☰   ▌▎ **pandas**   🔍 ⮜

pandas.DataFrame.values

property `DataFrame.`**`values`**   **[source]**

Return a Numpy representation of the DataFrame.

> ⚠️ **Warning**
>
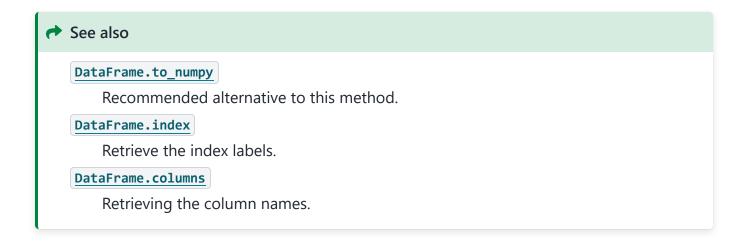> We recommend using `DataFrame.to_numpy()` instead.

Only the values in the DataFrame will be returned, the axes labels will be removed.

Returns:

    numpy.ndarray

        The values of the DataFrame.

> ➹ **See also**
>
> `DataFrame.to_numpy`
>
>   Recommended alternative to this method.
>
> `DataFrame.index`
>
>   Retrieve the index labels.
>
> `DataFrame.columns`
>
>   Retrieving the column names.

Notes

The dtype will be a lower-common-denominator dtype (implicit upcasting); that is to say if the dtypes (even of numeric types) are mixed, the one that accommodates all will be chosen. Use this with care if you are not dealing with the blocks.

e.g. If the dtypes are float16 and float32, dtype will be upcast to float32. If dtypes are int32 and

Skip to main content

uint64 will result in a float64 dtype.

Examples

A DataFrame where all columns are the same type (e.g., int64) results in an array of the same
type.

```
>>> df = pd.DataFrame({'age':     [ 3,  29],
...                    'height': [94, 170],
...                    'weight': [31, 115]})
>>> df
   age  height  weight
0    3      94      31
1   29     170     115
>>> df.dtypes
age       int64
height    int64
weight    int64
dtype: object
>>> df.values
array([[  3,  94,  31],
       [ 29, 170, 115]])
```

A DataFrame with mixed type columns(e.g., str/object, int64, float32) results in an ndarray of the
broadest type that accommodates these mixed types (e.g., object).

```
>>> df2 = pd.DataFrame([('parrot',    24.0, 'second'),
...                     ('lion',      80.5, 1),
...                     ('monkey', np.nan, None)],
...                   columns=('name', 'max_speed', 'rank'))
>>> df2.dtypes
name          object
max_speed    float64
rank          object
dtype: object
>>> df2.values
array([['parrot', 24.0, 'second'],
       ['lion', 80.5, 1],
       ['monkey', nan, None]], dtype=object)
```

Skip to main content