13.Input_Output

January 20, 2019

1 IO

1.1 Input-Output Purpose

- read data from files
- write data to files

How to perform it?
There is an `open()` function to open files

```
In [2]: file = open('test_file', 'w')
```

1st argument is a name of file, while the second is a mode of operation. Mode consists of 3 parts: 1. operation mode * **r** - read file, writing to file is prohibited, default * **w** - write to file, if file was before, it will delete all its content * **a** - append file, new data will be written below previous * **x** - create new file, if file exists before, it will throw an error 2. file content * **t** - plain text, default * **b** - binary 3. additional **+** - for both reading and writing
To write a text to the file we use `write()` method. It returns # of written characters

```
In [3]: # This method returns
        file.write('A new line of text in this file\n')
```

```
Out[3]: 32
```

```
In [39]: file.write('Another line')
```

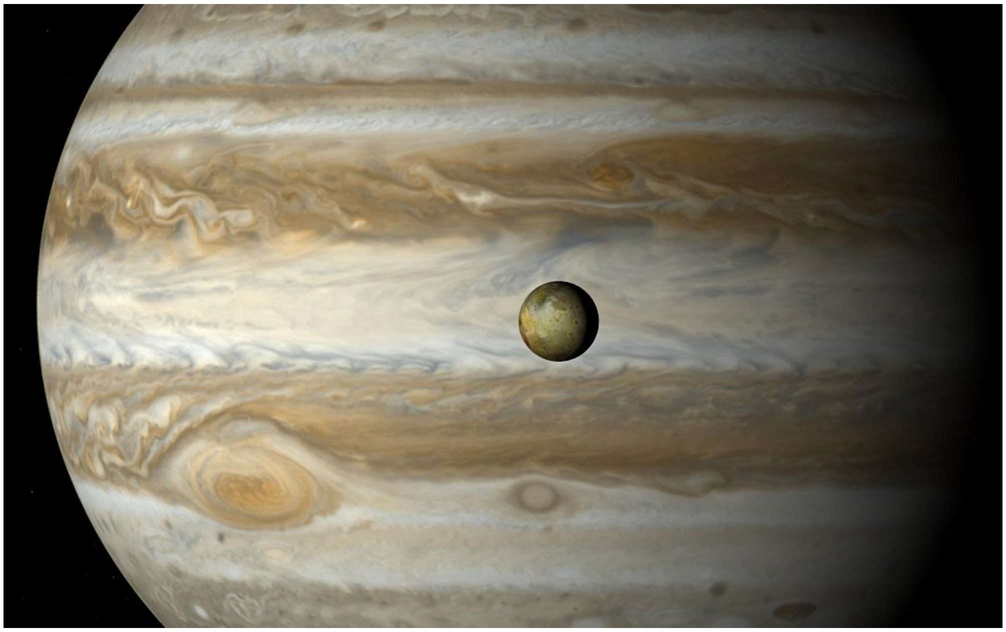
```
Out[39]: 12
```

After working with file it should be closed, opened abundoned files isn't a good thing. For this task we have a method `close()`

```
In [40]: file.close()
```

There is a better way for opening and closing than that. It utilizes context manager. This is a preferrable way

```
In [41]: with open('test_file', 'a') as file:
             file.write(' continuation of line\n')
```



```
# Open filename in mode and assign to variable_name
with open(filename, mode) as variable_name:
    operations
# Here file is already closed!
```

Now let's talk about reading from file. There are several ways to do it

```
In [43]: with open('test_file') as file:
             text = file.read()
             print(text)
```

```
A new line of text in this file
Another line continuation of line
```

```
In [44]: with open('test_file') as file:
             text = file.readlines()
             print(text)
```

```
['A new line of text in this file\n', 'Another line continuation of line\n']
```

```
In [46]: with open('test_file') as file:
             first = file.readline()
             print(repr(first))
             print(file.read())
```

```
'A new line of text in this file\n'
Another line continuation of line
```

```
In [47]: with open('test_file') as file:
             for line in file:
                 print(line)
```

```
A new line of text in this file

Another line continuation of line
```

Some methods for reading * `read()` - read whole file into RAM (usually not recommended) * `readline()` - read 1 line from the file * `readlines()` - read whole file and return list of lines splitted by newline characters * `seek(position)` - remote file to a given position * `readable()` - whether file can be read