Pandas
Data Cleaning

1. Sample of Data

```
     Duration          Date  Pulse  Maxpulse  Calories
0          60  '2020/12/01'    110       130     409.1
1          60  '2020/12/02'    117       145     479.0
...................................................................
17         60  '2020/12/17'    100       120     300.0
18         45  '2020/12/18'     90       112       NaN
19         60  '2020/12/19'    103       123     323.0
20         45  '2020/12/20'     97       125     243.0
21         60  '2020/12/21'    108       131     364.2
22         45           NaN    100       119     282.0
23         60  '2020/12/23'    130       101     300.0
24         45  '2020/12/24'    105       132     246.0
25         60  '2020/12/25'    102       126     334.5
26         60   2020/12/26     100       120     250.0
27         60  '2020/12/27'     92       118     241.0
28         60  '2020/12/28'    103       132       NaN
```

Pandas
Data **Cleaning**

2. Cleaning Empty Cells

```python
import pandas as pd

df = pd.read_csv('data.csv')

new_df = df.dropna()

print(new_df.to_string())
```

Note: By default, the dropna() method returns a new DataFrame, and will not change the original.

Pandas
Data Cleaning

3. Replace Empty Values

```python
import pandas as pd


df = pd.read_csv('data.csv')


df.fillna(130, inplace = True)
```

The fillna() method allows us to replace empty cells with a value

Pandas
Data Cleaning

4. Convert Date Into a Correct Format

```python
import pandas as pd

df = pd.read_csv('data.csv')

df['Date'] = pd.to_datetime(df['Date'])

print(df.to_string())
```

Pandas has a to_datetime() method to convert all cells
in the 'Date' column into dates.

Pandas
Data Cleaning

5. Removing Duplicates
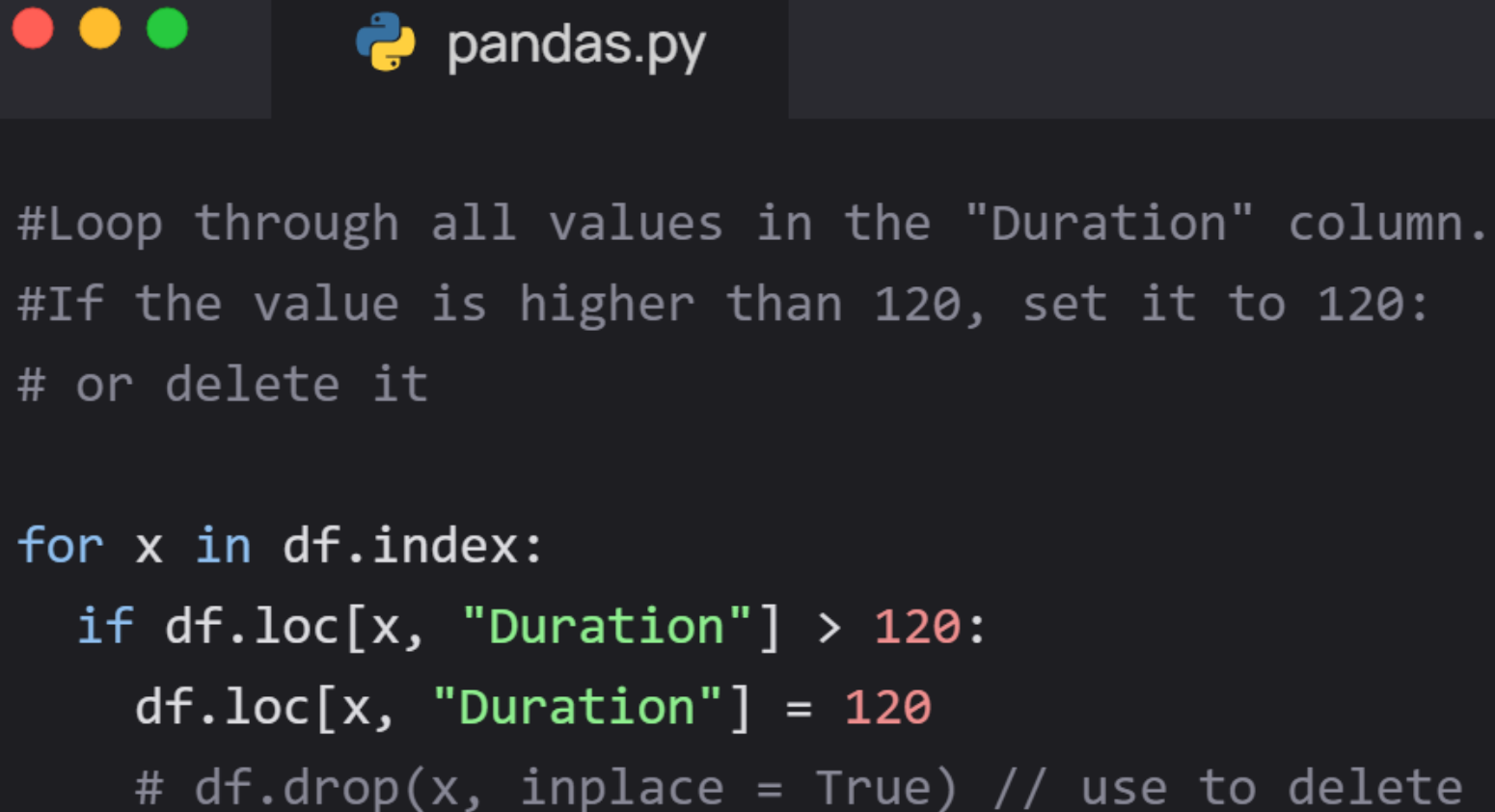
```python
import pandas as pd

df = pd.read_csv('data.csv')

df.drop_duplicates()

print(df.to_string())
```

To remove duplicates, use the drop_duplicates() method.

Pandas
Data Cleaning

6. Fixing Wrong Data

```python
#Loop through all values in the "Duration" column.
#If the value is higher than 120, set it to 120:
# or delete it

for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.loc[x, "Duration"] = 120
    # df.drop(x, inplace = True) // use to delete
```

One way to fix wrong values is to replace them with something else. Another way of handling wrong data is to remove the rows that contains wrong data.

Pandas
Data Cleaning

Find this Helpful? Follow Ashfiquzzaman Sajal for more content!

Pandas
Data Cleaning

Get me in Touch

Ashfiquzzaman Sajal