Pandas is a really good library for data science
Pandas docs: https://pandas.pydata.org/docs/

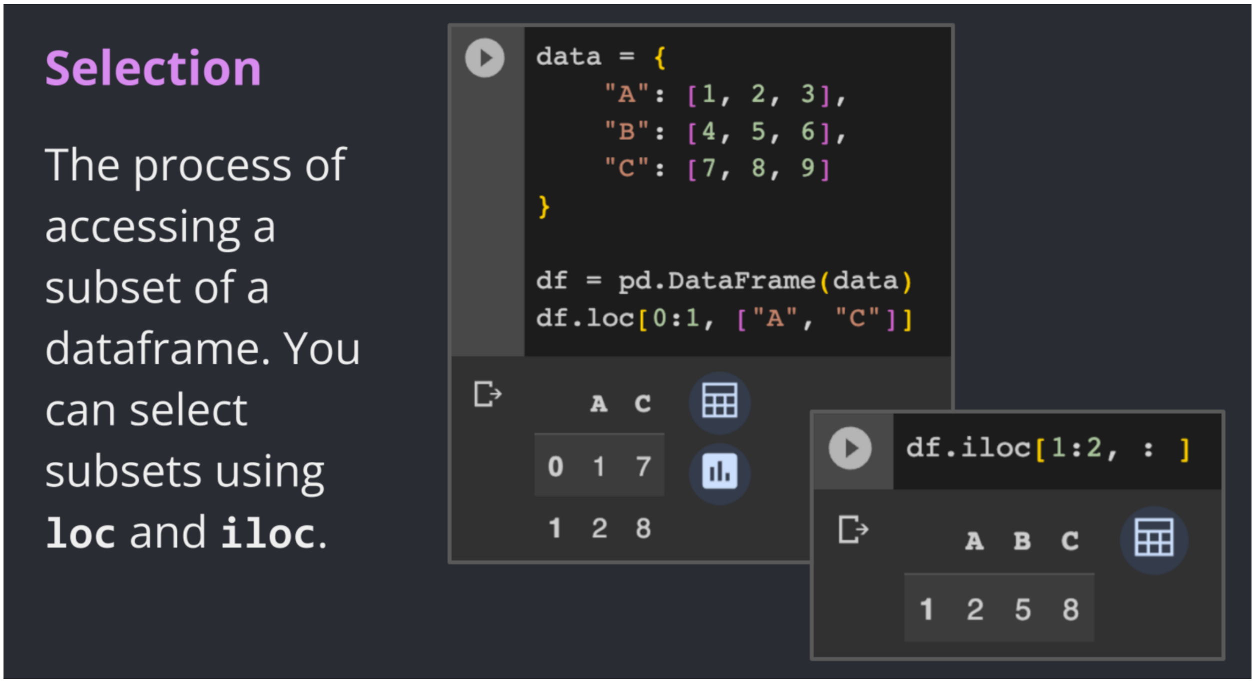
Two main objects of pandas

- Data Frames
    - Col and row-orientated
    - Simplify everything
    - Can create it using a dictionary of lists
    - 2d
        - Could you make it 3d/∞d?
- Series
    - ✨Fancy list✨
    - 1d

Indexing

- loc[]/at[]
    - Label based indexing
- iloc[]/iat[]
    - Numerical based indexing

Selection



Selection

The process of accessing a subset of a dataframe. You can select subsets using `loc` and `iloc`.

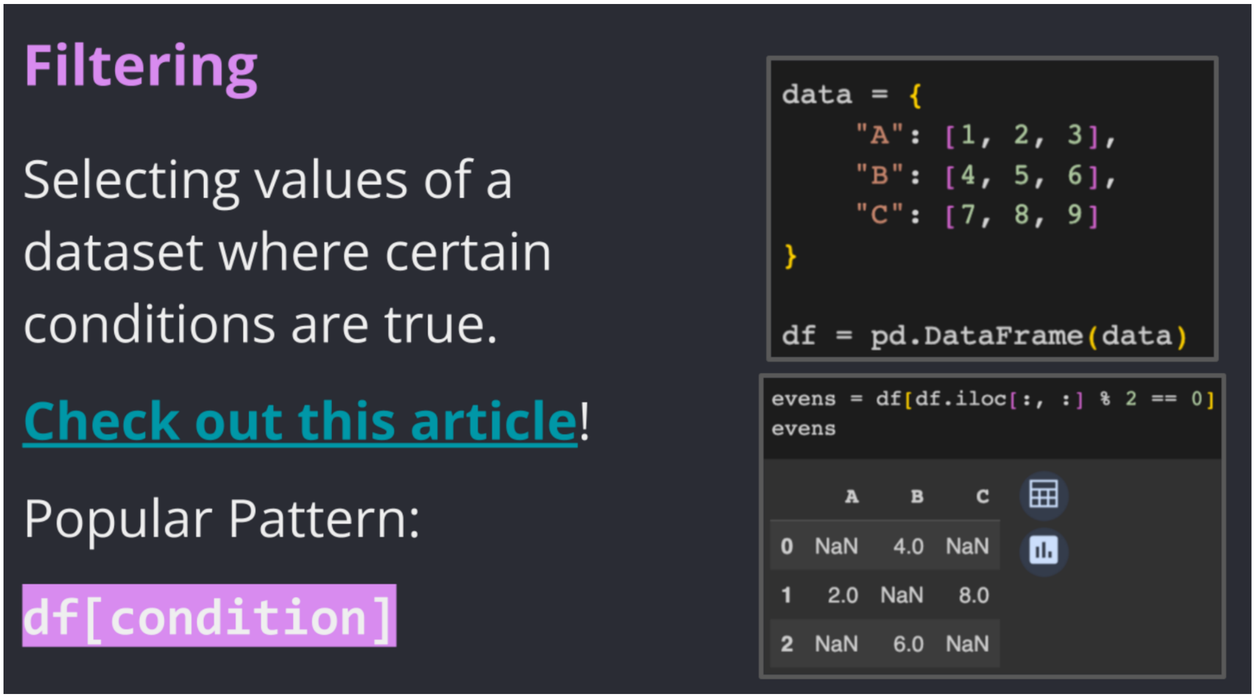
```
data = {
    "A": [1, 2, 3],
    "B": [4, 5, 6],
    "C": [7, 8, 9]
}

df = pd.DataFrame(data)
df.loc[0:1, ["A", "C"]]
```

```
   A  C
0  1  7
1  2  8
```

```
df.iloc[1:2, : ]
```

```
   A  B  C
1  2  5  8
```

- Loc and iloc are different and have different inclusivity



Filtering

Selecting values of a dataset where certain conditions are true.

Check out this article!

Popular Pattern:

`df[condition]`

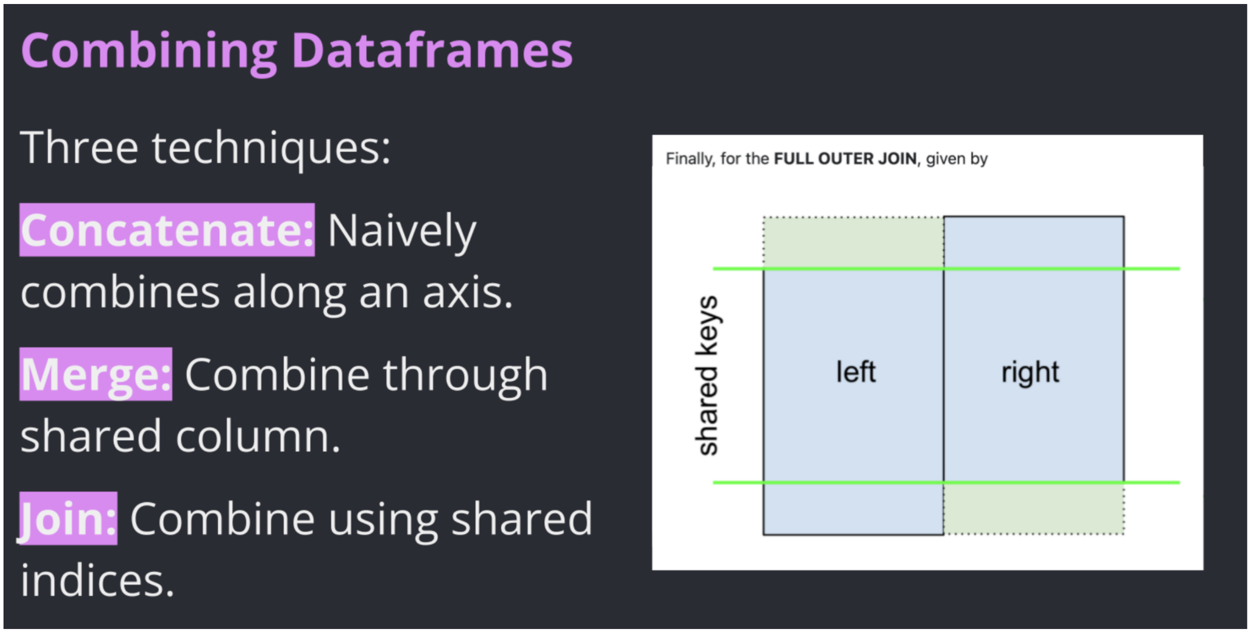
```
data = {
      "A": [1, 2, 3],
      "B": [4, 5, 6],
      "C": [7, 8, 9]
}

df = pd.DataFrame(data)
```

```
evens = df[df.iloc[:, :] % 2 == 0]
evens
```

	A	B	C
0	NaN	4.0	NaN
1	2.0	NaN	8.0
2	NaN	6.0	NaN

https://builtin.com/data-science/pandas-filter

Pandas uses binary operators, |/&, instead of the python way or the normal programming language way, because why would things be standardized?



Combining Dataframes

Three techniques:

Concatenate: Naively combines along an axis.

Merge: Combine through shared column.

Join: Combine using shared indices.

Finally, for the **FULL OUTER JOIN**, given by