Intermediate Pandas

```
import pandas as pd
```

```
penguins = pd.read_csv("penguins.csv")
```

```
#preview first 5 rows
penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
penguins.tail()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
# shape of dataframe
penguins.shape
```

```
(344, 7)
```

```
#information of dataframe
penguins.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   species            344 non-null    object 
 1   island             344 non-null    object 
 2   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

```
# select columns
penguins["species"]
# same answer
penguins.species
```

```
0      Adelie
```

```
penguins[["species", "island", "sex"]].tail() #defalut=5
```

	species	island	sex

```
#integer location based indexing (iloc)
penguins.iloc[[0,1,2]]
#or
penguins.iloc[0:3]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
penguins.iloc[0:5, [0,1,5]]
```

	species	island	body_mass_g

```
mini_penguins = penguins.iloc[0:5, 0:3]
mini_penguins
```

	species	island	bill_length_mm

```
#filter rows by a condition
#ดึงเพนกวินทุกตัวที่อยู่ในเกาะ Torgersen
penguins[penguins["island"] == "Torgersen"]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#ดึงเพนกวินที่มี bill length น้อยกว่า 34mm
penguins[penguins["bill_length_mm"] <34]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#filter more than one condition
#ดึงเพนกวินที่อยู่เกาะ torgersen "และ" มี bill_length น้อยกว่า 35mm
penguins[(penguins["island"] == "Torgersen") & (penguins["bill_length_mm"]< 35)]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#ดึงเพนกวินที่อยู่เกาะ torgersen "หรือ" มี bill_length น้อยกว่า 35mm
penguins[(penguins["island"] == "Torgersen") | (penguins["bill_length_mm"]< 35)]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#filter with query()
penguins.query("island == 'Torgersen' & bill_length_mm < 35")
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
# check missing in each column
penguins.isna().sum()
```

```
species              0
```

```
# filter missing values in column sex
penguins[penguins["sex"].isna()]
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#drop na ลบแถวที่มี na ออก
clean_penguins = penguins.dropna()
clean_penguins
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

Mean imputation

replace nan with column mean

```
#fill missing values
top5_penguins= penguins.head(5)
```

```
avg_value = top5_penguins["bill_length_mm"].mean()
print(avg_value)
```

```
38.9
```

```
top5_penguins=top5_penguins['bill_length_mm'].fillna(value=avg_value)
top5_penguins
```

```
0    39.1
```

```
#sort Bill_length_mm low to high, high to low
penguins.dropna().sort_values('bill_length_mm', ascending=False).head(10)
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#sort multiple columns
#เรียงตามชื่อเกาะ ตามด้วย bill_length_mm
penguins.dropna().sort_values(['island', 'bill_length_mm'])
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex

```
#unique values
#เอามาเฉพาะสายพันธุ์ที่ไม่ซ้ำกัน
penguins['species'].unique()
```

```
array(['Adelie', 'Chinstrap', 'Gentoo'], dtype=object)
```

```
#count values
#อยากรู้ว่า มีสายพันธุ์ละกี่ตัว
penguins['species'].value_counts()
```

```
Adelie       152
```

```
#count more than one columns
result = penguins[ [ 'island', 'species']].value_counts().reset_index()

result.columns = ['island', 'species', 'count']

result
```

	island	species	count