1. What does an empty dictionary's code look like?

D={}

D=dict() : dict is a keyword to represent dictionary

1. What is the value of a dictionary value with the key 'foo' and the value 42?

D={“foo”:42}

1. What is the most significant distinction between a dictionary and a list?

Dictionary is always in key value pair & key should always be unique,but list are not like that.

Dictionaries are written in {} braces while list are written in [] braces.

Dictionaries are unordered while list are ordered.

1. What happens if you try to access spam['foo'] if spam is {'bar': 100}?

Will get this error:**KeyError**: 'foo'

1. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys()?

.key() is used to check whether the item is present in dictionary as key ar not. So if cat is present as key in the spam than there is no difference.

1. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()?

values() is used to check whether the item is present in dictionary as value ar not. So if cat is present as value in the spam than there is no difference.

7. What is a shortcut for the following code?

if 'color' not in spam:

spam['color'] = 'black'

spam={"bar":100}

spam.setdefault("color","black")

output: {'bar': 100, 'color': 'black'}

1. How do you "pretty print" dictionary values using which module and function?

[pprint](https://docs.python.org/3/library/pprint.html) is a Python module that provides the capability to pretty print Python data types to be more readable. This module also supports pretty-printing dictionary.

Within the pprint module there is a function with the same name pprint(), which is the function used to pretty-print the given string or object.

Here are the functions:

a.pprint()

Example:

import pprint

dct\_arr = [

{'Name': 'John', 'Age': '23', 'Country': 'USA'},

{'Name': 'Jose', 'Age': '44', 'Country': 'Spain'},

{'Name': 'Anne', 'Age': '29', 'Country': 'UK'},

{'Name': 'Lee', 'Age': '35', 'Country': 'Japan'}

]

pprint.pprint(dct\_arr)

output:

[{'Age': '23', 'Country': 'USA', 'Name': 'John'},

{'Age': '44', 'Country': 'Spain', 'Name': 'Jose'},

{'Age': '29', 'Country': 'UK', 'Name': 'Anne'},

{'Age': '35', 'Country': 'Japan', 'Name': 'Lee'}]

b.json.dumps()  
Within the Python json module, there is a function called dumps(), which converts a Python object into a JSON string. Aside from the conversion, it also formats the dictionary into a pretty JSON format, so this can be a viable way to pretty print a dictionary by first converting it into JSON.

The dumps() function accepts 3 parameters used for pretty printing: the object for conversion, a boolean value sort\_keys, which determines whether the entries should be sorted by key, and indent, which specifies the number of spaces for indentation.

Example:

import json

dct\_arr = [

{'Name': 'John', 'Age': '23', 'Country': 'USA'},

{'Name': 'Jose', 'Age': '44', 'Country': 'Spain'},

{'Name': 'Anne', 'Age': '29', 'Country': 'UK'},

{'Name': 'Lee', 'Age': '35', 'Country': 'Japan'}

]

print(json.dumps(dct\_arr, sort\_keys=False, indent=4))

output:

[

{

"Name": "John",

"Age": "23",

"Country": "USA"

},

{

"Name": "Jose",

"Age": "44",

"Country": "Spain"

},

{

"Name": "Anne",

"Age": "29",

"Country": "UK"

},

{

"Name": "Lee",

"Age": "35",

"Country": "Japan"

}

]

c.yaml.dumps()

Another way to pretty print a dictionary is by using the dump() function of the yaml module. It serves the same purpose as the json.dumps() function but in YAML format instead of JSON.

default\_flow\_style, which determines whether the dump’s output style should be inline or block

Example:

- Name: John

Age: '23'

Residence:

Country: USA

City: New York

- Name: Jose

Age: '44'

Residence:

Country: Spain

City: Madrid

- Name: Anne

Age: '29'

Residence:

Country: UK

City: England

- Name: Lee

Age: '35'

Residence:

Country: Japan

City: Osaka