FileHandling in Python

Text file

In [8]:
```
# open() ---> in-build
# mode --
# read   --r
# write  --- w
# append  -- a
# binary ---b
file_object = open('sample.txt',mode='w')
```

In [10]:
```
file_object.write('this is my second text message')
```

Out[10]: 30

In [11]:
```
file_object.close()
```

In [12]:
```
file_object = open('sample.txt', mode='r')
file_object.read()
```

Out[12]: 'this is my second text message'

In [13]:
```
file_object.close()
```

In [16]:
```
file_object = open('sample.txt', mode='a+')
file_object.write('this is my third text message')
```

Out[16]: 29

In [17]:
```
file_object.close()
```

In [18]:
```
file_object = open('sample.txt',mode='r')
file_object.read()
```

Out[18]: 'this is my second text messagethis is my third text message'

In [20]:
```
with open('sample.txt', mode='a') as fa:
    fa.write('this is my new text message in file append mode \n')
```

In [22]:
```
with open('sample.txt', mode='r') as fa:
    read_text_sample = fa.read()
```

In [23]:
```
read_text_sample
```

Out[23]:
```
'this is my second text messagethis is my third text messagethis is my new te
xt message in file append mode \n'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Json file

In [38]:
```
import json
# import python keyword
# json library name

json_file_object  = open('example_1.json')
```

In [39]:
```
json_file_read = json.load(json_file_object)
```

In [41]:
```
type(json_file_read)
```

Out[41]:
```
dict
```

In [43]:
```
json_file_read.keys()
```

Out[43]:
```
dict_keys(['fruit', 'size', 'color'])
```

In [44]:
```
json_file_read['fruit']
```

Out[44]:
```
'Apple'
```

In [45]:
```
json_file_read['size']
```

Out[45]:
```
'Large'
```

In [46]:
```
sample_json_data = {
    'name':'mehul wankhede',
    'age':27,
    'job':'Software engineer',
    'mob':982222222
}
```

In [53]:
```
file_data_json = open('sample_json_data.json', mode='w')
```

In [54]:
```
json.dump(sample_json_data, file_data_json)
```

In [55]:
```
file_data_json.close()
```

In [ ]:

In [ ]:

In [28]:
```
my_dict = {'name':'khushi', 'age':20}
```

In [31]:
```
m
```

Out[31]:
```
('khushi', 20)
```

Pickle File

In [56]:
```
import pickle
```

In [57]:
```
data_list = [1,2,3,4,5,6,6,7,8]
```

In [58]:
```
type(data_list)
```

Out[58]:
```
list
```

In [ ]:
```
# syntax pickle.dump(data_object, file_object)
# to dump data into file
# # syntax pickle.load(file_object)
# to read data from file
```

In [61]:
```
with open('sample_list.pkl', mode='wb') as fw:
    pickle.dump(data_list, fw)
```

In [62]:
```
with open('sample_list.pkl', mode='rb') as fr:
    read_picle_data = pickle.load(fr)
```

In [63]:
```
read_picle_data
```

Out[63]:
```
[1, 2, 3, 4, 5, 6, 6, 7, 8]
```

In [65]:
```
smaple_list = [1,2,3,4,]
```

In [66]:
```
smaple_list
```

Out[66]:
```
[1, 2, 3, 4]
```

In [68]:
```
str(smaple_list)
```

Out[68]:
```
'[1, 2, 3, 4]'
```

In [69]:
```
float(smaple_list)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [69], in <cell line: 1>()
----> 1 float(smaple_list)

TypeError: float() argument must be a string or a number, not 'list'
```

In [70]:
```
tuple_ = (1,2,3,4)
```

In [71]:
```
tuple_[0] = 10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [71], in <cell line: 1>()
----> 1 tuple_[0] = 10

TypeError: 'tuple' object does not support item assignment
```

In [72]:
```
with open('dsalfjaslfjl.txt', mode='r') as fr:
    fr.read()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [72], in <cell line: 1>()
----> 1 with open('dsalfjaslfjl.txt', mode='r') as fr:
      2     fr.read()

FileNotFoundError: [Errno 2] No such file or directory: 'dsalfjaslfjl.txt'
```

Exception Handling

In [75]:
```
number = 30
my_list = [2,3, 5, 6, 0, 1, 4]

for value in my_list:
    try:
        print('result :', number/value)
    except Exception as e:
        print('error',e)
```

```
result : 15.0
result : 10.0
result : 6.0
result : 5.0
error division by zero
result : 30.0
result : 7.5
```

In [76]:
```
my_dict
```

Out[76]:
```
{'name': 'khushi', 'age': 20}
```

In [77]:
```
my_dict['name']
```

Out[77]:
```
'khushi'
```

In [81]:
```
try:
    print(my_dict['address'])
except Exception as e :
    print('error ',e)
```

```
error  'address'
```

In [82]:
```
# Implement a function that transforms a dictionary by swapping keys and value
my_dict = {'name':'mehul', 'age':27,'address':'Taloda'}
```

In [84]:
```
swaaping_dict = {'mehul':'name', 27:'age', 'Taloda':'address'}
```

In [85]:
```
# step 1 -->
keys_list = list(my_dict.keys())
values_list = list(my_dict.values())
```

In [86]:
```
swaping_dict = {}
for index, value in enumerate(values_list):
    swaping_dict[value] = keys_list[index]
```

In [87]:
```
swaping_dict
```

Out[87]:
```
{'mehul': 'name', 27: 'age', 'Taloda': 'address'}
```

In [91]:
```
number = 10
is_prime = True
for value in range(2,number):
    if number%value == 0:
        is_prime = False
if is_prime:
    print(f'number {number} is prime number')
else:
    print(f'number {number} is not prime number')
```

```
number 10 is not prime number
```

In [94]:
```
my_list = [2,4,6,2, 1,3]

even_list = [value for value in my_list  if value %2 == 0]
odd_list = [value for value in my_list if value%2!=0 ]
```

In [95]:
```
even_list
```

Out[95]:
```
[2, 4, 6, 2]
```

In [102]:
```
 odd_list
```

Out[102]:
```
[1, 3]
```

In [103]:
```
# range(len(even_list)/2)
# list append
# odd range(len(odd_list) * 2 )
# list append
```

In [ ]:
```

```