

# pandas.DataFrame.reset\_index

`DataFrame.reset_index(level=None, drop=False, inplace=False, col_level=0, col_fill="")`

[\[source\]](#)

Reset the index, or a level of it.

Reset the index of the DataFrame, and use the default one instead. If the DataFrame has a Multi-Index, this method can remove one or more levels.

**level** : *int, str, tuple, or list, default None*

Only remove the given levels from the index. Removes all levels by default.

**drop** : *bool, default False*

Do not try to insert index into dataframe columns. This resets the index to the default integer index.

**inplace** : *bool, default False*

**Parameters:** Modify the DataFrame in place (do not create a new object).

**col\_level** : *int or str, default 0*

If the columns have multiple levels, determines which level the labels are inserted into. By default it is inserted into the first level.

**col\_fill** : *object, default ""*

If the columns have multiple levels, determines how the other levels are named. If None then the index name is repeated.

**DataFrame**

**Returns:** DataFrame with the new index.

## See also:

[DataFrame.set\\_index](#)

Opposite of reset\_index.

[DataFrame.reindex](#)

Change to new indices or expand indices.

[DataFrame.reindex\\_like](#)

Change to same indices as other DataFrame.

## Examples

```
>>> df = pd.DataFrame([('bird', 389.0),
...                    ('bird', 24.0),
...                    ('mammal', 80.5),
...                    ('mammal', np.nan)],
...                    index=['falcon', 'parrot', 'lion', 'monkey'],
...                    columns=('class', 'max_speed'))
>>> df
```

	class	max_speed
falcon	bird	389.0
parrot	bird	24.0

[Scroll To Top](#)

lion	mammal	80.5
monkey	mammal	NaN

When we reset the index, the old index is added as a column, and a new sequential index is used:

```
>>> df.reset_index()
   index  class  max_speed
0  falcon   bird    389.0
1  parrot   bird     24.0
2    lion  mammal     80.5
3  monkey  mammal     NaN
```

We can use the *drop* parameter to avoid the old index being added as a column:

```
>>> df.reset_index(drop=True)
   class  max_speed
0   bird    389.0
1   bird     24.0
2  mammal     80.5
3  mammal     NaN
```

You can also use *reset\_index* with *MultiIndex*.

```
>>> index = pd.MultiIndex.from_tuples([('bird', 'falcon'),
...                                  ('bird', 'parrot'),
...                                  ('mammal', 'lion'),
...                                  ('mammal', 'monkey')],
...                                  names=['class', 'name'])
>>> columns = pd.MultiIndex.from_tuples([('speed', 'max'),
...                                     ('species', 'type')])
>>> df = pd.DataFrame([(389.0, 'fly'),
...                   ( 24.0, 'fly'),
...                   ( 80.5, 'run'),
...                   (np.nan, 'jump')],
...                   index=index,
...                   columns=columns)
>>> df
```

		speed	species
		max	type
class	name		
bird	falcon	389.0	fly
	parrot	24.0	fly
mammal	lion	80.5	run
	monkey	NaN	jump

If the index has multiple levels, we can reset a subset of them:

```
>>> df.reset_index(level='class')
   class  speed species
   name  max  type
falcon  bird  389.0   fly
parrot  bird  24.0   fly
lion    mammal 80.5   run
monkey  mammal NaN   jump
```

[Scroll To Top](#)

If we are not dropping the index, by default, it is placed in the top level. We can place it in another level:

```
>>> df.reset_index(level='class', col_level=1)
           speed species
name      class    max   type
falcon    bird  389.0    fly
parrot    bird   24.0    fly
lion     mammal   80.5    run
monkey   mammal    NaN   jump
```

When the index is inserted under another level, we can specify under which one with the parameter `col_fill`:

```
>>> df.reset_index(level='class', col_level=1, col_fill='species')
           species speed species
name      class    max   type
falcon    bird  389.0    fly
parrot    bird   24.0    fly
lion     mammal   80.5    run
monkey   mammal    NaN   jump
```

If we specify a nonexistent level for `col_fill`, it is created:

```
>>> df.reset_index(level='class', col_level=1, col_fill='genus')
           genus speed species
name      class    max   type
falcon    bird  389.0    fly
parrot    bird   24.0    fly
lion     mammal   80.5    run
monkey   mammal    NaN   jump
```

[Scroll To Top](#)