



Just IT

 B2Wgroup

Apprenticeships | Training | Recruitment

Data Technician

Name:

Course Date:

Table of contents

Day 2: Task 1	3
Day 3: Task 1	4
Exercise 1: Loading and Exploring the Data.....	4
Exercise 2: Indexing and Slicing.....	4
Exercise 3: Data Manipulation.....	6
Exercise 4: Aggregation and Grouping.....	8
Exercise 5: Advanced Operations.....	9
Exercise 6: Exporting Data	11
Exercise 7: If finished early try visualising the results	12
Day 4: Task 1	12
Day 4: Task 2	15
Course Notes.....	20
Additional Information.....	21



Day 2: Task 1

It is a common software development interview question to create the below with a certain programming language. Create the below using Python syntax, test it and past the completed syntax and output below.

FizzBuzz:

Go through the integers from 1 to 100.

If a number is divisible by 3, print "fizz."

If a number is divisible by 5, print "buzz."

If a number is both divisible by 3 and by 5, print "fizzbuzz."

Otherwise, print just the number.

**Paste your completed
work to the right**



Day 3: Task 1

Download the 'student.csv', complete the below exercises as a group and paste your input and output. Although this is a group activity, everyone should have the below answered so it supports your portfolio:

Exercise 1: Loading and Exploring the Data

1. Question: "Write the code to read a CSV file into a Pandas DataFrame."
2. Question: "Write the code to display the first 5 rows of the DataFrame."
3. Question: "Write the code to get the information about the DataFrame."
4. Question: "Write the code to get summary statistics for the DataFrame."

```
import pandas as pd
df = pd.read_csv("student.csv")
df.head()
df.info()
df.describe()
```

Exercise 2: Indexing and Slicing

1. Question: "Write the code to select the 'name' column."
2. Question: "Write the code to select the 'name' and 'mark' columns."
3. Question: "Write the code to select the first 3 rows."
4. Question: "Write the code to select all rows where the 'class' is 'Four'."



.0]
0s



```
df["name"]
```



	name
0	John Deo
1	Max Ruin
2	Arnold
3	Krish Star
4	John Mike
5	Alex John
6	My John Rob
7	Asruid

s





```
df[["name", "mark"]]
```





	name	mark
0	John Deo	75
1	Max Ruin	85
2	Arnold	55
3	Krish Star	60





5] 0s  `df.head(3)`




	id	name	class	mark	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male

 `df[df["class"] == "four"]`




	id	name	class	mark	gender
--	----	------	-------	------	--------




Exercise 3: Data Manipulation

1. Question: "Write the code to add a new column 'passed' that indicates whether the student passed (mark >= 60)."
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."



```
df["passed"] = df["mark"] >= 60
print('1. added "passed" column:')
print(df[["name", "mark", "passed"]].head())
```

 1. added "passed" column:

	name	mark	passed
0	John Deo	75	True
1	Max Ruin	85	True
2	Arnold	55	False
3	Krish Star	60	True
4	John Mike	60	True

```
print(f"columns: {df.columns.tolist()}")
```

1. added "passed" column:

	name	mark	passed
0	John Deo	75	True
1	Max Ruin	85	True
2	Arnold	55	False
3	Krish Star	60	True
4	John Mike	60	True

. renamed 'mark' to 'score'

columns: ['id', 'name', 'class', 'score', 'gender', 'passed']

/n3. dropped 'Passed' column

columns: ['id', 'name', 'class', 'score', 'gender']

```
df = df.rename(columns={"mark": "score"})
print("\n2. renamed 'mark' to 'score'")
print(f"columns: {df.columns.tolist()}")
```

2.

renamed 'mark' to 'score'

columns: ['id', 'name', 'class', 'score', 'gender', 'passed']

columns: ['id', 'name', 'class', 'score', 'gender']

1. mean score by class:

class

Eight 79.000000

Fifth 78.000000

Five 80.000000

Four 68.750000

Nine 41.500000

Seven 77.600000

Six 82.571429

Three 73.666667

Name: score, dtype: float64

```
mean score by class:
class
Eight      79.000000
Fifth      78.000000
Five       80.000000
Four       68.750000
Nine       41.500000
Seven      77.600000
Six        82.571429
Three      73.666667
Name: score, dtype: float64
```

mean score as DataFrame:

	class	score
0	Eight	79.000000
1	Fifth	78.000000
2	Five	80.000000
3	Four	68.750000
4	Nine	41.500000
5	Seven	77.600000
6	Six	82.571429
7	Three	73.666667

Exercise 4: Aggregation and Grouping

1. Question: "Write the code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group."
2. Question: "Write the code to count the number of students in each class."
3. Question: "Write the code to calculate the average mark for each gender."


```

▶ class_means = df.groupby("class") ["score"].mean()
print("average mark by class")
print(class_means)
print("\n" + "="*50 + "\n")

```

```

⇒ average mark by class
class
Eight      79.000000
Fifth      78.000000
Five       80.000000
Four       68.750000
Nine       41.500000
Seven      77.600000
Six        82.571429
Three      73.666667
Name: score, dtype: float64

```

```

=====

average mark by gender
gender
female      77.312500
male        71.588235
Name: score, dtype: float64

```

Exercise 5: Advanced Operations

1. Question: "Write the code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values."
2. Question: "Write the code to create a new column 'grade' where marks ≥ 85 are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'."
3. Question: "Write the code to sort the DataFrame by 'mark' in descending order."

⇒ pivot table – average score by class and gender:

gender	female	male
class		
Eight	NaN	79.0
Fifth	NaN	78.0
Five	NaN	80.0
Four	63.8	77.0
Nine	65.0	18.0
Seven	81.4	73.8
Six	89.2	54.0
Three	NaN	70.0

=====

⇒ added 'grade' column:

	name	score	grade
0	John Deo	75	B
1	Max Ruin	85	A
2	Arnold	55	D
3	Krish Star	60	C
4	John Mike	60	C
5	Alex John	55	D
6	My John Rob	78	B
7	Asruid	85	A
8	Tes Qry	78	B
9	Big John	55	D

Grade distribution:

grade

A 14

B 9

C 5

D 7

Name: count, dtype: int64

=====

⇒ DataFrame sorted by score (descending):

	name	score
32	Kenn Rein	96
11	Recky	94
31	Binn Rott	90
10	Ronald	89
30	Marry Toeey	88
34	Rows Noump	88
24	Giff Tow	88
14	Tade Row	88
15	Gimmy	88
12	Kty	88

Exercise 6: Exporting Data

1. Question: "Write the code to save the DataFrame with the new 'grade' column to a new CSV file."

```
j] df.to_csv("personal_data.csv", index=False)
```

Exercise 7: If finished early try visualising the results

Day 4: Task 1

Using the 'GDP (nominal) per Capita.csv' which can be downloaded from the shared Folder, complete the below exercises and paste your input and output. Work individually, but we will work and support each other in the room.

- Read and save the 'GDP (nominal) per Capita' data to a data frame called "df" in Jupyter notebook
- Print the first 10 rows
- Print the last 5 rows
- Print 'Country/Territory' and 'UN_Region' columns

▼ Pandas, Matplotlib and Seaborn

```
[13] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[Dataset Source](#)



```
[3] #print the first 10 rows
print("print first 10 rows:")
print(df.head(10))
```

```
print first 10 rows:
Country/Territory UN_Region IMF_Estimate IMF_Year WorldBank_Estimate \
1 Monaco Europe 0 0 234316
2 Liechtenstein Europe 0 0 157755
3 Luxembourg Europe 132372 2023 133590
4 Ireland Europe 114581 2023 100172
5 Bermuda Americas 0 0 114090
6 Norway Europe 101103 2023 89154
7 Switzerland Europe 98767 2023 91992
8 Singapore Asia 91100 2023 72794
9 Isle of Man Europe 0 0 87158
10 Cayman Islands Americas 0 0 86569

WorldBank_Year UN_Estimate UN_Year
1 2021 234317 2021
2 2020 169260 2021
3 2021 133745 2021
4 2021 101109 2021
5 2021 112653 2021
6 2021 89242 2021
7 2021 93525 2021
8 2021 66822 2021
9 2019 0 0
10 2021 85250 2021
```

```
# print last five rows
print("print last five rows:")
print(df.tail())
```

```
print last five rows:
Country/Territory UN_Region IMF_Estimate IMF_Year WorldBank_Estimate \
219 Malawi Africa 496 2023 635
220 South Sudan Africa 467 2023 1072
221 Sierra Leone Africa 415 2023 480
222 Afghanistan Asia 611 2020 369
223 Burundi Africa 249 2023 222

WorldBank_Year UN_Estimate UN_Year
219 2021 613 2021
220 2015 400 2021
221 2021 505 2021
222 2021 373 2021
223 2021 311 2021
```

```
print("\ncountry/territory UN_Region columns:")
```

220	2015	400	2021
221	2021	505	2021
222	2021	373	2021
223	2021	311	2021

```
2] print("\ncountry/territory, UN_Refion columns:")
0s print(df[["Country/Territory", "UN_Region"]])
```

```
country/territory, UN_Refion columns:
Country/Territory UN_Region
1 Monaco Europe
2 Liechtenstein Europe
3 Luxembourg Europe
4 Ireland Europe
5 Bermuda Americas
.. ...
219 Malawi Africa
220 South Sudan Africa
221 Sierra Leone Africa
222 Afghanistan Asia
223 Burundi Africa

[223 rows x 2 columns]
```

Start coding or generate with AI

List all column names in the DataFrame Print 'Country/Territory' and 'UN_Region' columns Exp

What can I help you build?

Which country has highest Worlbank Estimate?

```
2] hihest_WB_estimate = df[df["WorldBank_Estimate"] == df["WorldBank_Estimate"].max()]
0s print("Country with the highest World Bank estimate:")
print(hihest_WB_estimate[["Country/Territory", "WorldBank_Estimate"]])
```

```
Country with the highest World Bank estimate:
Country/Territory WorldBank_Estimate
1 Monaco 234316
```

Start coding or generate with AI.

List all column names in the DataFrame Print 'Country/Territory' and 'UN_Region' columns Exp



Which country has highest IMF Estimate?

+ Code

+ Text

↑ ↓ ↺ 🗑 ⋮

▶

highest_country_IMF_estimate = df[df["IMF_Estimate"] == df["IMF_Estimate"].max()]
print("Country with the highest IMF estimate:")
print(highest_country_IMF_estimate[["Country/Territory", "IMF_Estimate"]])

↗

Country with the highest IMF estimate:
Country/Territory IMF_Estimate
3 Luxembourg 132372


Start coding or [generate](#) with AI.

Day 4: Task 2

Back with 'GDP (nominal) per Capita'. As a group, import and work your way through the Day_4_Python_Activity.ipynb notebook which can be found on the shared Folder. There are questions to answer, but also opportunities to have fun with the data – paste your input and output below.

Once complete, and again as a group, work with some more data and have some fun – there is no set agenda for this section, other than to embed the skills developed this week. Paste your input and output below and upon return we'll discuss progress made.

[Additional data found here.](#)

 Data Technician | Workbook | v1.

Page 15 of 21

all

Start coding or generate with AI.

[13]
✓ Os

number of countries per region

df.UN_Region.value_counts()

↗

UN_Region	count
Africa	55
Asia	51
Americas	48
Europe	48
Oceania	20
World	1

dtype: int64

Start coding or generate with AI.

Start coding or generate with AI.

List all column names in the DataFrame

Print 'Country/Territory' and 'UN_Region' columns

Exp

↗ number of European countries: 48

below average GDP per capita in Europe: £34446.75

Tools Help

all

Start coding or generate with AI.

Global average GDP per capita: £15351.63

Number of countries with below GDP than the global average: 159
number of countries above global average: 64

Countries below global average GDP (IMF estimates):

	Country/Territory	IMF_Estimate
1	Monaco	0
2	Liechtenstein	0
5	Bermuda	0
9	Isle of Man	0
10	Cayman Islands	0
...
91	China	13721
90	Russia	14403
89	Costa Rica	14733
88	Palau	14804
87	Bulgaria	14893

[159 rows x 2 columns]

Start coding or generate with AI.

Start coding or generate with AI.

List all column names in the DataFrame

Print 'Country/Territory' and 'UN_Region' columns

Exp



colab.research.google.com/drive/1gzUF_u7HiwJlq8ExE28wLKTAKFIOWi#scrollTo=b1ee3245

Day_4_Python_Activity 1 and 2 (1).ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- sample_data
- GDP (nominal) per Capita.csv

IMF estimate 0 values

```
zero_count = (df["IMF_Estimate"] == 0).sum()
print(f"Number of countries with zero IMF estimate: {zero_count}")
```

Number of countries with zero IMF estimate: 26

Which country has highest UN Estimate?

List all column names in the DataFrame Print 'Country/Territory' and 'UN_Estimate' columns Exp

Start coding or generate with AI.

What can I help you build?

68.26 GB available

Variables Terminal

19:29 Python 3

UK GDP per capita: £46371.00

Number of European countries with higher GDP than UK: 13

European countries with higher GDP per capita than UK:

Country/Territory	IMF_Estimate
Luxembourg	132372
Ireland	114581
Norway	101103
Switzerland	98767
Iceland	75180
Denmark	68827
Netherlands	61098
Austria	56802
Sweden	55395
Finland	54351
Belgium	53377
San Marino	52949
Germany	51383

+ Code + Text

Start coding or generate with AI.

List all column names in the DataFrame Print 'Country/Territory' and 'UN_Estimate' columns Exp

Which country has highest UN Estimate?

```
highest_UN_estimate = df[df["UN_Estimate"] == df["UN_Estimate"].max()]
print("Country with the highest UN estimate:")
print(highest_UN_estimate[["Country/Territory", "UN_Estimate"]])
```

Country with the highest UN estimate:

Country/Territory	UN_Estimate
Monaco	234317

Start coding or generate with AI.

Which country has highest Worldbank Estimate?



```
# replace 0 with null values
```

```
df = pd.read_csv("GDP (nominal) per Capita.csv", encoding='unicode_escape', index_col=0)

df = df.replace(0, np.nan)

print("zero replaced with NaN. Summary:")
print(df.isnull().sum())
```

```
zero replaced with NaN. Summary:
Country/Territory    0
UN_Region            0
IMF_Estimate         26
IMF_Year            26
WorldBank_Estimate   7
WorldBank_Year       7
UN_Estimate          9
UN_Year              0
dtype: int64
```

```
# Calculate the average of 'Worldbank_Estimate' and 'UN_Estimate' columns
```

```
wb_avg = df["WorldBank_Estimate"].mean()
print(f"Average Worldbank Estimate: {wb_avg}")
```

```
un_avg = df["UN_Estimate"].mean()
print(f"Average UN_Estimate: {un_avg}")
```

```
Average Worldbank Estimate: 19540.805555555555
Average UN_Estimate: 18514.528037383177
```

```
# Fill the null values in 'imf' column with the calculated average
```

```
df["IMF_Estimate"] = df["IMF_Estimate"].fillna(df["IMF_Estimate"].mean())
print("\n== AFTER FILLING NULL VALUES ==")
print(df.isnull().sum())
```

```
== AFTER FILLING NULL VALUES ==
Country/Territory    0
UN_Region            0
IMF_Estimate         0
IMF_Year            26
WorldBank_Estimate   7
WorldBank_Year       7
UN_Estimate          9
UN_Year              0
dtype: int64
```

```

dataframe with temporay column added
Country/Territory UN_Region IMF_Estimate IMF_Year WorldBank_Estimate \
1 Monaco Europe 17377.736041 NaN 234316.0
2 Liechtenstein Europe 17377.736041 NaN 157755.0
3 Luxembourg Europe 132372.000000 2023.0 133590.0
4 Ireland Europe 114581.000000 2023.0 100172.0
5 Bermuda Americas 17377.736041 NaN 114090.0

WorldBank_Year UN_Estimate UN_Year avg_worldbank_un
1 2021.0 234317.0 2021 19540.805556
2 2020.0 169260.0 2021 19540.805556
3 2021.0 133745.0 2021 19540.805556
4 2021.0 101109.0 2021 19540.805556
5 2021.0 112653.0 2021 19540.805556

Dataframe after dropping temporary column:
Country/Territory UN_Region IMF_Estimate IMF_Year WorldBank_Estimate \
1 Monaco Europe 17377.736041 NaN 234316.0
2 Liechtenstein Europe 17377.736041 NaN 157755.0
3 Luxembourg Europe 132372.000000 2023.0 133590.0
4 Ireland Europe 114581.000000 2023.0 100172.0
5 Bermuda Americas 17377.736041 NaN 114090.0

WorldBank_Year UN_Estimate UN_Year
1 2021.0 234317.0 2021
2 2020.0 169260.0 2021
3 2021.0 133745.0 2021
4 2021.0 101109.0 2021
5 2021.0 112653.0 2021

```

Course Notes

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:



We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

END OF WORKBOOK

Please check through your work thoroughly before submitting and update the table of contents if required.

Please send your completed work booklet to your trainer.

