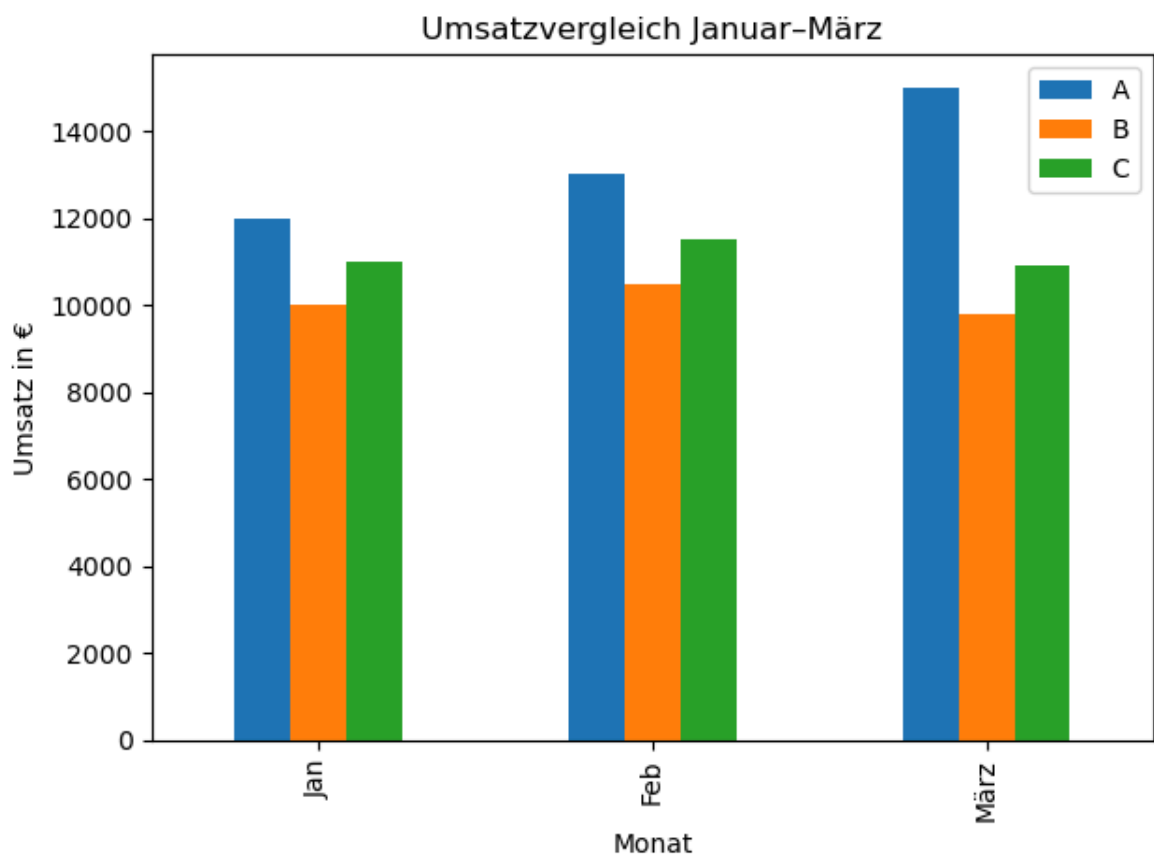


```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [3]: df = pd.DataFrame({
    "Monat": ["Jan", "Feb", "März"],
    "A": [12000, 13000, 15000],
    "B": [10000, 10500, 9800],
    "C": [11000, 11500, 10900]
})
```

```
In [4]: df.set_index("Monat").plot(kind="bar") # Säulendiagramm
plt.title("Umsatzvergleich Januar-März")
plt.ylabel("Umsatz in €")
plt.xlabel("Monat")
plt.tight_layout()
plt.show()
```



```
In [5]: jupyter nbconvert Daten visualisieren.ipynb --to pdf
```

```
Cell In[5], line 1
    jupyter nbconvert Daten visualisieren.ipynb --to pdf
    ^
SyntaxError: invalid syntax
```

```
In [10]: !jupyter nbconvert Daten_visualisieren_matplotlib.ipynb --to pdfdas
```

[NbConvertApp] WARNING | pattern 'Daten_visualisieren_matplotlib.ipynb' matched no files

This application is used to convert notebook files (*.ipynb) to various other formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options

=====

The options below are convenience aliases to configurable class-options, as listed in the "Equivalent to" description-line of the aliases.

To see all configurable class-options for some <cmd>, use:

<cmd> --help-all

--debug

set log level to logging.DEBUG (maximize logging output)

Equivalent to: [--Application.log_level=10]

--show-config

Show the application's configuration (human-readable format)

Equivalent to: [--Application.show_config=True]

--show-config-json

Show the application's configuration (json format)

Equivalent to: [--Application.show_config_json=True]

--generate-config

generate default config file

Equivalent to: [--JupyterApp.generate_config=True]

-y

Answer yes to any questions instead of prompting.

Equivalent to: [--JupyterApp.answer_yes=True]

--execute

Execute the notebook prior to export.

Equivalent to: [--ExecutePreprocessor.enabled=True]

--allow-errors

Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '--execute' was specified, too.

Equivalent to: [--ExecutePreprocessor.allow_errors=True]

--stdin

read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'

Equivalent to: [--NbConvertApp.from_stdin=True]

--stdout

Write notebook output to stdout instead of files.

Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]

--inplace

Run nbconvert in place, overwriting the existing notebook (only relevant when converting to notebook format)

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory=]

--clear-output

Clear output of current file and save in place, overwriting the existing notebook.

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory= --ClearOutputPreprocessor.enabled=True]

--no-prompt

Exclude input and output prompts from converted document.

Equivalent to: [--TemplateExporter.exclude_input_prompt=True --TemplateExporter.exclude_output_prompt=True]

--no-input

```

    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --TemplateExporter.exclude_input=True --TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is only useful for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
    ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf']
    or a dotted object name that represents the import path for an
    ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized. This should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the

```

results of the conversion

```

Default: 'FileWriter'
Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                results of the conversion
Default: ''
Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    Overwrite base name use for output files.
        Supports pattern replacements '{notebook_name}'.
Default: '{notebook_name}'
Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                to output to the directory of each notebook. To
recover
                                previous default behaviour (outputting to the c
urrent
                                working directory) use . as the flag value.
Default: ''
Equivalent to: [--FileWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
        This defaults to the reveal CDN, but can be any url pointing to a cop
y
        of reveal.js.
        For speaker notes to work, this must be a relative path to a local
copy of reveal.js: e.g., "reveal.js".
        If a relative path is given, it must be a subdirectory of the
current directory (from which the server is run).
        See the usage documentation
        (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html
-slideshow)
        for more details.
Default: ''
Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
        Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

```

Examples

The simplest way to use nbconvert is

```
> jupyter nbconvert mynotebook.ipynb --to html
```

Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf'].

```
> jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes 'base', 'article' and 'report'. HTML includes 'basic', 'lab' and 'classic'. You can specify the flavor of the format used.

```
> jupyter nbconvert --to html --template lab mynotebook.ipynb
```

You can also pipe the output to stdout, rather than a file

```
> jupyter nbconvert mynotebook.ipynb --stdout
```

PDF is generated via latex

```
> jupyter nbconvert mynotebook.ipynb --to pdf
```

You can get (and serve) a Reveal.js-powered slideshow

```
> jupyter nbconvert myslides.ipynb --to slides --post serve
```

Multiple notebooks can be given at the command line in a couple of different ways:

```
> jupyter nbconvert notebook*.ipynb
```

```
> jupyter nbconvert notebook1.ipynb notebook2.ipynb
```

or you can specify the notebooks list in a config file, containing::

```
c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
> jupyter nbconvert --config mycfg.py
```

To see all available configurables, use `--help-all`.

```
In [11]: + nbconvert
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[11], line 1
----> 1 + nbconvert
NameError: name 'nbconvert' is not defined
```

```
In [12]: Daten_visualisieren_matplotlib nbconvert
```

```
Cell In[12], line 1
    Daten_visualisieren_matplotlib nbconvert
                                   ^
SyntaxError: invalid syntax
```

```
In [13]: !jupyter nbconvert Daten_visualisieren_matplotlib.ipynb --to html
```

[NbConvertApp] WARNING | pattern 'Daten_visualisieren_matplotlib.ipynb' matched no files

This application is used to convert notebook files (*.ipynb) to various other formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options

=====

The options below are convenience aliases to configurable class-options, as listed in the "Equivalent to" description-line of the aliases.

To see all configurable class-options for some <cmd>, use:

<cmd> --help-all

--debug

set log level to logging.DEBUG (maximize logging output)

Equivalent to: [--Application.log_level=10]

--show-config

Show the application's configuration (human-readable format)

Equivalent to: [--Application.show_config=True]

--show-config-json

Show the application's configuration (json format)

Equivalent to: [--Application.show_config_json=True]

--generate-config

generate default config file

Equivalent to: [--JupyterApp.generate_config=True]

-y

Answer yes to any questions instead of prompting.

Equivalent to: [--JupyterApp.answer_yes=True]

--execute

Execute the notebook prior to export.

Equivalent to: [--ExecutePreprocessor.enabled=True]

--allow-errors

Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '--execute' was specified, too.

Equivalent to: [--ExecutePreprocessor.allow_errors=True]

--stdin

read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'

Equivalent to: [--NbConvertApp.from_stdin=True]

--stdout

Write notebook output to stdout instead of files.

Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]

--inplace

Run nbconvert in place, overwriting the existing notebook (only relevant when converting to notebook format)

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory=]

--clear-output

Clear output of current file and save in place, overwriting the existing notebook.

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory= --ClearOutputPreprocessor.enabled=True]

--no-prompt

Exclude input and output prompts from converted document.

Equivalent to: [--TemplateExporter.exclude_input_prompt=True --TemplateExporter.exclude_output_prompt=True]

--no-input

```

    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --TemplateExporter.exclude_input=True --TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is only useful for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
    ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf']
    or a dotted object name that represents the import path for an
    ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized. This should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the

```

results of the conversion

```

Default: 'FileWriter'
Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                results of the conversion
Default: ''
Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    Overwrite base name use for output files.
        Supports pattern replacements '{notebook_name}'.
Default: '{notebook_name}'
Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                to output to the directory of each notebook. To
recover
                                previous default behaviour (outputting to the c
urrent
                                working directory) use . as the flag value.
Default: ''
Equivalent to: [--FileWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
        This defaults to the reveal CDN, but can be any url pointing to a cop
y
        of reveal.js.
        For speaker notes to work, this must be a relative path to a local
        copy of reveal.js: e.g., "reveal.js".
        If a relative path is given, it must be a subdirectory of the
        current directory (from which the server is run).
        See the usage documentation
        (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html
-slideshow)
        for more details.
Default: ''
Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
        Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

```

Examples

The simplest way to use nbconvert is

```
> jupyter nbconvert mynotebook.ipynb --to html
```

Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf'].

```
> jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes 'base', 'article' and 'report'. HTML includes 'basic', 'lab' and 'classic'. You can specify the flavor of the format used.


```
> jupyter nbconvert --to html --template lab mynotebook.ipynb
```

You can also pipe the output to stdout, rather than a file

```
> jupyter nbconvert mynotebook.ipynb --stdout
```

PDF is generated via latex

```
> jupyter nbconvert mynotebook.ipynb --to pdf
```

You can get (and serve) a Reveal.js-powered slideshow

```
> jupyter nbconvert myslides.ipynb --to slides --post serve
```

Multiple notebooks can be given at the command line in a couple of different ways:

```
> jupyter nbconvert notebook*.ipynb
```

```
> jupyter nbconvert notebook1.ipynb notebook2.ipynb
```

or you can specify the notebooks list in a config file, containing::

```
c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
> jupyter nbconvert --config mycfg.py
```

To see all available configurables, use `--help-all`.

```
In [14]: !jupyter nbconvert Daten_visualisieren_matplotlib.ipynb --to html
```

[NbConvertApp] WARNING | pattern 'Daten_visualisieren_matplotlib.ipynb' matched no files

This application is used to convert notebook files (*.ipynb) to various other formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options

=====

The options below are convenience aliases to configurable class-options, as listed in the "Equivalent to" description-line of the aliases.

To see all configurable class-options for some <cmd>, use:

<cmd> --help-all

--debug

set log level to logging.DEBUG (maximize logging output)

Equivalent to: [--Application.log_level=10]

--show-config

Show the application's configuration (human-readable format)

Equivalent to: [--Application.show_config=True]

--show-config-json

Show the application's configuration (json format)

Equivalent to: [--Application.show_config_json=True]

--generate-config

generate default config file

Equivalent to: [--JupyterApp.generate_config=True]

-y

Answer yes to any questions instead of prompting.

Equivalent to: [--JupyterApp.answer_yes=True]

--execute

Execute the notebook prior to export.

Equivalent to: [--ExecutePreprocessor.enabled=True]

--allow-errors

Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '--execute' was specified, too.

Equivalent to: [--ExecutePreprocessor.allow_errors=True]

--stdin

read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'

Equivalent to: [--NbConvertApp.from_stdin=True]

--stdout

Write notebook output to stdout instead of files.

Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]

--inplace

Run nbconvert in place, overwriting the existing notebook (only relevant when converting to notebook format)

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory=]

--clear-output

Clear output of current file and save in place, overwriting the existing notebook.

Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory= --ClearOutputPreprocessor.enabled=True]

--no-prompt

Exclude input and output prompts from converted document.

Equivalent to: [--TemplateExporter.exclude_input_prompt=True --TemplateExporter.exclude_output_prompt=True]

--no-input

```

    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --TemplateExporter.exclude_input=True --TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is only useful for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
    ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf']
    or a dotted object name that represents the import path for an
    ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized. This should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the

```

results of the conversion

```

Default: 'FileWriter'
Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                results of the conversion
Default: ''
Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    Overwrite base name use for output files.
        Supports pattern replacements '{notebook_name}'.
Default: '{notebook_name}'
Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                to output to the directory of each notebook. To
recover
                                previous default behaviour (outputting to the c
urrent
                                working directory) use . as the flag value.
Default: ''
Equivalent to: [--FileWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
        This defaults to the reveal CDN, but can be any url pointing to a cop
y
        of reveal.js.
        For speaker notes to work, this must be a relative path to a local
        copy of reveal.js: e.g., "reveal.js".
        If a relative path is given, it must be a subdirectory of the
        current directory (from which the server is run).
        See the usage documentation
        (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html
-sideshow)
        for more details.
Default: ''
Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
        Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

```

Examples

The simplest way to use nbconvert is

```
> jupyter nbconvert mynotebook.ipynb --to html
```

Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf'].

```
> jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes 'base', 'article' and 'report'. HTML includes 'basic', 'lab' and 'classic'. You can specify the flavor of the format used.

```
> jupyter nbconvert --to html --template lab mynotebook.ipynb
```

You can also pipe the output to stdout, rather than a file

```
> jupyter nbconvert mynotebook.ipynb --stdout
```

PDF is generated via latex

```
> jupyter nbconvert mynotebook.ipynb --to pdf
```

You can get (and serve) a Reveal.js-powered slideshow

```
> jupyter nbconvert myslides.ipynb --to slides --post serve
```

Multiple notebooks can be given at the command line in a couple of different ways:

```
> jupyter nbconvert notebook*.ipynb
```

```
> jupyter nbconvert notebook1.ipynb notebook2.ipynb
```

or you can specify the notebooks list in a config file, containing::

```
c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
> jupyter nbconvert --config mycfg.py
```

To see all available configurables, use `--help-all`.

```
In [15]: !jupyter nbconvert Daten_visualisieren_matplotlib.ipynb --to html!dir
```

```
2_Diagrammwahl.ipynb
Abschlussarbeit
Daten\ visualisieren\ matplotlib.ipynb
Diagrammwahl.ipynb
Durchschnitt_berechnen.py
Erste\ Analyse.ipynb
Erstelle\ balkend.ipynb
Filiale_hoechster_Umsatz_März.py
Filter_alle_Filialen_Januar_größer_11000.py
Hoechsten_Umsatz_ermitteln_März.py
Umsatzdaten.csv
Untitled5.ipynb
anaconda_projects
scikitlearn.ipynb
umsatzdiagramm.png
```

```
In [ ]: !jupyter nbconvert Daten\ visualisieren\ matplotlib.ipynb --to html
```

```
In [ ]:
```