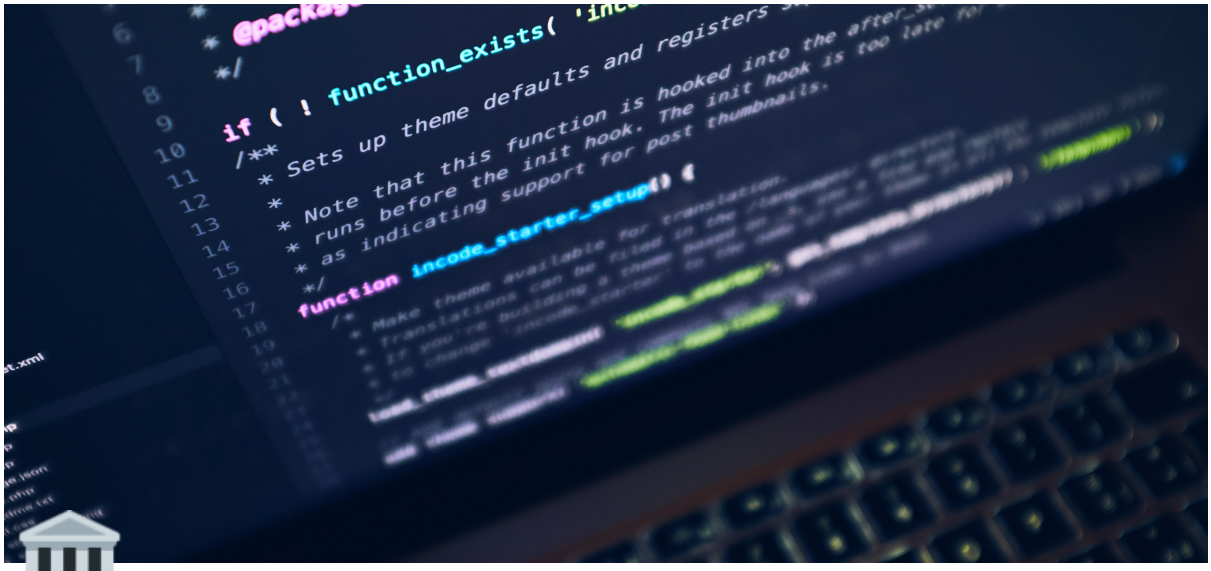


Introduction to Data Science in Python (Course 1)- Coursera U. Michigan

https://github.com/bondeanikets/Introduction-to-Data-Science-in-Python

https://github.com/agniiyer/Introduction-to-Data-Science-in-Python

https://github.com/SayanSeth/Introduction-to-Data-Science-in-Python/tree/master/Assignments

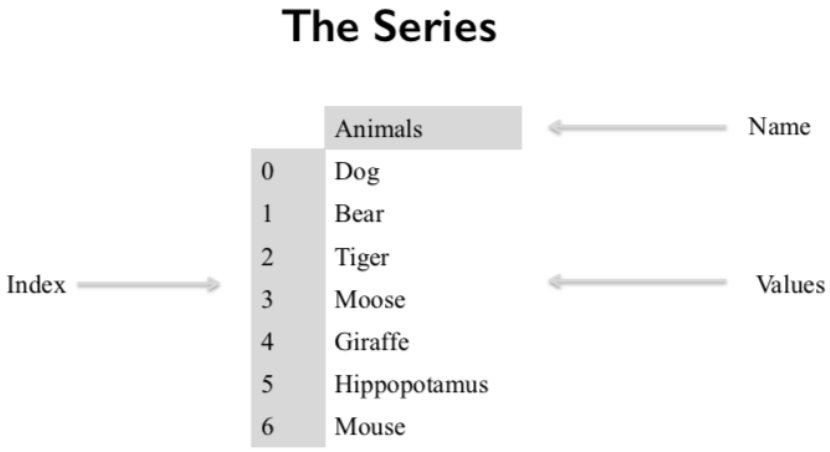
PANDAS

https://medium.com/dunder-data/selecting-subsets-of-data-in-pandas-6fcd0170be9c

https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/

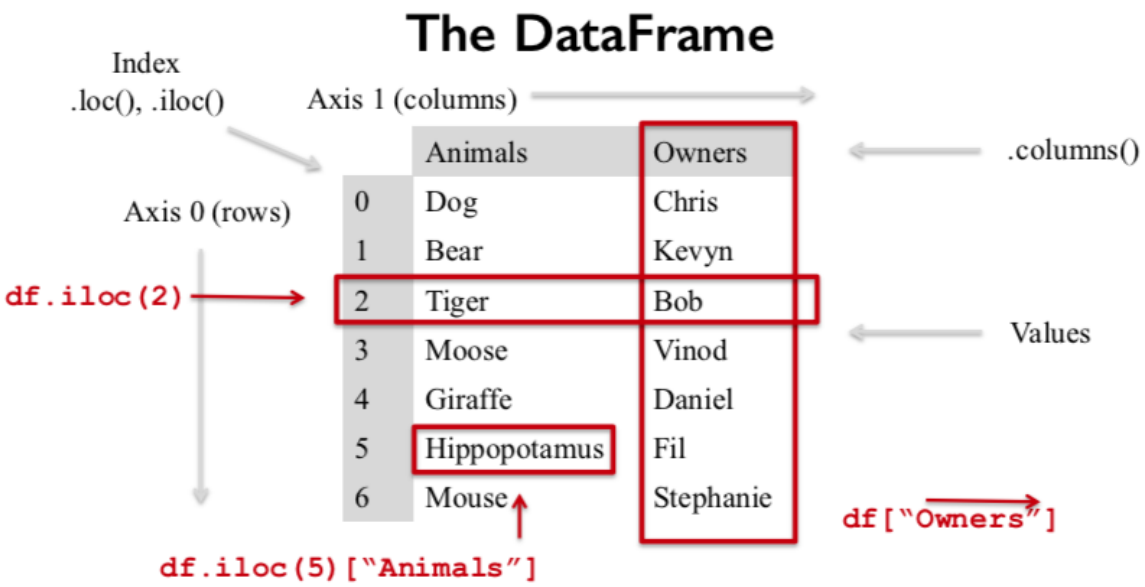
Series Object (1 dimensional, a row).

- An idx and data



- Index can be accessed using `seriesname.index`
- *None* si es todo strings, se convierte en *None*. Si la serie es de números, se convierte en *NaN*
- ```
  sports = {
  'Archery': 'Bhutan',
  'Golf': 'Scotland',
  'Sumo': 'Japan',
  'Taekwondo': 'South Korea'}
  s = pd.Series(sports)
  ```
- `s = pd.Series(['Tiger', 'Bear', 'Moose'], index=['India', 'America', 'Canada'])`
- Iteration using for loop IF vectorization not possible:
  - for label, value in s.iteritems():
    s.loc[label]= value+2

DataFrame Object (2 dimensional, a table).
  ```



- Querying (both series and df):
  - iloc[] attribute, for querying based on position. s.iloc[3] → 'South Korea'. = s[3]
  - loc[] attribute, for querying based on label. s.loc['Golf'] → 'Scotland' = s['Golf']. If indexes are int numbers, pandas will have a problem identifying if you want loc or iloc!
    - When using df, we can pass two labels, one can be a row index and the other column name a.loc['row index name', 'column name'] = a['row index name', 'column name']

- Querying for df directly
  - Projecting a subset of columns
  - Using a boolean mask to filter data.

df

	Animals	Owners
0	Dog	Chris
1	Bear	Kevyn
2	Tiger	Bob
3	Moose	Vinod
4	Giraffe	Daniel
5	Hippo	Fil
6	Mouse	Stephanie

+

Boolean mask

True	True
True	True
False	False
True	True
False	False
False	False
False	False

=

result

	Animals	Owners
0	Dog	Chris
1	Bear	Kevyn
3	Moose	Vinod

df['Gold'] > 0

```
Afghanistan (AFG)                    False
Algeria (ALG)                         True
Argentina (ARG)                       True
Armenia (ARM)                         True
Australasia (ANZ) [ANZ]               True
Australia (AUS) [AUS] [Z]             True
Austria (AUT)                         True
Azerbaijan (AZE)                      True
Bahamas (BAH)                         True
```

only_gold = df.where(df['Gold'] > 0) = df[df['Gold']>0] (filtra solo los Nan)
only_gold.head()

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.1	Total.1	# Games	Gold.2	Silver.2	Bronze.2
Afghanistan (AFG)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Algeria (ALG)	12.0	5.0	2.0	8.0	15.0	3.0	0.0	0.0	0.0	0.0	15.0	5.0	2.0	8.0
Argentina (ARG)	23.0	18.0	24.0	28.0	70.0	18.0	0.0	0.0	0.0	0.0	41.0	18.0	24.0	28.0
Armenia (ARM)	5.0	1.0	2.0	9.0	12.0	6.0	0.0	0.0	0.0	0.0	11.0	1.0	2.0	9.0
Australasia (ANZ) [ANZ]	2.0	3.0	4.0	5.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	3.0	4.0	5.0

- Each boolean mask has to be in parenthesis when using more than one.
- To change index names:
  - `df['country'] = df.index`
    `df = df.set_index('Gold')`
- We can have multiple indexes: `df.set_index(['Idx1', 'Idx2'])`

- Useful fx
  - idxmax()
  - Count distict values, use nunique()
  - Pandas Series.sort_values()