1.Read a jpeg image and print the image file

In [ ]:

```
from PIL import Image
```

In [ ]:

```
a=Image.open("/content/python.jpg")
```

Prints FUll size image

In [ ]:

```
a
```

Out[ ]:

For accurate image

In [ ]:

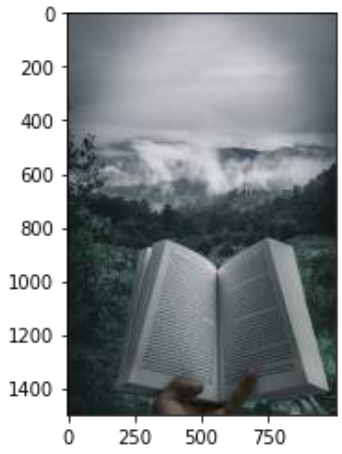
```
import matplotlib.pyplot as plt
```

In [ ]:

```
plt.imshow(a)
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x7f8d7f284bd0>
```



2.Merge two pdf files using python script¶

In [ ]:

```
#Merging two pdf
```

In [ ]:

```
!pip install PyPDF2
```

```
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.7/dist-packages (1.26.0)
```

In [ ]:

```
from PyPDF2 import PdfFileMerger
import os
path="/content/"
pdf_files=['Day13 Assignment.pdf','Day14 Assignment.pdf']
merger=PdfFileMerger()
```

```
for items in pdf_files:
  merger.append(path+items)
if not os.path.exists(path+'merged.pdf'):
  merger.write(path+'merged.pdf')
merger.close()
```

In [ ]:

```
a=open('/content/merged.pdf')
```

In [ ]:

```
a
```

Out[ ]:

```
<_io.TextIOWrapper name='/content/merged.pdf' mode='r' encoding='UTF-8'>
```

In [ ]:

```
3.Scrape a website and store the data into DB
```

In [ ]:

```
! pip install bs4
```

```
Requirement already satisfied: bs4 in /usr/local/lib/python3.7/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (
from bs4) (4.6.3)
```

In [ ]:

```
import urllib.request
```

In [ ]:

```
from bs4 import BeautifulSoup as bs
```

In [ ]:

```
import re
import pandas as pd
```

In [ ]:

```
page = urllib.request.urlopen("https://docs.python.org/3/library/random.html")
soup = bs(page)

#find all function names
names = soup.body.findAll('dt')
function_names = re.findall('id="random.\w+', str(names))
function_names = [item[4:] for item in function_names]

#find all function descriptions
description = soup.body.findAll('dd')
function_usage = []

for item in description:
  item = item.text
  item = item.replace('\n', ' ')
  function_usage.append(item)

print('list of function names:',function_names[:5])
print('\nfunction description:', function_usage[0])
print('\nnumber of items in function names:', len(function_names))
print('number of items in function description:', len(function_usage))
```

```
list of function names: ['random.seed', 'random.getstate', 'random.setstate', 'random.ran
dbytes', 'random.randrange']

function description: Initialize the random number generator. If a is omitted or None, th
```

```
e current system time is used.  If randomness sources are provided by the operating syste
m, they are used instead of the system time (see the os.urandom() function for details on
availability). If a is an int, it is used directly. With version 2 (the default), a str,
bytes, or bytearray object gets converted to an int and all of its bits are used. With ve
rsion 1 (provided for reproducing random sequences from older versions of Python), the al
gorithm for str and bytes generates a narrower range of seeds.  Changed in version 3.2: M
oved to the version 2 scheme which uses all of the bits in a string seed.   Deprecated si
nce version 3.9: In the future, the seed must be one of the following types: NoneType, in
t, float, str, bytes, or bytearray.

number of items in function names: 25
number of items in function description: 25
```

In [ ]:

```
print(len(function_names))
```

```
25
```

In [ ]:

```
print(len(function_usage))
```

```
25
```

In [ ]:

```
data=pd.DataFrame({'f_name':function_names,'f_usage':function_usage})
```

In [ ]:

```
data
```

Out[ ]:

	f_name	f_usage
0	random.seed	Initialize the random number generator. If a i...
1	random.getstate	Return an object capturing the current interna...
2	random.setstate	state should have been obtained from a previou...
3	random.randbytes	Generate n random bytes. This method should no...
4	random.randrange	Return a randomly selected element from range(...
5	random.randint	Return a random integer N such that a <= N <= ...
6	random.getrandbits	Returns a non-negative Python integer with k r...
7	random.choice	Return a random element from the non-empty seq...
8	random.choices	Return a k sized list of elements chosen from ...
9	random.shuffle	Shuffle the sequence x in place. The optional ...
10	random.sample	Return a k length list of unique elements chos...
11	random.random	Return the next random floating point number i...
12	random.uniform	Return a random floating point number N such t...
13	random.triangular	Return a random floating point number N such t...
14	random.betavariate	Beta distribution. Conditions on the paramete...
15	random.expovariate	Exponential distribution. lambd is 1.0 divide...
16	random.gammavariate	Gamma distribution. (Not the gamma function!)...
17	random.gauss	Gaussian distribution. mu is the mean, and si...
18	random.lognormvariate	Log normal distribution. If you take the natu...
19	random.normalvariate	Normal distribution. mu is the mean, and sigm...
20	random.vonmisesvariate	mu is the mean angle, expressed in radians bet...

	f_name	f_usage
21	random.paretovariate	Pareto distribution. alpha is the shape param...
22	random.weibullvariate	Weibull distribution. alpha is the scale para...
23	random.Random	Class that implements the default pseudo-rando...
24	random.SystemRandom	Class that uses the os.urandom() function for ...

In [ ]:

```
data.to_csv('my_file.csv')
```