**Assignment 16**

1. Create a list called years\_list, starting with the year of your birth, and each year thereafter until

the year of your fifth birthday. For example, if you were born in 1980. the list would be years\_list =

[1980, 1981, 1982, 1983, 1984, 1985].

Here's the solution to the first part of your assignment, which creates a list of years starting from the year of your birth until your fifth birthday:

python

Copy code

years\_list = [year for year in range(1995, 2000)]

print(years\_list)

Assuming your birth year is 1995, this will output:

yaml

Copy code

[1995, 1996, 1997, 1998, 1999]

You can modify the range function arguments based on your birth year.

2. In which year in years\_list was your third birthday? Remember, you were 0 years of age for your

first year.

If you were born in 1995, then your third birthday would have been in the year 1998.

This is because your first year of age was from 1995 to 1996, your second year of age was from 1996 to 1997, and your third year of age was from 1997 to 1998. So, 1998 was the year in years\_list when you turned three years old.

3.In the years list, which year were you the oldest?

If the years\_list starts from the year of your birth and goes up to the year of your fifth birthday, then you were the oldest in the last year of the list, which would be the year you turned 4.

Continuing from the previous example, if you were born in 1995, the years\_list would be [1995, 1996, 1997, 1998, 1999], and in this case, you were the oldest in the year 1999.

4. Make a list called things with these three strings as elements: “mozzarella”, “cinderella”,

“salmonella”.

Here's the solution to create a list called things with the three strings as elements:

python

Copy code

things = ["mozzarella", "cinderella", "salmonella"]

print(things)

This will output:

css

Copy code

["mozzarella", "cinderella", "salmonella"]

Now the things list contains the three strings "mozzarella", "cinderella", and "salmonella" as its elements.

5. Capitalize the element in things that refers to a person and then print the list. Did it change the

element in the list?

Yes, we can capitalize the element in things that refers to a person, which is "cinderella", using the capitalize() method in Python.

Here's how to do it:

python

Copy code

things = ["mozzarella", "cinderella", "salmonella"]

things[1] = things[1].capitalize()

print(things)

This will output:

css

Copy code

["mozzarella", "Cinderella", "salmonella"]

As you can see, the second element of the things list, which was "cinderella", has been changed to "Cinderella" after applying the capitalize() method. This method capitalizes the first letter of the string, while keeping the rest of the string as it is.

Note that the capitalize() method returns a new string and does not modify the original string, which is why we need to assign the capitalized string back to the same index of the things list.

6. Make a surprise list with the elements “Groucho,” “Chico,” and “Harpo.”

Here's the solution to create a list called surprise with the elements "Groucho", "Chico", and "Harpo":

python

Copy code

surprise = ["Groucho", "Chico", "Harpo"]

print(surprise)

This will output:

css

Copy code

["Groucho", "Chico", "Harpo"]

Now the surprise list contains the three elements "Groucho", "Chico", and "Harpo".

7. Lowercase the last element of the surprise list, reverse it, and then capitalize it.

Sure, here's how to lowercase the last element of the surprise list, reverse it, and then capitalize it:

python

Copy code

surprise = ["Groucho", "Chico", "Harpo"]

last\_element = surprise[-1].lower() # lowercase the last element

reversed\_element = last\_element[::-1] # reverse the lowercase element

capitalized\_element = reversed\_element.capitalize() # capitalize the reversed element

surprise[-1] = capitalized\_element # assign the capitalized element back to the last index

print(surprise)

This will output:

css

Copy code

["Groucho", "Chico", "Oprah"]

As you can see, the last element of the surprise list has been changed from "Harpo" to "Oprah". The original string was first converted to lowercase using the lower() method, then reversed using slicing with step -1, and finally capitalized using the capitalize() method. The capitalized string was then assigned back to the last index of the surprise list.

8. Make an English-to-French dictionary called e2f and print it. Here are your starter words: dog is

chien, cat is chat, and walrus is morse.

Here's how to create an English-to-French dictionary called e2f with the starter words and print it:

python

Copy code

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

print(e2f)

This will output:

arduino

Copy code

{'dog': 'chien', 'cat': 'chat', 'walrus': 'morse'}

Now the e2f dictionary contains the English words "dog", "cat", and "walrus" as keys, and their corresponding French translations "chien", "chat", and "morse" as values.

9. Write the French word for walrus in your three-word dictionary e2f.

The French word for "walrus" in the e2f dictionary is "morse".

To access the French word for "walrus" in e2f, we can use its corresponding English word "walrus" as the key, like this:

python

Copy code

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

french\_word = e2f["walrus"]

print(french\_word)

This will output:

Copy code

morse

As you can see, the French word for "walrus" in the e2f dictionary is "morse".

10. Make a French-to-English dictionary called f2e from e2f. Use the items method.

Here's how to create a French-to-English dictionary called f2e from e2f using the items() method:

python

Copy code

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

# create f2e dictionary by swapping keys and values of e2f dictionary using items() method

f2e = {value: key for key, value in e2f.items()}

print(f2e)

This will output:

arduino

Copy code

{'chien': 'dog', 'chat': 'cat', 'morse': 'walrus'}

As you can see, the f2e dictionary contains the French words "chien", "chat", and "morse" as keys, and their corresponding English translations "dog", "cat", and "walrus" as values. We created the f2e dictionary by swapping the keys and values of the e2f dictionary using a dictionary comprehension with the items() method.

11. Print the English version of the French word chien using f2e.

To print the English version of the French word "chien" using f2e, we can use "chien" as the key to access its corresponding English translation:

python

Copy code

f2e = {

"chien": "dog",

"chat": "cat",

"morse": "walrus"

}

english\_word = f2e["chien"]

print(english\_word)

This will output:

Copy code

dog

As you can see, the English version of the French word "chien" is "dog", which is the value corresponding to the key "chien" in the f2e dictionary.

12. Make and print a set of English words from the keys in e2f.

To make a set of English words from the keys in e2f dictionary and print it, we can use the set() function on the keys of the dictionary:

python

Copy code

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

english\_words = set(e2f.keys())

print(english\_words)

This will output:

arduino

Copy code

{'dog', 'cat', 'walrus'}

As you can see, we created a set of the English words "dog", "cat", and "walrus" from the keys of the e2f dictionary.

13. Make a multilevel dictionary called life. Use these strings for the topmost keys: ‘animals’, ‘plants’,

and ‘other’. Make the ‘animals’ key refer to another dictionary with the keys ‘cats’, ‘octopi’, and

‘emus’. Make the ‘cats’ key refer to a list of strings with the values ‘Henri’, ‘Grumpy’, and ‘Lucy’.

Make all the other keys refer to empty dictionaries.

Here's how to create a multilevel dictionary called life as described:

python

Copy code

life = {

"animals": {

"cats": ["Henri", "Grumpy", "Lucy"],

"octopi": {},

"emus": {}

},

"plants": {},

"other": {}

}

print(life)

This will output:

arduino

Copy code

{

'animals': {'cats': ['Henri', 'Grumpy', 'Lucy'], 'octopi': {}, 'emus': {}},

'plants': {},

'other': {}

}

As you can see, the life dictionary has three top-level keys: "animals", "plants", and "other". The "animals" key refers to another dictionary with three keys: "cats", "octopi", and "emus". The "cats" key refers to a list of strings with the values "Henri", "Grumpy", and "Lucy". The other keys ("octopi", "emus", "plants", and "other") refer to empty dictionaries.

14. Print the top-level keys of life.

To print the top-level keys of the life dictionary, we can use the keys() method:

python

Copy code

life = {

"animals": {

"cats": ["Henri", "Grumpy", "Lucy"],

"octopi": {},

"emus": {}

},

"plants": {},

"other": {}

}

top\_keys = life.keys()

print(top\_keys)

This will output:

css

Copy code

dict\_keys(['animals', 'plants', 'other'])

As you can see, the keys() method returns a view object of the keys in the life dictionary, which we then printed using print().

15. Print the keys for life[‘animals’].

To print the keys for the dictionary that the key 'animals' in life refers to, we can simply access it using life['animals'] and then call the keys() method on it:

python

Copy code

life = {

"animals": {

"cats": ["Henri", "Grumpy", "Lucy"],

"octopi": {},

"emus": {}

},

"plants": {},

"other": {}

}

animals\_keys = life['animals'].keys()

print(animals\_keys)

This will output:

css

Copy code

dict\_keys(['cats', 'octopi', 'emus'])

As you can see, we printed the keys for the dictionary that the key 'animals' in life refers to, which are 'cats', 'octopi', and 'emus'.

16. Print the values for life[‘animals’][‘cats’]

To print the values for the list that the key 'cats' in life['animals'] refers to, we can simply access it using life['animals']['cats']:

python

Copy code

life = {

"animals": {

"cats": ["Henri", "Grumpy", "Lucy"],

"octopi": {},

"emus": {}

},

"plants": {},

"other": {}

}

cats\_values = life['animals']['cats']

print(cats\_values)

This will output:

css

Copy code

['Henri', 'Grumpy', 'Lucy']

As you can see, we printed the values for the list that the key 'cats' in life['animals'] refers to, which are 'Henri', 'Grumpy', and 'Lucy'.