ipynb\_to\_docx

This is not part of the book, this is a editing tool that is used to create markdownfiles for the jupyter noteboks and then convert them to Microsoft Word in order to use Word and Grammerly to do editing

In [1]:

from pathlib import Path

In [2]:

path=Path('.')  
notebooks=[]  
for p in path.rglob("\*.ipynb"):  
 if '\_build' in str(p.absolute()):  
 continue  
   
 elif 'ipynb\_checkpoints' in str(p.absolute()):  
 continue  
 else:  
 notebooks.append(p)  
notebooks

Out[2]:

[PosixPath('intro.ipynb'),  
 PosixPath('ipynb\_to\_docx.ipynb'),  
 PosixPath('needed\_libraries.ipynb'),  
 PosixPath('AC\_2/AC\_1.ipynb'),  
 PosixPath('AC\_2/AC\_1\_One-Two-Three\_Phase\_AC.ipynb'),  
 PosixPath('AC\_2/AC\_2\_RCL\_filters.ipynb'),  
 PosixPath('AC\_2/AC\_3\_CoupledMag.ipynb'),  
 PosixPath('AC\_2/AC\_4\_PZ.ipynb'),  
 PosixPath('AC\_2/AC\_5\_twoports.ipynb'),  
 PosixPath('AC\_2/AC\_6\_S\_transmissionlines.ipynb'),  
 PosixPath('AC\_2/Untitled.ipynb'),  
 PosixPath('Appendix/skidl\_2\_pyspice\_check.ipynb'),  
 PosixPath('DC\_1/DC\_1.ipynb'),  
 PosixPath('DC\_1/DC\_1\_op\_ohm.ipynb'),  
 PosixPath('DC\_1/DC\_2\_op\_source\_transform.ipynb'),  
 PosixPath('DC\_1/DC\_3\_PracticalSourcs\_sweeps\_subcirucirts.ipynb'),  
 PosixPath('DC\_1/DC\_4\_dependent\_sources\_and\_tf.ipynb'),  
 PosixPath('DC\_1/DC\_5\_Thvenin\_Norton.ipynb'),  
 PosixPath('DC\_1/DC\_6\_DC\_MaxPower\_MaxEfficiency.ipynb'),  
 PosixPath('Diodes\_7/Diodes\_1\_Chartiztion.ipynb'),  
 PosixPath('Diodes\_7/Diodes\_2\_Clipers.ipynb')]

In [3]:

notebooks[2].relative\_to(notebooks[2].parent.parent)

Out[3]:

PosixPath('needed\_libraries.ipynb')

In [ ]:

for nb in notebooks:  
 #base=str(nb.absolute())  
 ipynb='./'+str(nb.relative\_to(nb.parent.parent))  
 base=ipynb[:-6]  
 html=base+'.html'  
 docx=base+'.docx'  
 print(ipynb)  
   
 #uncommment to use  
 #!jupyter nbconvert --to html {ipynb}  
   
 !jupyter nbconvert --TagRemovePreprocessor.enabled=True --TagRemovePreprocessor.remove\_cell\_tags="['remove\_output']" --to html {ipynb}  
   
 !pandoc -s {html} -o {docx}

./intro.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./ipynb\_to\_docx.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./needed\_libraries.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_1.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_1\_One-Two-Three\_Phase\_AC.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_2\_RCL\_filters.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_3\_CoupledMag.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_4\_PZ.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_5\_twoports.ipynb  
[NbConvertApp] WARNING | pattern '{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  
-------  
  
Arguments that take values are actually convenience aliases to full  
Configurables, whose aliases are listed on the help line. For more information  
on full configurables, see '--help-all'.  
  
--debug  
 set log level to logging.DEBUG (maximize logging output)  
--generate-config  
 generate default config file  
-y  
 Answer yes to any questions instead of prompting.  
--execute  
 Execute the notebook prior to export.  
--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.  
--stdin  
 read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.\*'  
--stdout  
 Write notebook output to stdout instead of files.  
--inplace  
 Run nbconvert in place, overwriting the existing notebook (only   
 relevant when converting to notebook format)  
--clear-output  
 Clear output of current file and save in place,   
 overwriting the existing notebook.  
--no-prompt  
 Exclude input and output prompts from converted document.  
--no-input  
 Exclude input cells and output prompts from converted document.   
 This mode is ideal for generating code-free reports.  
--log-level=<Enum> (Application.log\_level)  
 Default: 30  
 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')  
 Set the log level by value or name.  
--config=<Unicode> (JupyterApp.config\_file)  
 Default: ''  
 Full path of a config file.  
--to=<Unicode> (NbConvertApp.export\_format)  
 Default: 'html'  
 The export format to be used, either one of the built-in formats  
 ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc',  
 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs',  
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage',  
 'slides', 'slides\_with\_lenvs'] or a dotted object name that represents the  
 import path for an `Exporter` class  
--template=<Unicode> (TemplateExporter.template\_file)  
 Default: ''  
 Name of the template file to use  
--writer=<DottedObjectName> (NbConvertApp.writer\_class)  
 Default: 'FilesWriter'  
 Writer class used to write the results of the conversion  
--post=<DottedOrNone> (NbConvertApp.postprocessor\_class)  
 Default: ''  
 PostProcessor class used to write the results of the conversion  
--output=<Unicode> (NbConvertApp.output\_base)  
 Default: ''  
 overwrite base name use for output files. can only be used when converting  
 one notebook at a time.  
--output-dir=<Unicode> (FilesWriter.build\_directory)  
 Default: ''  
 Directory to write output(s) to. Defaults to output to the directory of each  
 notebook. To recover previous default behaviour (outputting to the current  
 working directory) use . as the flag value.  
--reveal-prefix=<Unicode> (SlidesExporter.reveal\_url\_prefix)  
 Default: ''  
 The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,  
 but can be any url pointing to a copy 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.  
--nbformat=<Enum> (NotebookExporter.nbformat\_version)  
 Default: 4  
 Choices: [1, 2, 3, 4]  
 The nbformat version to write. Use this to downgrade notebooks.  
  
To see all available configurables, use `--help-all`  
  
Examples  
--------  
  
 The simplest way to use nbconvert is  
   
 > jupyter nbconvert mynotebook.ipynb  
   
 which will convert mynotebook.ipynb to the default format (probably HTML).  
   
 You can specify the export format with `--to`.  
 Options include ['asciidoc', 'custom', 'html', 'html\_ch', 'html\_embed', 'html\_toc', 'html\_with\_lenvs', 'html\_with\_toclenvs', 'latex', 'latex\_with\_lenvs', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'selectLanguage', 'slides', 'slides\_with\_lenvs'].  
   
 > jupyter nbconvert --to latex mynotebook.ipynb  
   
 Both HTML and LaTeX support multiple output templates. LaTeX includes  
 'base', 'article' and 'report'. HTML includes 'basic' and 'full'. You  
 can specify the flavor of the format used.  
   
 > jupyter nbconvert --to html --template basic 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  
  
./AC\_2/AC\_6\_S\_transmissionlines.ipynb

# note to self to work building book and github page[¶](#note-to-self-to-work-building-book-and-github-page)

build book: be looking at this folder not in it

jupyter-book build Python-and-SPICE-Book/

build book site: be inside the top repo

ghp-import -n -p -c https://pylcars.github.io/Python-and-SPICE-Book/docs -f \_build/html

In [ ]: