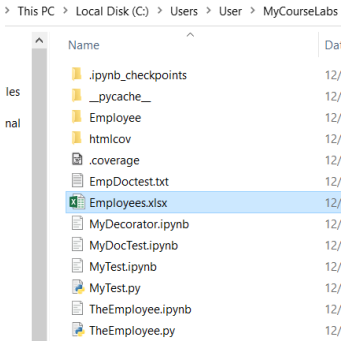
Reading from Excel File

1- Create an Excel file with the following content:

	A	B	C
1	**EmployeeID**	**Name**	**Department**
2	100	Khalil	IT
3	101	Jameela	Sales
4	102	Ahmed	Marketing

2- Save it in the same directory of your project



3- Before you start reading excel files using python code, you might need to download the following module to enable you to work with excel files:

```
pip install xlrd
```

4- Then open a new Jupyter notebook and add the following code:

```
# We will read data from an excel file using Pandas
import pandas as pd

# Read the first sheet from the file [by default]
# and store the content in a data frame
df = pd.read_excel('Employees.xlsx')

# Display the data frame content
df
```

	EmployeeID	Name	Department
0	100	Khalil	IT
1	101	Jameela	Sales
2	102	Ahmed	Marketing

5- Displaying one entry:

```
# We will read data from an excel file using Pandas
import pandas as pd

# Read the first sheet from the file [by default]
# and store the content in a data frame
df = pd.read_excel('Employees.xlsx')

# Display a single entry from the data frame content
df.loc[0]
```

```
EmployeeID       100
Name          Khalil
Department        IT
Name: 0, dtype: object
```

6- Displaying specific value from an entry

```
# We will read data from an excel file using Pandas
import pandas as pd

# Read the first sheet from the file [by default]
# and store the content in a data frame
df = pd.read_excel('Employees.xlsx')

# Display the name value of a single entry from the data frame content
df.loc[0,'Name']
```

```
'Khalil'
```

7- Looping through the entries:

```
# We will read data from an excel file using Pandas
import pandas as pd

# Read the first sheet from the file [by default]
# and store the content in a data frame
df = pd.read_excel('Employees.xlsx')

# Display the name value for each entry from the data frame content
# using a loop
for count in df.index:
    print("Employee No[" + str(count+1) + "]:")
    print(df.loc[count,'Name'])
```

```
Employee No[1]:
Khalil
Employee No[2]:
Jameela
Employee No[3]:
Ahmed
```

8- Writing to an existing Excel file:

```
# We will write some data to an existing Excel file
# using data frames
import pandas as pd
import openpyxl

# Define the entries of new employees as tuples
empIDs = [103,104]
empNames = ["Nadir","Salwa"]
empDepts = ["Management","Marketing"]
# Store the columns names in a tuple as well
myCols = ["EmployeeID","Name","Department"]
# When creating the data frame we have to make sure that the number of columns
# is equal to the number of entries. Here we have 3 entries which are
# IDs, names, and departments values, and we have 3 column header values
df = pd.DataFrame(list(zip(empIDs,empNames,empDepts)), columns = myCols)
# Here we define the path to the Excel file
path = 'Employees.xlsx'

# Here we create our writer, give it the path to the file, specify the engine
# used to deal with the Excel file, and choosing the mode [a --> append]
with pd.ExcelWriter(path, engine="openpyxl", mode="a") as writer:
    # Here we start writing to the file [indexing is stopped so that we
    # don't store the index with each entry]
    df.to_excel(writer, sheet_name='Employees', index = False)
```

Using regular expressions to extract part of data from a file

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # Using regular expression to search for a word in each line [Case-sensitive]
    if re.search(r"Khobeib", line):
        print(line, end="")
```

```
Khobeib: Good morning
Khobeib: I am fine, thanks for asking.
Khobeib: Yeah, it is available on our main dashboard.
```

If we want to conduct a search without focusing on the case-sensitivity:

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # Using regular expression to search for a word in each line [Case-insensitive]
    if re.search(r"kHobEib", line, re.IGNORECASE):
        print(line, end="")
```

```
Khobeib: Good morning
Khobeib: I am fine, thanks for asking.
Khobeib: Yeah, it is available on our main dashboard.
```

Using the “\s” to look for specific sequence of characters

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # Using regular expression to the word 'than'. We notice that we are getting
    # another line where 'than' is part of 'thanks'
    if re.search(r"than", line, re.IGNORECASE):
        print(line, end="")
```

```
Khobeib: I am fine, thanks for asking.
John: Wow, that is even better than printing it and distribute it among meeting members.
```

The solution is using “\s”

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # We can use '\s' to denote a space before and after 'than'
    # which will result in getting only the 'than' word
    if re.search(r"\sthan\s", line, re.IGNORECASE):
        print(line, end="")
```

```
John: Wow, that is even better than printing it and distribute it among meeting members.
```

We can search for different words that share some similar characters

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # We can use a list of characters as a searching criteria
    if re.search(r"fin[eai]", line, re.IGNORECASE):
        print(line, end="")
```

```
Khobeib: I am fine, thanks for asking.
Khobeib: Ohh, finally someone acknowledges my efforts.
John: Of course, you have done a great job finishing that report.
```

Using regular expressions to substitute characters

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # We want all the responses of the employee Khobeib
    if re.search(r"Khobeib", line, re.IGNORECASE):
        # When printing the responese, we want to keep employee
        # identity anonymous. We use the substitue function to do that
        newLine = re.sub(r"Khobeib","Employee",line)
        # We print the line with substituted data instead of the original line
        print(newLine, end="")
```

```
Employee: Good morning
Employee: I am fine, thanks for asking.
Employee: Yeah, it is available on our main dashboard.
Employee: Ohh, finally someone acknowledges my efforts.
```

Extracting employee contact number

```
# Adding a regular expression module
import re
# Reading the lines from a text file and storing it in a variable
lines = open("Transcript.txt", "r").readlines()
# Looping through the lines
for line in lines:
    # Extracting the employee contact number
    empPhone = re.findall('[0-9]+',line)
print("The employee contact number is: " + str(empPhone))
```

```
The employee contact number is: ['077755321']
```

Finished