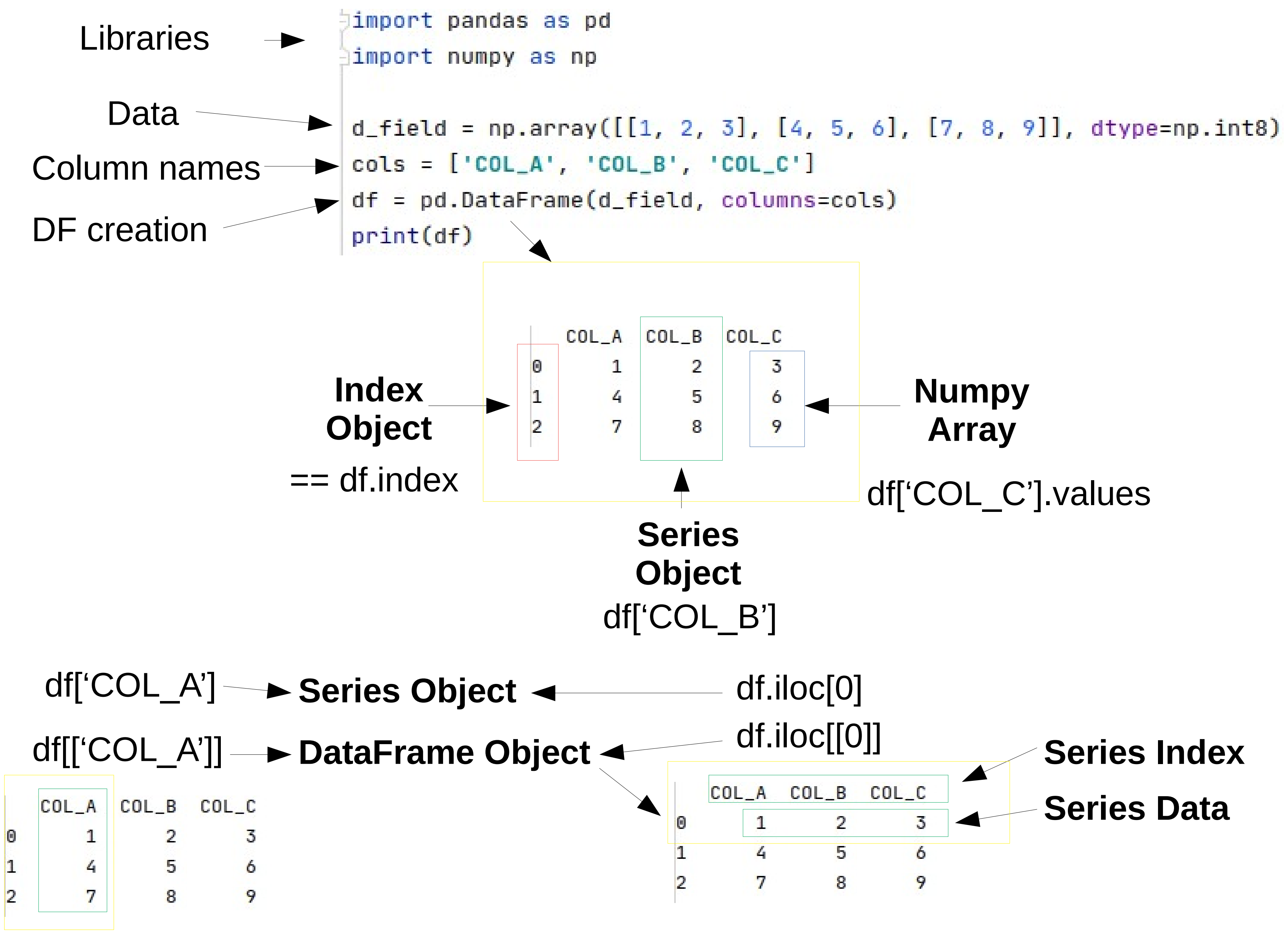
Dataframe



```python
import pandas as pd
import numpy as np

d_field = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]], dtype=np.int8)
cols = ['COL_A', 'COL_B', 'COL_C']
df = pd.DataFrame(d_field, columns=cols)
print(df)
```

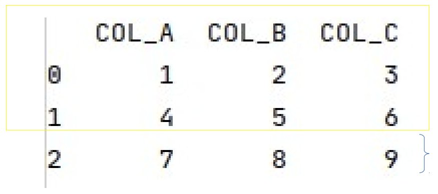
```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

ILOC vs LOC

df.iloc[0:2] →

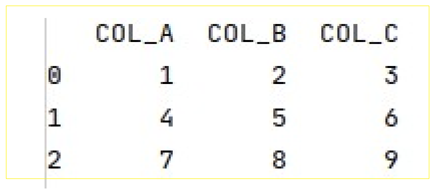


```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

DataFrame Object

df.iloc[start:end], end NOT included!

df.loc[0:2] →



```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

DataFrame Object

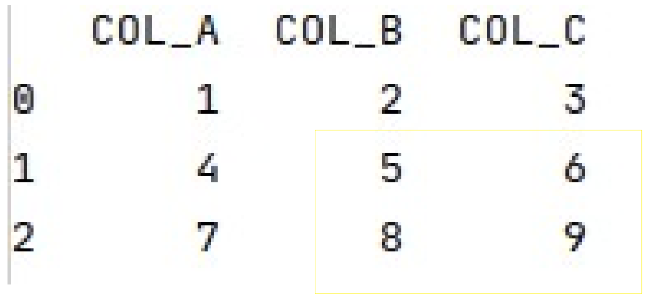
df.loc[start:end], end IS included!

LOC

df.loc[df['COL_A'] > 2, ['COL_B', 'COL_C']]

ILOC

df.iloc[(df['COL_A'] > 2).values, [1, 2]]



```
   COL_A  COL_B  COL_C
0      1      2      3
1      4      5      6
2      7      8      9
```

Debugging

Jupyter-notebooks de-facto debugging is to print. IDE’s provide complete debugger.

Typical errors

Data type:

- arrays vs int vs float vs complex
- Expected Input object type and output object type
  - DataFrame vs Series vs Numpy Array

Data shape (np.array):

- Shape is (n,) → transpose not possible, reshape(n,-1) or (-1,n)
- Shape is (n,1) instead of (1,n) → need transpose (.T)
- Shape is -1 on +1 of expected → check inclusive or noninclusive. [start:end] what is end?

Check list

1. Check error message
2. Check type: print(type(your_data_object))
3. Check data shape: print(data_object.shape) or len/size if not a numpy array

Helps!

A. ?library.function “?pd.DataFrame”
B. help(your.function) “help(pd.DataFrame)”
C. Google for it: network is full of examples, be aware Python 2 vs 3
D. Make small mock-up and and test with prints each step of script is what expected