File handling is an important part of any application.

Python has several functions for creating, reading, updating, and deleting files.

File Handling

The key function for working with files in Python is the `open()` function.

The `open()` function takes two parameters; `filename` , and `mode` .

read file

The following mode is used for opening a file for reading:

- `"r"` - Read - Default value. Opens a file for reading, error if the file does not exist

In addition you can specify if the file should be handled as binary or text mode

- `"t"` - Text - Default value. Text mode
- `"b"` - Binary - Binary mode (e.g. images)

In [ ]:
```
# To open a file for reading it is enough to specify the name of the file and mode:
f = open("sample.txt", "rt")
```

In [ ]:
```
# OR same thing
f = open("sample.txt")
# because "r" for read, and "t" for text are the default values, you do not need to spe
```

In [ ]:
```
# use .read() method for reading the content of the file:
print(f.read())
```

In [ ]:
```
# OR if the file is too long
# then we can read only the 5 first characters of the file:

f = open("sample.txt", "r")
print(f.read(5))
```

write to file

There are two modes for opening a file for writing:

- `"a"` - Append - Opens a file for appending, creates the file if it does not exist
- `"w"` - Write - Opens a file for writing, creates the file if it does not exist

In [ ]:
```
# warning 'w' mode erases all content of the file
# is your file becomes new and empty
f = open("sample2.txt", "w")
f.write("My first ever string saved to file\n")
f.close()

#open and read the file after the appending:
f = open("sample2.txt", "r")
print(f.read())
```

In [ ]:
```
# to add data to file provide explicitly 'a' mode
f = open("sample2.txt", "a")
f.write("My second string\n")
f.close()

#open and read the file after the appending:
f = open("sample2.txt", "r")
print(f.read())
```

In [ ]:
```
# as we don't know the exact number of symbols in file
# we will prefer to read files line by line using for loop

c = 1
for line in open('sample.txt'):
    print ('line ' + str(c) + ': ' + line )
    c += 1
```

enumerate function

`enumerate()` is a built-in function of Python. It allows us to loop over something and have an automatic counter.

In [ ]:
```
my_list = ['nol', 'bir', 'eki', 'ush']
for counter, value in enumerate(my_list):
    print(counter, value)
```

In [ ]:
```
# we can use enumerate function to count lines:
for counter, line in enumerate(open("sample.txt")):
    print('line ' + str(counter) + ': ' + line)
```

.split() method

The `.split()` method splits a string into a list.

You can specify the separator, default separator is any whitespace.

In [ ]:
```
s = 'Happy Birthday to You'
s_list = s.split(' ')
print(s_list)
```

In [ ]:
```
# split sentences
txt = "Hello. My name is Peter. I am 26 years old."
x = txt.split(". ")
print(x)
```

Practice

In [ ]:
```

```