Pandas reset_index

SHARP SIGHT

What you'll learn

- How to use the `reset_index()` method

- How to "reset" an index for a Pandas DataFrame
  - Remove the current index
  - Turn the index back into a column

A quick introduction to Pandas reset_index

DataFrames have an index, which is like a set of labels for each row

employee_id	first_name
101	John
102	Paul
103	George
104	Ringo
105	Stewart

There are a variety of ways to set an index ...

Including setting an index during DataFrame creation, and also using the `set_index()` method

We can *remove* an existing index with the reset_index method

```
emp_fst_nm.reset_index()
```

emp_fst_nm

employee_id	first_name
101	John
102	Paul
103	George
104	Ringo
105	Stewart

Out:

index	employee_id	first_name
0	101	John
1	102	Paul
2	103	George
3	104	Ringo
4	105	Stewart

When we reset the index, the index will become the range of numbers starting at 0

```
emp_fst_nm.reset_index()
```

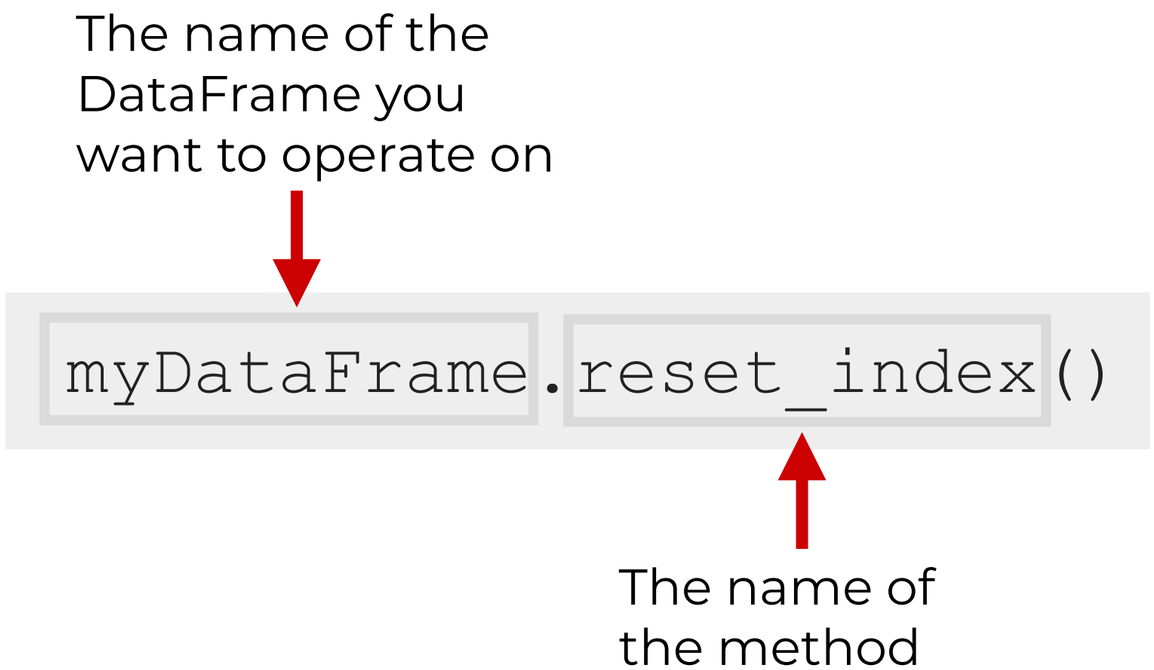
Out:

index	employee_id	first_name
0	101	John
1	102	Paul
2	103	George
3	104	Ringo
4	105	Stewart

But the exact details of what happens depend on the parameters you use when you call `reset_index()`

Pandas reset_index syntax

Basic syntax: pandas reset_index



PANDAS RESET_INDEX WITH A FEW ADDITIONAL PARAMETERS

The name of the DataFrame you want to operate on

Which "level" of the index you want to remove, if the index has multiple levels (default is all levels)

Whether you want to operate on directly on the DataFrame (default is `False`)

```
myDataFrame.reset_index(level=, drop=, inplace= )
```

The name of the method

Whether you want to delete the index when it is removed (default is `False`)

Parameters of reset_index

The parameters of Pandas reset_index

Parameter	What it does	Format	Default	Required?
`level=`	Specify which level you want to "reset" and remove from the index. Only applicable if you have multiple levels in your index.	`int`, `str`, `tuple`, or `list`		No
`drop=`	Specify whether or not the set_index method will “drop” the index from the DataFrame	`True` or `False`	`False`	No
`inplace=`	Specify whether you want to modify your DataFrame “in place”	`True` or `False`	`False`	No

Note: There are other parameters for reset_index() but they are rarely used, so we won't discuss them here.

The output of Pandas reset_index

- By default, `reset_index()` will produce a new DataFrame with the new index
  - by default, the `inplace` parameter is set to `inplace = False`

- If you set `inplace = True`, `reset_index()` will directly modify the original DataFrame

- By default, the `reset_index()` will turn the index back into a normal column in the output DataFrame
  - If you set `drop = True`, `reset_index()` will *drop* the index entirely

Recap

Recap of what we learned

- You can reset the index of a DataFrame with `reset_index()`

- By default, `reset_index()` will create a new DataFrame
  - If you set `inplace = True`, the method will directly modify the original data

- By default, `reset_index()` will turn the index into a column
  - If you set `drop = True`, it will drop the index entirely

- **Next Steps**: Watch the code walkthrough video for step-by-step examples of `reset_index()`