01 Creating file in write mode

December 13, 2025

0.1 Checking the folders and create it

[2]:
```
import os

if not os.path.exists('data'):
        os.makedirs('data')


# another way to check it os.makedirs('data', exist_ok=True)
```

0.2 writing the content after checking the specific folders and files

[5]:
```
fp = open('data/example.txt', 'w')
fp.write('this is the some content1\n')
fp.write('this is the some content2\n')
fp.close()
```

[6]:
```
with open('data/example.txt', 'w') as fp:
        fp.write('this is some content3\n')
```

[1]:
```
lines = ['first line', 'second line', 'third line']
text = '\n'.join(lines)
with open('data/example1.txt', 'a') as fp:
        fp.write(text)
```

[3]:
```
with open('data/example1.txt', 'a') as fp:
        fp.write('\nhello how are you')
```

[4]:
```
lines = ['fourt line', 'fifth line', 'sixth line']
text = '\n'.join(lines)
with open('data/example1.txt', 'a') as fp:
        fp.write('\n')
        fp.write(text)
```

0.3 Reading and Evaluating the files

[6]:
```
with open('data/example1.txt', 'r') as fp:
        text = fp.read().splitlines()
```

[7]:
```
text
```

[7]:
```
['first line',
 'second line',
 'third line',
 'hello how are you',
 'fourt line',
 'fifth line',
 'sixth line']
```

[8]:
```
with open('data/expression.txt', 'w') as fp:
        fp.write('2+2')
        fp.write('\n')
        fp.write(str(2+2.45))
```

[9]:
```
with open('data/expression.txt', 'r') as fp:
        text = fp.read().splitlines()
        text = [eval(x) for x in text]
text
```

[9]:
```
[4, 4.45]
```

[11]:
```
l = list(range(1,30))
l
```

[11]:
```
[1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
```

```
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
29]
```

[12]:
```
with open('data/example.txt', 'w') as fp:
        fp.write(str(l))
```

[14]:
```
with open('data/example.txt', 'r') as fp:
        l1 = fp.read()
```

[15]:
```
l1
```

[15]:
```
'[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29]'
```

[16]:
```
type(l1)
```

[16]:
```
str
```

[17]:
```
l1= eval(l1)

l1[10:18]
```

[17]:
```
[11, 12, 13, 14, 15, 16, 17, 18]
```

02 reading CSV and TSV Files

December 13, 2025

0.1 Reading and Writing .CSV and .TSV Files with Pandas

[2]:
```
import pandas as pd
```

[4]:
```
df = pd.read_csv('data/example1.txt', header=None)
df
```

[4]:
```
                    0
0          first line
1         second line
2          third line
3  hello how are you
4          fourt line
5          fifth line
6          sixth line
```

0.2 Converting the column data in the list

[6]:
```
l = df[0].tolist()
l
```

[6]:
```
['first line',
 'second line',
 'third line',
 'hello how are you',
 'fourt line',
 'fifth line',
 'sixth line']
```

0.2.1 Reading CSV file from online

[8]:
```
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪jamesbond.csv'
df1 = pd.read_csv(url)
df1.head()
```

[8]:
```
                  Film  Year         Actor       Director  Box Office  \
0               Dr. No  1962  Sean Connery  Terence Young       448.8
```

```
1  From Russia with Love  1963  Sean Connery  Terence Young       543.8
2             Goldfinger  1964  Sean Connery   Guy Hamilton       820.4
3            Thunderball  1965  Sean Connery  Terence Young       848.1
4          Casino Royale  1967   David Niven     Ken Hughes       315.0

   Budget  Bond Actor Salary
0     7.0                0.6
1    12.6                1.6
2    18.6                3.2
3    41.9                4.7
4    85.0                NaN
```

[ ]:
```
df =pd.read_csv(url, nrows=10) # reading only the 10 rows
df
```

[ ]:
```
                             Film  Year           Actor       Director  \
0                         Dr. No  1962    Sean Connery  Terence Young
1             From Russia with Love  1963    Sean Connery  Terence Young
2                        Goldfinger  1964    Sean Connery   Guy Hamilton
3                      Thunderball  1965    Sean Connery  Terence Young
4                    Casino Royale  1967     David Niven     Ken Hughes
5                You Only Live Twice  1967    Sean Connery  Lewis Gilbert
6  On Her Majesty's Secret Service  1969  George Lazenby  Peter R. Hunt
7              Diamonds Are Forever  1971    Sean Connery   Guy Hamilton
8                  Live and Let Die  1973     Roger Moore   Guy Hamilton
9       The Man with the Golden Gun  1974     Roger Moore   Guy Hamilton

   Box Office  Budget  Bond Actor Salary
0       448.8     7.0                0.6
1       543.8    12.6                1.6
2       820.4    18.6                3.2
3       848.1    41.9                4.7
4       315.0    85.0                NaN
5       514.2    59.9                4.4
6       291.5    37.3                0.6
7       442.5    34.7                5.8
8       460.3    30.8                NaN
9       334.0    27.7                NaN
```

0.2.2 Reading TSV file online

[11]:
```
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪spam.tsv'

df = pd.read_csv(url, sep='\t')
df
```

[11]:

	label	message	length	punct
0	ham	Go until jurong point, crazy.. Available only …	111	9
1	ham	Ok lar… Joking wif u oni…	29	6
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…	155	6
3	ham	U dun say so early hor… U c already then say…	49	6
4	ham	Nah I don't think he goes to usf, he lives aro…	61	2
…	…	…	…	…
5567	spam	This is the 2nd time we have tried 2 contact u…	160	8
5568	ham	Will ü b going to esplanade fr home?	36	1
5569	ham	Pity, * was in mood for that. So…any other s…	57	7
5570	ham	The guy did some bitching but I acted like i'd…	125	1
5571	ham	Rofl. Its true to its name	26	1

[5572 rows x 4 columns]

[13]:
```
df = pd.read_csv(url, sep='\t', usecols= ['message', 'label'])
df
```

[13]:

	label	message
0	ham	Go until jurong point, crazy.. Available only …
1	ham	Ok lar… Joking wif u oni…
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…
3	ham	U dun say so early hor… U c already then say…
4	ham	Nah I don't think he goes to usf, he lives aro…
…	…	…
5567	spam	This is the 2nd time we have tried 2 contact u…
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So…any other s…
5570	ham	The guy did some bitching but I acted like i'd…
5571	ham	Rofl. Its true to its name

[5572 rows x 2 columns]

03 Read files in chunks

December 13, 2025

[2]:
```
import pandas as pd
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪IMDB-Dataset.csv'
df = pd.read_csv(url, chunksize=10000)
df
```

[2]:
```
<pandas.io.parsers.readers.TextFileReader at 0x2a457ec0bd0>
```

[4]:
```
for chunk in df:
        print(chunk.shape)
```

```
(10000, 2)
(10000, 2)
(10000, 2)
(10000, 2)
```

[5]:
```
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪IMDB-Dataset.csv'
df = pd.read_csv(url, chunksize=10000)
data = pd.DataFrame()
for chunk in df:
        data = pd.concat([data, chunk])
```

[6]:
```
data.head()
```

[6]:

	review	sentiment
0	One of the other reviewers has mentioned that …	positive
1	A wonderful little production. The…	positive
2	I thought this was a wonderful way to spend ti…	positive
3	Basically there's a family where a little boy …	negative
4	Petter Mattei's "Love in the Time of Money" is…	positive

[7]:
```
data.shape
```

[7]:
```
(50000, 2)
```

[8]:
```
data.to_csv('data/imdb_movie.csv')
```

[12]:
```
pd.read_csv('data/imdb_movie.csv').head()
```

[12]:

	Unnamed: 0	review	sentiment
0	0	One of the other reviewers has mentioned that …	positive
1	1	A wonderful little production. The…	positive
2	2	I thought this was a wonderful way to spend ti…	positive
3	3	Basically there's a family where a little boy …	negative
4	4	Petter Mattei's "Love in the Time of Money" is…	positive

[10]:
```
data.to_csv('data/imdb_movie.tsv', sep='\t')
```

[14]:
```
pd.read_csv('data/imdb_movie.tsv', sep='\t').head()
```

[14]:

	Unnamed: 0	review	sentiment
0	0	One of the other reviewers has mentioned that …	positive
1	1	A wonderful little production. The…	positive
2	2	I thought this was a wonderful way to spend ti…	positive
3	3	Basically there's a family where a little boy …	negative
4	4	Petter Mattei's "Love in the Time of Money" is…	positive

04 Read and write in Excel

December 13, 2025

[1]:
```
import pandas as pd
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪Data%20-%20Multiple%20Worksheets.xlsx'
df = pd.read_excel(url)
df
```

[1]:
```
  First Name Last Name           City Gender
0    Brandon     James          Miami      M
1       Sean   Hawkins         Denver      M
2       Judy       Day    Los Angeles      F
3     Ashley      Ruiz  San Francisco      F
4  Stephanie     Gomez       Portland      F
```

0.1 Checking that how many sheets it have

[2]:
```
df = pd.read_excel(url, sheet_name=None)
df
```

[2]:
```
{'Data 1':   First Name Last Name           City Gender
 0    Brandon     James          Miami      M
 1       Sean   Hawkins         Denver      M
 2       Judy       Day    Los Angeles      F
 3     Ashley      Ruiz  San Francisco      F
 4  Stephanie     Gomez       Portland      F,
 'Data 2':   First Name Last Name           City Gender
 0     Parker     Power        Raleigh      F
 1    Preston  Prescott   Philadelphia      F
 2    Ronaldo   Donaldo         Bangor      M
 3      Megan   Stiller  San Francisco      M
 4     Bustin    Jieber         Austin      F}
```

So it have 2 sheets in the excel, if we don't provide the sheet name then it is take first sheet by default

[3]:
```
df = pd.read_excel(url, sheet_name='Data 1')
df
```

[3]:
	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

[4]:
```
df = pd.read_excel(url, sheet_name='Data 2')
df
```

[4]:
	First Name	Last Name	City	Gender
0	Parker	Power	Raleigh	F
1	Preston	Prescott	Philadelphia	F
2	Ronaldo	Donaldo	Bangor	M
3	Megan	Stiller	San Francisco	M
4	Bustin	Jieber	Austin	F

0.2 Other ways to extract the sheets

[5]:
```
df = pd.read_excel(url, sheet_name=None)
```

[6]:
```
df.items()
```

[6]:
```
dict_items([('Data 1',   First Name Last Name           City Gender
0    Brandon     James          Miami      M
1       Sean   Hawkins         Denver      M
2       Judy       Day    Los Angeles      F
3     Ashley      Ruiz  San Francisco      F
4  Stephanie     Gomez       Portland      F), ('Data 2',   First Name Last Name
City Gender
0     Parker     Power        Raleigh      F
1    Preston  Prescott   Philadelphia      F
2    Ronaldo   Donaldo         Bangor      M
3      Megan   Stiller  San Francisco      M
4     Bustin    Jieber         Austin      F)])
```

[7]:
```
df1 = df['Data 1']
df2 = df['Data 2']
```

[8]:
```
df1
```

[8]:
	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

[9]:
```
df2
```

[9]:

	First Name	Last Name	City	Gender
0	Parker	Power	Raleigh	F
1	Preston	Prescott	Philadelphia	F
2	Ronaldo	Donaldo	Bangor	M
3	Megan	Stiller	San Francisco	M
4	Bustin	Jieber	Austin	F

0.3 saving the excel data

[10]:
```
df = pd.read_excel(url, sheet_name=None)
```

[13]:
```
df1 = df['Data 1']
df1.to_excel('data/df1.xlsx', index = None)
```

0.4 writing the multiple sheets

[14]:
```
with pd.ExcelWriter('data/multiple_sheet.xlsx') as fp:
        df1.to_excel(fp, sheet_name ='data 1')
        df2.to_excel(fp, sheet_name ='data 2')
```

05 Data with json

December 13, 2025

[4]:
```
import pandas as pd
url = 'https://github.com/laxmimerit/All-CSV-ML-DATA-Files-Download/raw/master/
 ↪Data%20-%20Multiple%20Worksheets.xlsx'
df = pd.read_excel(url, sheet_name = 'Data 1')
df
```

[4]:

	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

[5]:
```
import json
```

[6]:
```
df.to_json('data/output.json', orient = 'records')
```

[7]:
```
df.to_json('data/output1.json', orient = 'records', lines = True)
```

[11]:
```
d1 = json.load(open('data/output.json'))
pd.DataFrame(d1)
```

[11]:

	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

[13]:
```
d1 = pd.read_json('data/output.json')
d1
```

[13]:

	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

[14]:
```
df2 = pd.read_json('data/output1.json', lines=True)
df2
```

[14]:

	First Name	Last Name	City	Gender
0	Brandon	James	Miami	M
1	Sean	Hawkins	Denver	M
2	Judy	Day	Los Angeles	F
3	Ashley	Ruiz	San Francisco	F
4	Stephanie	Gomez	Portland	F

0.0.1 Working with the nested json data

[15]:
```
nested_data = {
        'passenger': {
                'name': 'bob',
                'age': 23,
                'ticket':{
                        'number':'23sdlfj',
                        'price':123
                }
        }
}
```

0.0.2 How you read or show the nested json in the DataFrame

[16]:
```
pd.json_normalize(nested_data)
```

[16]:

	passenger.name	passenger.age	passenger.ticket.number	passenger.ticket.price
0	bob	23	23sdlfj	123

06 Extract the data from PDF

December 13, 2025

[3]:
```
import warnings
warnings.filterwarnings('ignore')
```

[4]:
```
!pip install PyPDF2==3.0.1
```

```
Requirement already satisfied: PyPDF2==3.0.1 in
c:\users\uditya\.conda\envs\udemy\lib\site-packages (3.0.1)
```

[5]:
```
import PyPDF2
```

[8]:
```
with open('data/ML.pdf', 'rb') as fp:
        reader = PyPDF2.PdfReader(fp)
        text = reader.pages[0].extract_text()
```

[9]:
```
print(text)
```

```
Introduction to Machine learning In Detailed
July 20, 2025
1 Introduction of Machine Learning
In this module we are going to learn to we can use the machine in a effective
ways and for the
better utilization of any machine and how we can use the different techniques to
teach a machine
so that a machine can replicate the human behavior and work effectively . so
there are some basics
terms, which is we have to learn it.
• Machine Learning is the ability of a machine to learn from and replicate the
human behavior.
• It allows programs to learn automatically and make computers more intelligent.
1.0.1 Key Differences in Artificial Intelligence(AI), Machine Learning(ML) and
Deep
Learning(DL)
• ML referes the algorithms that learn and perform based on the data exposed to
it. Machine
Learning is used to classify the profile of the bot’s competitor, the human, as:
• DL referes to layers of neural networks built with Machine Learning
algorithms.
• AI makes decisions with the help of a multi-layered neural network that uses
DL AI leverages
```

different techniques, including ML and DL.
1.0.2 Machine Learning Algorithms
1. Algorithms are a set of instructions used to solve problems.
2. ML algorithms help to:
• Predect.
• Classify .
• Improve performance of any software application.
1.0.3 Uses of Machine Learning
1. The predictions by ML are used to:
• Shape policies.
• Make weather forecast.
• Determine tra ic rules, CO2, emission, forest degradation, visibility in the sky , wildfire,
heatwaves, and even the size of chimneys.
1.0.4 Data and ML
1. ML is dependent on data.
1

0.1 Reading all the pages

[13]:
```
pages = []
with open('data/ML.pdf', 'rb') as fp:
        reader = PyPDF2.PdfReader(fp)
        for i in range(len(reader.pages)):
                page = reader.pages[i]
                text = page.extract_text()
                pages.append(text)
```

[14]:
```
pages
```

[14]: ['Introduction to Machine learning In Detailed\nJuly 20, 2025\n1 Introduction of Machine Learning\nIn this module we are going to learn to we can use the machine in a effective ways and for the\nbetter utilization of any machine and how we can use the different techniques to teach a machine\nso that a machine can replicate the human behavior and work effectively . so there are some basics\nterms, which is we have to learn it.\n• Machine Learning is the ability of a machine to learn from and replicate the human behavior.\n• It allows programs to learn automatically and make computers more intelligent.\n1.0.1 Key Differences in Artificial Intelligence(AI), Machine Learning(ML) and Deep\nLearning(DL)\n• ML referes the algorithms that learn and perform based on the data exposed to it. Machine\nLearning is used to classify the profile of the bot's competitor, the human, as:\n• DL referes to layers of neural networks built with Machine Learning algorithms.\n• AI makes decisions with the help of a multi-layered neural network that uses DL AI leverages\ndifferent techniques, including ML and DL.\n1.0.2 Machine Learning Algorithms\n1. Algorithms are a set of instructions used to solve problems.\n2. ML algorithms help to:\n• Predect.\n• Classify .\n• Improve performance of any software

application.\n1.0.3 Uses of Machine Learning\n1. The predictions by ML are used to:\n• Shape policies.\n• Make weather forecast.\n• Determine tra ic rules, CO2, emission, forest degradation, visibility in the sky , wildfire,\nheatwaves, and even the size of chimneys.\n1.0.4 Data and ML\n1. ML is dependent on data.\n1',
 '2. performance of algorithms is evaluated based on the quality of the input data.\n1.0.5 Types of Machine Learning\nDepending on wheather the algorithm can self-train and predict a condtion or needs to identify\npatterns to derive outcomes.\n1.0.6 1. Supervised Learning\nIn supervised learning, the algorithm learns from labeled training data, meaning each input is paired\nwith the correct output. The model makes predictions based on this data and is corrected when\nwrong. It is used for: - Classification : Predicting categorical outcomes (e.g., spam vs. non-spam\nemails). - Regression : Predicting continuous outcomes (e.g., stock prices, temperature).\nExamples : - Linear Regression - Decision T rees - Support V ector Machines (SVM)\n1.0.7 2. Unsupervised Learning\nIn unsupervised learning, the model learns from unlabeled data and tries to uncover hidden patterns\nor structures. The data does not have pre-defined outputs.\n• Clustering : Grouping similar data points (e.g., customer segmentation).\n• Dimensionality Reduction : Reducing the number of input features (e.g., Principal Com-\nponent Analysis).\nExamples : - K-means clustering - Hierarchical clustering - Autoencoders\n1.0.8 3. Semi-Supervised Learning\nSemi-supervised learning uses a combination of labeled and unlabeled data. It is useful when\nlabeling a large dataset is time-consuming or expensive. The model first learns from a small set of\nlabeled data and then extends this learning to a larger set of unlabeled data.\nExample : - Self-training algorithms\n1.0.9 4. Reinforcement Learning\nIn reinforcement learning, an agent learns to make decisions by interacting with an environment\nand receiving feedback in the form of rewards or penalties. It learns to take actions that maximize\ncumulative rewards over time.\nExamples : - Q-learning - Deep Q-Networks (DQN) - Policy gradient methods\n1.0.10 5. Self-Supervised Learning\nSelf-supervised learning is a type of unsupervised learning where the data itself provides the labels.\nThe model learns from part of the data to predict another part (e.g., predicting the next word in\na sentence).\nExamples : - Generative models like GPT, BER T\n2',
 '1.0.11 6. T ransfer Learning\nT ransfer learning allows a model trained on one task to be reused on a new but related task. This\napproach is common when there's limited data for the new task but abundant data for the original\ntask.\nExample : - Pre-trained models like ResNet, BER T, and GPT used in different domains.\nEach type of machine learning is used for different problems, and the choice depends on the specific\ntask and available data.\n1.0.12 Machine Learning Pipeline\n1. A machine learning(ML) pipeline is a series of sequential steps used to codify and automate\nML workflows to produce ML models.\n2. ML pipeline is an end-to-end construction that includes and orchestrates:\n• Data extraction.\n• Raw data input.\n• Preprocessing.\n• Model parameters.\n• Model training.\n• Deployment.\n• Predicting outputs.\n3. The term pipeline implies a one-way flow of data but ML pipelines are cyclical and iterative.\n4. Every step in the sequence is repeated until a successful algorithm is achieved.\n1.0.13 Machine Learning and Operations Professionals

```
(MLOps)\nMLOps is a set of practices that combines: - ML - DevOps - Data
engineering\nEnsures reliable and e icient deployment and maintenance of ML
models in production systems.\nMLOps aims to: - Improve communication and
collaboration between MLOps. - Shorten and\nmanage complete development life
cycle. - Ensure continuous delivery of high-quality predictive\nservices.\n3
phases of MLOps\n• Design:\n- Undersanding the business and data, and then
designs the ML-powered software\n3',
 '• Model development:\n- Applicability of ML for the problem is verified by
implementing proof of concept(POC)\n• Operations.\n- Delevers the developed ML
model in production\nAll three MLOps phases are interconnected and influence one
another.\nThe tools used for MLOps are\n• Kubeflowl\n• MLFlow\n• Data version
control(DVC)\n• Pachyderm\n• Metaflow\n• Kedro\n• Seldon Core\n• Flyte\n1.0.14
CI/CD Pipeline Automation\nA continuous integration(CI) or continuous
delivery(CD) system is used to: - T est and deploy\nnew pipeline implementations
automatically - A CI/CD pipeline automation helps with dynamic\nchanges in the
data and business environment\nAutomated Machine Learning(AutoML)\n• Combines
the best practices in automation and machine learning\n• It enables
organizations to build and deploy ML models using:\n- Predefined templates\n- F
rameworks\n- Processes to speed up time to completion\n- Enhances functionality
of ML models\nThe tools used for AutoML are:\n• Run:AI\n• Auto-Keras\n•
H2OAutoML\n• SMAC\n• AUTOWEKA\n• AUTO-SKLEARN\n• AUTO-PYTORCH\n• ROBO\n2 Python
Libraries Used in Machine Learning\n• Numpy\n• Pandas\n• Matplotlib\n4',
 '• T ensorFlow\n• Aesara based on Theano\n• SciPy\n• Scikit-learn\n• Keras\n•
PyT orch\nNumpy and pandas helps to manage preparation, loading, and
manipulation of data.\nT ensorFlow and Aesara is used for fast numerical
computing.\nMatplotlib is used to plot data.\nSciPy helps solve mathematical
equations and algorithms.\nScikit-learn provides e icient versions of common
algorithms to develop ML models.\nkeras makes the implementation of neural
networks easy .\nPyT orch specializes in deep learning applications and
accelerates the path from prototyping to\ndeployment.\n5']
```

[15]:
```
text = '\n'.join(pages)
with open('data/ml.txt', 'w') as fp:
        fp.write(text)
```

0.2 reading only selected pages

[16]:
```
pages = []
with open('data/ML.pdf', 'rb') as fp:
        reader = PyPDF2.PdfReader(fp)
        for i in range(len(reader.pages)):
                if i+1 in [1,2]:
                        page = reader.pages[i]
                        text = page.extract_text()
                        pages.append(text)

text = '\n'.join(pages)
```

```
with open('data/ml1.txt', 'w') as fp:
        fp.write(text)
```

07 Record the audio and convert to text

December 13, 2025

[1]:
```
import pyaudio
```

[2]:
```
audio = pyaudio.PyAudio()
audio.get_device_count()
```

[2]:
```
33
```

[4]:
```
for i in range(audio.get_device_count()):
        info = audio.get_device_info_by_index(i)
        # print(info)
```

[5]:
```
import speech_recognition as sr
```

[6]:
```
recognizer = sr.Recognizer()
with sr.Microphone() as source:
        print('Say Something...')
        audio = recognizer.listen(source)
```

```
Say Something…
```

[8]:
```
text = recognizer.recognize_google(audio)
```

[9]:
```
print(text)
```

```
hello how are you what are you doing
```

0.0.1 Saving in the local system

[12]:
```
import os

os.makedirs('audio', exist_ok=True)
```

[13]:
```
with open('audio/record1.wav', 'wb') as fp:
        fp.write(audio.get_wav_data())
```

[14]:
```
recognizer = sr.Recognizer()
with sr.Microphone() as source:
        print('Say Something...')
        audio1 = recognizer.listen(source)
```

Say Something…

[15]:
```
text = recognizer.recognize_google(audio1)
```

[16]:
```
print(text)
```

```
hello so we are going to record this particular Syntax so that we can use this
microphone properly
```

[19]:
```
with open('audio/record2.wav', 'wb') as fp:
        fp.write(audio1.get_wav_data())
```

0.1 Recording for some time limit

[21]:
```
with sr.Microphone() as source:
        print('Recoding for 10 sec...')
        audio2 = recognizer.record(source, duration=10)
        text = recognizer.recognize_google(audio2)
```

```
Recoding for 10 sec…
```

[22]:
```
print(text)
```

```
Sudesh particular snippet is responsible for giving me time span of 10 second
after 10 second it will automatically close it
```

[25]:
```
with open('audio/record3.wav', 'wb') as fp:
        fp.write(audio2.get_wav_data())
```

0.2 Read the audio and convert in text data

[27]:
```
with sr.AudioFile('audio/record1.wav') as fp:
        audio_data = recognizer.record(fp)
        text = recognizer.recognize_google(audio_data)

print(text)
```

```
hello how are you what are you doing
```

0.3 Read the audio for specific time duration

[ ]:
```
with sr.AudioFile('audio/record1.wav') as fp:
        audio_data = recognizer.record(fp, duration=3)
        text = recognizer.recognize_google(audio_data)

print(text)
```