_dict = {"apple": 1, "banana": 2, "orange": 3}

if "apple" in my\_dict:

print("The key 'apple' exists!")

else:

print("The key 'apple' does not exist.")

1. 獲取字典中所有的鍵或值

使用 **keys()** 方法可以獲取字典中所有的鍵，使用 **values()** 方法可以獲取字典中所有的值，例如：

my\_dict = {"apple": 1, "banana": 2, "orange": 3}

keys = my\_dict.keys()

values = my\_dict.values()

print("Keys:", keys)

print("Values:", values)

1. 合併兩個字典

可以使用 **update()** 方法將兩個字典合併，例如：

dict1 = {"apple": 1, "banana": 2}

dict2 = {"orange": 3, "peach": 4}

dict1.update(dict2)

print(dict1)

輸出結果為：{'apple': 1, 'banana': 2, 'orange': 3, 'peach': 4}

1. 遍歷字典

可以使用 **items()** 方法遍歷字典，獲取鍵和值，例如：

my\_dict = {"apple": 1, "banana": 2, "orange": 3}

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

print(key, value)

輸出結果為：

apple 1

banana 2

orange 3

<https://medium.com/ccclub/ccclub-python-for-beginners-tutorial-533b8d8d96f3\>

**range() 函數**

Python 中的 **range()** 函數用於創建一個整數範圍的序列。它通常與 **for** 迴圈一起使用，用於迭代指定範圍內的整數。

**range()** 函數接受 1 至 3 個整數參數：

* **start**: 整數範圍的起始值，默認為 0。
* **stop**: 整數範圍的終止值，但不包括此值。必須指定。
* **step**: 進行迭代的步長。默認為 1。

以下是使用 **range()** 函數的示例：

# 創建一個整數範圍的序列

my\_range = range(0, 10, 2)

# 遍歷範圍內的每個整數，並將其打印出來

for i in my\_range:

print(i)

輸出結果為：0

2

4

6

### 8 range() 函數的注意事項

以下是使用 **range()** 函數時需要注意的一些事項：

* 如果只給出一個整數參數，那麼它將被視為終止值，並且起始值將默認為 0，步長將默認為 1。
* 如果給出兩個整數參數，那麼它們將被視為起始值和終止值，步長將默認為 1。
* 如果給出三個整數參數，那麼它們將被視為起始值、終止值和步長。

**練習題**

以下是一些練習題，可以用來熟悉 **range()** 函數的使用：

1. 打印出一個範圍內的所有奇數，範圍從 1 到 10。
2. 計算一個範圍內的所有偶數的和，範圍從 1 到 20。
3. 打印出一個範圍內的所有 5 的倍數，範圍從 10 到 50。
4. 創建一個包含 10 個整數的列表，這些整數的值從 0 到 9。可以使用 **range()** 函數和列表解析來完成此操作。