**Full Walkthrough**

**Objective:**

The objective of this challenge is to decrypt the morse code provided in form of sounds, a general guide for morse code such as small beep stands for (‘.”) , longer beep (“-”) and two second silence for a space in between characters is provided. Finally, the user is required to use a python file to decrypt this challenge.

**Description:**

Morse Code is a type of communication. It uses dots, dashes, and spaces to represent letters and numbers. These dots and dashes are arranged to spell out a message. Morse Code has been extensively used in the 19th century during the world war ii period. Battleships and war planes could communicate over long distances and pass on information back to mainland Britain. Each ship/war plane had a morse code decrypting key which was used to decrypt any incoming messages and displayed it to the user. The Morse code is still in use today, Since radio navigational aids like VORs and NDBs still identify in it, aviation and aeronautical fields use morse code more frequently than other industries. Signal lights are still used by the US Navy and Coast Guard to communicate via Morse code.

The User is provided with MorseCode Dictionary. The Morse Code dictionary is the encryption key that is used to encrypt the “FLAG” in this challenge. Each character in the English alphabet has a morse code, use the morse code dictionary to figure out the character from the morse code provided. The morse code is provided to the user in forms of sound.mav file. Upon running the application, sound such as beeps will be produced. Small beep stands for dot ( “ . “ ) and a longer beep stands for dash ( “ - “ ). Additionally, A silence for 2 sec stands for a space in between. The User is required to write down all the beeps on note in form of .’s and -’s. Lastly using the Morse Code Dictionary, write a python script which translates the Morse Code provided into English Character / Flag

**Understanding Morse Code Dictionary:**

MORSE\_CODE\_DICT = { 'A':'.-', 'B':'-...',

'C':'-.-.', 'D':'-..', 'E':'.',

'F':'..-.', 'G':'--.', 'H':'....',

'I':'..', 'J':'.---', 'K':'-.-',

'L':'.-..', 'M':'--', 'N':'-.',

'O':'---', 'P':'.--.', 'Q':'--.-',

'R':'.-.', 'S':'...', 'T':'-',

'U':'..-', 'V':'...-', 'W':'.--',

'X':'-..-', 'Y':'-.--', 'Z':'--..',

'1':'.----', '2':'..---', '3':'...--',

'4':'....-', '5':'.....', '6':'-....',

'7':'--...', '8':'---..', '9':'----.',

'0':'-----', ', ':'--..--', '.':'.-.-.-',

'?':'..--..', '/':'-..-.', '-':'-....-',

'(':'-.--.', ')':'-.--.-'}

The morse code dictionary above is provided in a python file. The morse code is organised in a python function called dictionaries. Example: → Morse\_Code\_Dictionary = { “A” : “B” }, where A is the key for the dictionary function and B is the value for the Dictionary function called Morse\_Code\_Dict. The Morse Code Dictionary represents the Morse code for each character.

**Walkthrough:**

**Step 1:** Run the python program provided and listen to the sound. Small beep for “ . “ and longer beep for “ - “. Lastly, 2 second silence stands for a space in between. Users are required to note down these .’s and -’s. The expected output is to be as followed:

..-. .-.. .- --.

**Step 2:** To iterate through the noted morse code add the above noted Morse code in a list separated by space in between. The expected output is to be as followed:

Array\_Translated\_Morse\_Code = [ '..-.','.-..', '.-', '--.' ]

**Step 3:** Iterate through Array\_Translated\_Code and set the first items and of the list as default and continue this process.

**For i in array\_Translated\_Morse\_Code:**

**Check = i**

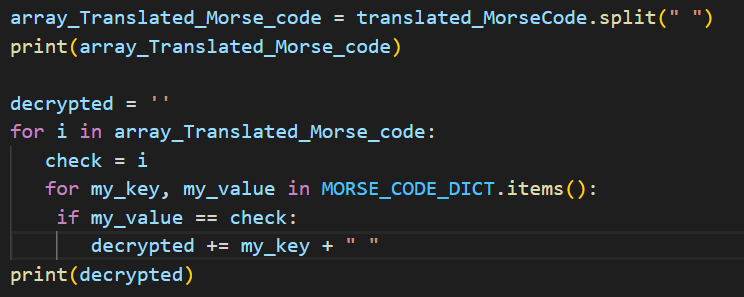
**Step 4:** iterate through the python dictionary called morse code dictionary. The key/important part is to understand how python iterates through a dictionary, as well as how to get the key and value from the dictionary. The Concepts are explained below and the expected codes are as follows:

**Step 5:** Check if provided morse Code matches with any of the values in Morse Code Dictionary using If statements. If any of the keys matches then add the key to a string variable.The variable will be the flag which is to be found.

**If my\_key, my\_value in MORSE\_CODE\_DICT.items():**

**Decrypted += my\_key + “ “**

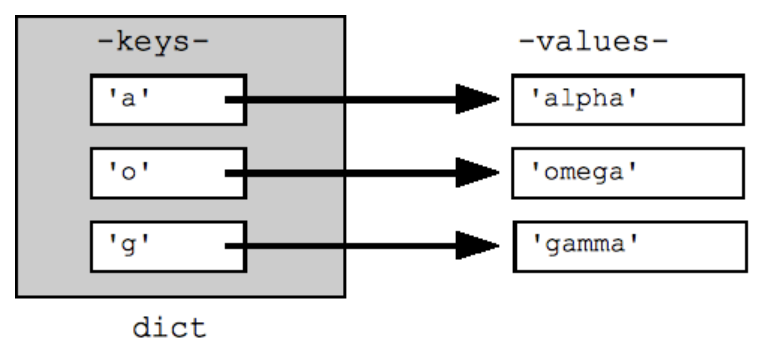
**Expected Solution:**



**FLAG FOR THIS CHALLENGE: “F L A G”**

**Theories**

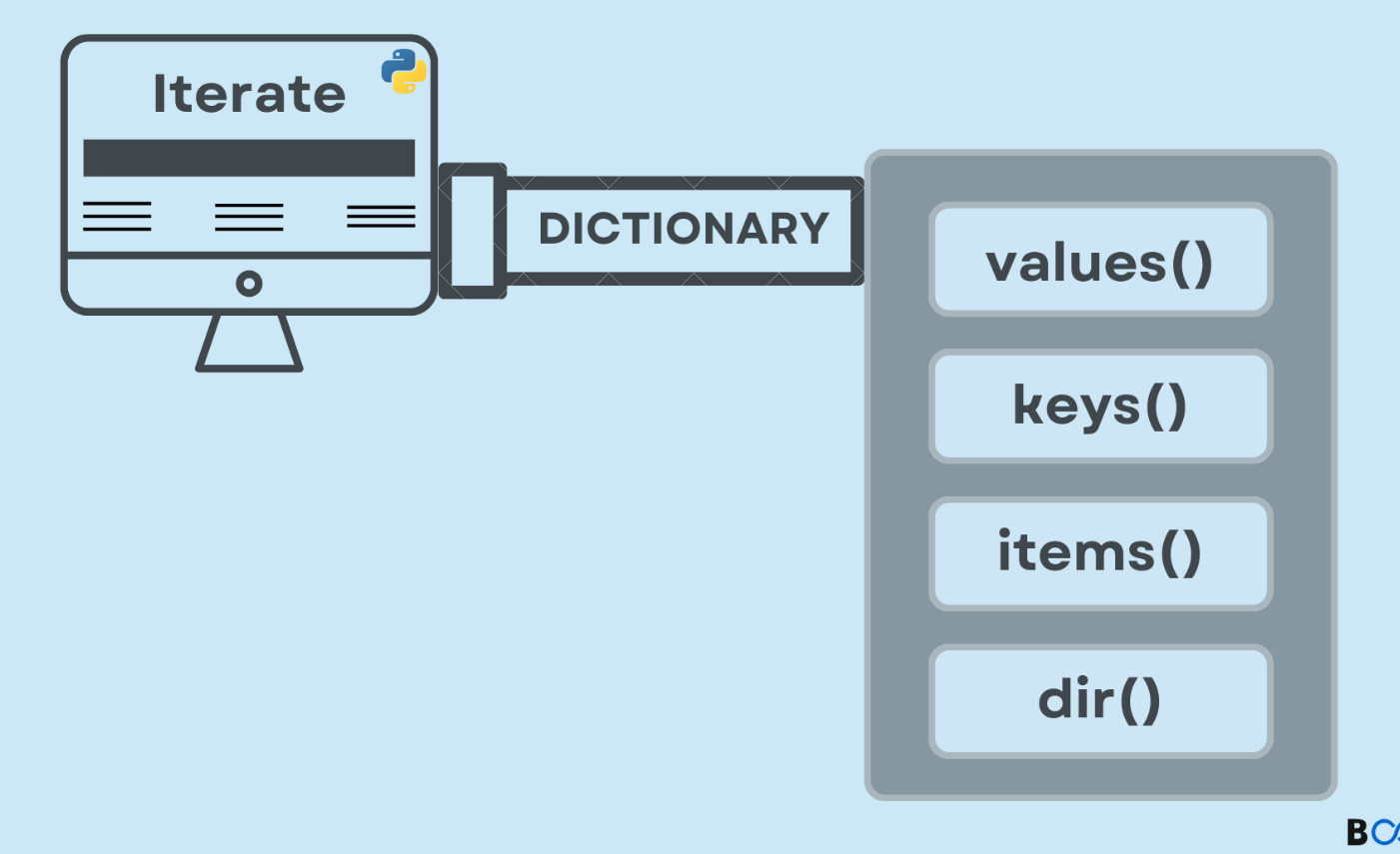
**Python Keys / Values in Dictionary:**



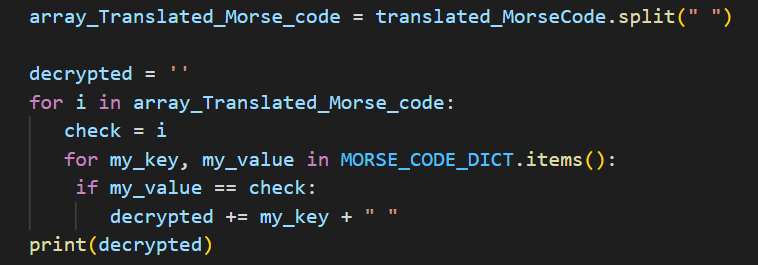
Morse\_Code\_Dictionary = { “A” : “B” }, where A is the key for the dictionary function and B is the value for the Dictionary function called Morse\_Code\_Dict. The Morse Code Dictionary represents the Morse code for each character.

**How to iterate over python dictionary:**

In Python, to iterate through a dictionary (dict) with a for loop, use the keys(), values(), and items() methods. You can also get a list of all keys and values in the dictionary with those methods and list().

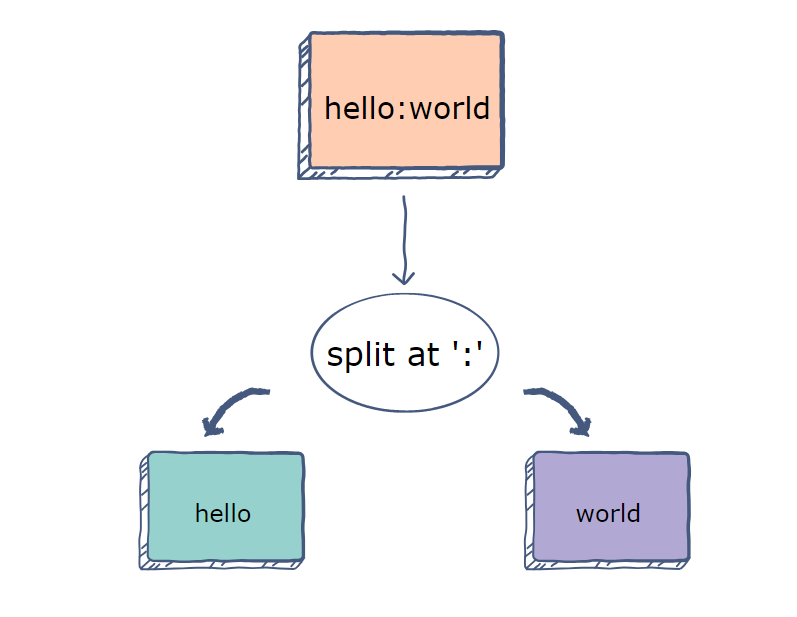


**This Theory used in the project:**



**How split method is used:**

The split method is used to split the strings by a specific command used in the middle such as .split(“ “). This method means to split strings by spaces. Then store the string in a variable. The result is stored in an array/list.



‌

**Bibliography:**

* **Educative: Interactive Courses for Software Developers. (n.d.). *How to execute a string split() function in Python*. [online] Available at:** [**https://www.educative.io/answers/how-to-execute-a-string-split-function-in-python**](https://www.educative.io/answers/how-to-execute-a-string-split-function-in-python) **[Accessed 8 June. 2023].**
* ***Python dict and file | python education | google for developers* (no date) *Google*. Available at:** [**https://developers.google.com/edu/python/dict-files**](https://developers.google.com/edu/python/dict-files) **(Accessed: 08 June 2023).**