User Guide

The User Guide covers all of pandas by topic area. Each of the subsections introduces a topic (such as “working with missing data”), and discusses how pandas approaches the problem, with many examples throughout.

Users brand-new to pandas should start with 10 minutes to pandas.

For a high level summary of the pandas fundamentals, see Intro to data structures and Essential basic functionality.

Further information on any specific method can be obtained in the API reference.

How to read these guides

In these guides you will see input code inside code blocks such as:

```
import pandas as pd
pd.DataFrame({'A': [1, 2, 3]})
```

or:

```
In [1]: import pandas as pd

In [2]: pd.DataFrame({'A': [1, 2, 3]})
Out[2]: 
   A
0  1
1  2
2  3
```

The first block is a standard python input, while in the second the `In [1]:` indicates the input is inside a notebook. In Jupyter Notebooks the last line is printed and plots are shown inline.

For example:

Skip to main content

```
In [4]: a
Out[4]: 1
```

is equivalent to:

```
a = 1
print(a)
```

Guides

- 10 minutes to pandas
  - Basic data structures in pandas
  - Object creation
  - Viewing data
  - Selection
  - Missing data
  - Operations
  - Merge
  - Grouping
  - Reshaping
  - Time series
  - Categoricals
  - Plotting
  - Importing and exporting data
  - Gotchas
- Intro to data structures
  - Series
  - DataFrame
- Essential basic functionality
  - Head and tail
  - Attributes and underlying data
  - Accelerated operations
  - Flexible binary operations

Skip to main content

  - Function application
  - Reindexing and altering labels
  - Iteration
  - .dt accessor
  - Vectorized string methods
  - Sorting
  - Copying
  - dtypes
  - Selecting columns based on `dtype`
- IO tools (text, CSV, HDF5, …)
  - CSV & text files
  - JSON
  - HTML
  - LaTeX
  - XML
  - Excel files
  - OpenDocument Spreadsheets
  - Binary Excel (.xlsb) files
  - Calamine (Excel and ODS files)
  - Clipboard
  - Pickling
  - msgpack
  - HDF5 (PyTables)
  - Feather
  - Parquet
  - ORC
  - SQL queries
  - Google BigQuery
  - Stata format
  - SAS formats
  - SPSS formats
  - Other file formats

Skip to main content

  - Performance considerations
- PyArrow Functionality
  - Data Structure Integration
  - Operations
  - I/O Reading
- Indexing and selecting data
  - Different choices for indexing
  - Basics
  - Attribute access
  - Slicing ranges
  - Selection by label
  - Selection by position
  - Selection by callable
  - Combining positional and label-based indexing
  - Selecting random samples
  - Setting with enlargement
  - Fast scalar value getting and setting
  - Boolean indexing
  - Indexing with isin
  - The `where()` Method and Masking
  - Setting with enlargement conditionally using `numpy()`
  - The `query()` Method
  - Duplicate data
  - Dictionary-like `get()` method
  - Looking up values by index/column labels
  - Index objects
  - Set / reset index
  - Returning a view versus a copy
- MultiIndex / advanced indexing
  - Hierarchical indexing (MultiIndex)
  - Advanced indexing with hierarchical index
  - Sorting a `MultiIndex`

Skip to main content

- Index types
- Miscellaneous indexing FAQ

Copy-on-Write (CoW)

- Previous behavior
- Migrating to Copy-on-Write
- Description
- Chained Assignment
- Read-only NumPy arrays
- Patterns to avoid
- Copy-on-Write optimizations
- How to enable CoW

Merge, join, concatenate and compare

- `concat()`
- `merge()`
- `DataFrame.join()`
- `merge_ordered()`
- `merge_asof()`
- `compare()`

Reshaping and pivot tables

- `pivot()` and `pivot_table()`
- `stack()` and `unstack()`
- `melt()` and `wide_to_long()`
- `get_dummies()` and `from_dummies()`
- `explode()`
- `crosstab()`
- `cut()`
- `factorize()`

Working with text data

- Text data types
- String methods
- Splitting and replacing strings
- Concatenation

Skip to main content

  - Extracting substrings
  - Testing for strings that match or contain a pattern
  - Creating indicator variables
  - Method summary
- Working with missing data
  - Values considered “missing”
  - `NA` semantics
  - Inserting missing data
  - Calculations with missing data
  - Dropping missing data
  - Filling missing data
- Duplicate Labels
  - Consequences of Duplicate Labels
  - Duplicate Label Detection
  - Disallowing Duplicate Labels
- Categorical data
  - Object creation
  - CategoricalDtype
  - Description
  - Working with categories
  - Sorting and order
  - Comparisons
  - Operations
  - Data munging
  - Getting data in/out
  - Missing data
  - Differences to R’s `factor`
  - Gotchas
- Nullable integer data type
  - Construction
  - Operations
  - Scalar NA Value

Skip to main content

- Indexing with NA values
- Kleene logical operations

Chart visualization

- Basic plotting: `plot`
- Other plots
- Plotting with missing data
- Plotting tools
- Plot formatting
- Plotting directly with Matplotlib
- Plotting backends

Table Visualization

- Styler Object and Customising the Display
- Formatting the Display
- Styler Object and HTML
- Methods to Add Styles
- Table Styles
- Setting Classes and Linking to External CSS
- Styler Functions
- Tooltips and Captions
- Finer Control with Slicing
- Optimization
- Builtin Styles
- Sharing styles
- Limitations
- Other Fun and Useful Stuff
- Export to Excel
- Export to LaTeX
- More About CSS and HTML
- Extensibility

Group by: split-apply-combine

- Splitting an object into groups
- Iterating through groups

Skip to main content

- Aggregation
- Transformation
- Filtration
- Flexible `apply`
- Numba Accelerated Routines
- Other useful features
- Examples

Windowing operations

- Overview
- Rolling window
- Weighted window
- Expanding window
- Exponentially weighted window

Time series / date functionality

- Overview
- Timestamps vs. time spans
- Converting to timestamps
- Generating ranges of timestamps
- Timestamp limitations
- Indexing
- Time/date components
- DateOffset objects
- Time Series-related instance methods
- Resampling
- Time span representation
- Converting between representations
- Representing out-of-bounds spans
- Time zone handling

Time deltas

- Parsing
- Operations
- Reductions

Skip to main content

- Attributes
- TimedeltaIndex
- Resampling

Options and settings

- Overview
- Available options
- Getting and setting options
- Setting startup options in Python/IPython environment
- Frequently used options
- Number formatting
- Unicode formatting
- Table schema display

Enhancing performance

- Cython (writing C extensions for pandas)
- Numba (JIT compilation)
- Expression evaluation via `eval()`

Scaling to large datasets

- Load less data
- Use efficient datatypes
- Use chunking
- Use Other Libraries

Sparse data structures

- SparseArray
- SparseDtype
- Sparse accessor
- Sparse calculation
- Interaction with *scipy.sparse*

Frequently Asked Questions (FAQ)

- DataFrame memory usage
- Using if/truth statements with pandas
- Mutating with User Defined Function (UDF) methods
- Missing value representation for NumPy types

Skip to main content

  - Thread-safety
  - Byte-ordering issues
- Cookbook
  - Idioms
  - Selection
  - Multiindexing
  - Missing data
  - Grouping
  - Timeseries
  - Merge
  - Plotting
  - Data in/out
  - Computation
  - Timedeltas
  - Creating example data
  - Constant series

‹ Previous

Next ›

© 2024, pandas via NumFOCUS, Inc. Hosted by OVHcloud.

Created using Sphinx 7.2.6.

Built with the PyData Sphinx Theme 0.14.4.