Dictionary

Dictionary simply consists of Key - Value pair.

where,

Key = Question

Value = Answer

Syntax

==> {key : value}

==> {key 1 : value 1, key 2 : value 2 ..... key n : value n}

In [3]:

```
# Example

dct = {
    "bug" : 'An error in a program that prevents the program for running as expected',
    "function" : 'A peice of code that you can easily call over and over again',
    "loop" : 'The action of doing something over and over again'
}

print(dct)
```

```
{'bug': 'An error in a program that prevents the program for running as ex
pected', 'function': 'A peice of code that you can easily call over and ov
er again', 'loop': 'The action of doing something over and over again'}
```

==> ***dictionary[key]*** gives the value of the relevant key

In [4]:

```
print(dct["bug"])
```

```
An error in a program that prevents the program for running as expected
```

==> ***dictionary[key] = value*** is used for adding and editing an item in dictionary

In [10]:

```
dct['source'] = 'Jupyter notebook' # Adding an item

print(dct)
```

```
{'bug': 'An error in a program that prevents the program for running as ex
pected', 'function': 'A peice of code that you can easily call over and ov
er again', 'loop': 'The action of doing something over and over again', 's
ource': 'Jupyter notebook'}
```

In [11]:

```
dct['source'] = 'google colab'       # Editing an item
print(dct)
```

```
{'bug': 'An error in a program that prevents the program for running as ex
pected', 'function': 'A peice of code that you can easily call over and ov
er again', 'loop': 'The action of doing something over and over again', 's
ource': 'google colab'}
```

==> ***dictionary = {}*** is used to wipe an dictionary

In [12]:

```
dct = {}         # wiping an dictionary
print(dct)
```

```
{}
```

Looping through an dictionary

In [14]:

```
dct_1 = {
    'name' : 'aswin',
    'age' : 19,
    'course' : 'Python'
}

for key in dct_1:
    print(f"key ==> {key} \t value ==> {dct_1[key]}")
```

```
key ==> name     value ==> aswin
key ==> age      value ==> 19
key ==> course   value ==> Python
```

In dictionaries, we can add dictionaries, list, tuple, string, and any data type.

Task

Convert marks of students in a given dictionary into grades and use empty dictionary "student_grades" for appending converted values.

91 - 100 = 'outstanding'

81 - 90 = 'exceeds expectation'

71 - 80 = 'acceptable'

below 70 = 'Fail'

In [21]:

```
# Given code

student_scores = {

    "Harry" : 81,

    "Ron" : 78,

    "Hermione" : 99,

    "Draco" : 62
}

# Your Code

student_grades = {}

for key in student_scores:

    if student_scores[key] <= 99 and student_scores[key] >= 91:

        student_grades[key] = "Outstanding"

    elif student_scores[key] <= 90 and student_scores[key] >= 81:

        student_grades[key] = "Exceeds Expectation"

    elif student_scores[key] <= 80 and student_scores[key] >= 71:

        student_grades[key] = "Acceptable"

    else:
        student_grades[key] = "Fail"

print(student_grades)
```

```
{'Harry': 'Exceeds Expectation', 'Ron': 'Acceptable', 'Hermione': 'Outstan
ding', 'Draco': 'Fail'}
```