Jupman

A template manager for online books made with Jupyter notebooks and NBSphinx doc generator

People That Write a Lot

Sep 10, 2022

Copyright © 2022 by People That Write a Lot.

Jupman is available under the Creative Commons Attribution 4.0 International License, granting you the right to copy, redistribute, modify, and sell it, so long as you attribute the original to People That Write a Lot and identify any changes that you have made. Full terms of the license are available at:

http://creativecommons.org/licenses/by/4.0/

The complete book can be found online for free at:

https://jupman.softpython.org/en/latest/

CONTENTS

			1 2
1	Over	view	3
	1.1		3
	1.2		3
	1.3		4
2	Onial	kstart	5
4	2.1		5 5
	2.1		8
	2.3	E Company of the Comp	8
	2.4		9
	2.4		9
	2.3	Further steps	9
3	Editi	ng worksheets 1	1
	3.1	Common files	_
	3.2	Running Jupyter	1
	3.3	Source code for chapters	
	3.4	Exercise types	
	3.5	Solution tags	
	3.6	Directive tags	
	3.7	Hiding cells	6
	3.8	Utilities	7
	3.9	Launch unit tests	7
	3.10	Python Tutor	7
	3.11	Pandas	8
	3.12	Showing function help	9
	3.13	Custom js and css	9
	3.14	Show table of contents	9
4	Custo	omize 2	1
	4.1	Change website theme	
	4.2	Fonts	
	4.3	font sizes	
	4.4	Warning about old versions	
	4.5	Fix nbsphinx to create rst files	
	4.6	Git performance notes	
	4.0	on performance notes	_
5		ns management 23	
	5.1	What is an exam	5

	5.2	Exam commands	25
6	Chap	eter examples	29
	6.1	Python example	29
	6.2	Jupyter example	31
	6.3	Jupyter example with custom js and css	
	6.4	Jupyter and Python example	
	6.5	Big sub chapter 1	
	6.6	Big sub chapter 2	
	6.7	Example Challenge	
7	Temp		125
	7.1	Changelog	
	7.2	June 4th 2022 - 3.5	125
	7.3	Past Exams	129
	7.4	Exam project	129
	7.5	Project ideas	130
	7.6	Jupman Project	130
	7.7	Markdown	131
8	Tests		133
	8.1	Tests	
	8.2	Python tutor tests	146
n	D . C		1 = 1
7	Refer	rences	151

About

A template for online books made with Jupyter¹ notebooks and NBSphinx² doc generator.

Features:

- inherits generation of static websites from NBSphinx³, with search, PDF, EPUB
- builds with ReadTheDocs, Github Actions⁴, or local Docker emulating ReadTheDocs
- exercises generation from solution templates (both .ipynb and .py)
- · zips chapters
- code sharing among chapters (so students don't need to install dependencies)
- Python Tutor integration (offline, no required dependencies)
- · supports hundreds pages and deep nesting
- · decent PDF layout
- · basic exam management
- documentation and tests⁵
- requires Python 3.7+
- Apache License v2.0, open source code on Github⁶

Currently lacking:

- generated PDF always displays solutions⁷
- Python Tutor doesn't work in JupyterLab⁸
- more testing for exam management⁹

Used by:

- SoftPython book: english¹⁰ | italian¹¹ (both 1000+ pdf pages from jupyter notebooks)
- Scientific Programming Lab @ University of Trento, Data Science Master¹² (english, many python exercises and exams in mixed jupyter + python format)

. About 1

¹ http://jupyter.org

² http://nbsphinx.readthedocs.io/

³ http://nbsphinx.readthedocs.io/

⁴ https://github.com/DavidLeoni/readthedocs-to-actions

⁵ https://github.com/DavidLeoni/jupman/tree/master/_test

⁶ https://github.com/DavidLeoni/jupman

https://github.com/DavidLeoni/jupman/issues/87

⁸ https://github.com/DavidLeoni/jupman/issues/42

⁹ https://github.com/DavidLeoni/jupman/issues?q=is%3Aopen+is%3Aissue+label%3Aexams

¹⁰ https://en.softpython.org/

¹¹ https://it.softpython.org/

¹² https://sciprog.davidleoni.it/

Preface

This book is the result of ... We thank this and that ...

2 CONTENTS

CHAPTER

ONE

OVERVIEW

1.1 Contents

- 1. Quickstart
- 2. Editing worksheets
- 3. Customize
- 4. Exams management
- 5. Chapter examples
 - 1. Python example
 - 2. Jupyter example
 - 3. Mixed jupyter and python example
 - 4. Challenges: example worksheet solution¹³
- 6. Templates
 - 1. Past exams
 - 2. Changelog
- 7. Tests
 - 1. Jupman tests
 - 2. Python Tutor tests

1.2 Revisions

- 4 June 2022: Released v3.5
- 30 April 2022: Released v3.4
- 22 February 2022: Released v3.3
- 16 October 2020: Released v3.2
- 16 January 2020: Released v3.1
- 29 December 2019: Released v3.0
- 24 September 2018: Released v2.0

 $^{^{13}\} https://github.com/DavidLeoni/jupman/blob/master/challenge-example/example-chal-sol.ipynb$

- **3 August 2018**: Released v0.8
- Change log

1.3 Credits

• This site was made with Jupyter using NBSphinx extension¹⁴ and Jupman template¹⁵.

http://nbsphinx.readthedocs.io/http://jupman.readthedocs.io/

QUICKSTART

Jupman uses NbSphinx¹⁶ and either ReadTheDocs¹⁷ or Github Actions¹⁸

2.1 Installation

(Instructions are for Ubuntu, on Windows may differ)

First, on Github, fork as a template jupman project¹⁹ to create yours, for example my-project.

Then, on your computer, clone the my-project from Github

You can choose to build either on:

- ReadTheDocs
- · Github Actions
- · locally with plain Sphinx
- locally with RTD Docker²⁰

(Note Jupman itself is building on both ReadTheDocs and Github Actions only for testing purposes, one is enough)

2.1.1 Building with ReadTheDocs:

IMPORTANT: choose a name which is NOT already on ReadTheDocs²¹

Create a ReadTheDocs account²² **using the same name as in Github** so the address in readthedocs will be something like *my-project.readthedocs.org*.

- Use ReadTheDocs panels to link the project to your Github repository.
- In Admin-> Advanced settings panel, set:
 - Python interpreter to CPython 3.x
 - Requirements to requirements-build.txt

¹⁶ http://nbsphinx.readthedocs.io/

¹⁷ https://readthedocs.org

¹⁸ https://github.com/features/actions

¹⁹ https://github.com/DavidLeoni/jupman

²⁰ https://github.com/DavidLeoni/readthedocs-to-actions

²¹ http://readthedocs.org

²² http://readthedocs.org

2.1.2 Building with Github Actions:

Configure .github/workflows/main.yml²³ on your computer to your needs - you will need to:

- 1. at the beginning in the build_docs_job section there is an if which makes the workflow only work in DavidLeoni/jupman repository, change it with your project repo and comment the following needs: run_tests line
- 2. set RTD_PRJ_NAME
- 3. If you want to publish to Github Pages²⁴: everything is set, just create an empty branch gh-pages in an new HTML_FOLDER **before** committing from some other folder in your file system:

```
git clone YOUR_REPO_ADDRESS HTML_FOLDER
cd HTML_FOLDER
git checkout --orphan gh-pages
git rm -rf .
touch bla
git add .
git commit -m "init"
git push origin gh-pages
```

2.1.3 Local build with Sphinx

Assuming you are on Linux/Mac:

- 1) Install Python 3.7+
- 2) Install Jupyter²⁵
- 3) Install required modules:
- 3.a) In a virtual environment (**recommended**) from the root of the project, run:

```
./create-env.sh
```

This will automatically install required modules in _private/jupman_env using python3 system binary.

If you want to usa a particular python binary (note it must already be on your system), run i.e.:

```
./create-env.sh python3.7
```

Afterwards, to activate the environment run:

• In Windows:

_private\jupman_env\bin\activate

• in Linux/Mac:

```
source activate
```

to deactivate (from anywhere):

```
deactivate
```

²³ https://github.com/DavidLeoni/jupman/blob/master/.github/workflows/main.yml

²⁴ https://pages.github.com/

²⁵ http://jupyter.org/install.html

3.b) without a virtual environment (**not recommended**): From the root of the project, run:

```
python3 -m pip install --user -r requirements-build.txt
```

Warning: to have reproducible builds requirements—build.txt pinpoints a lot of dependencies, installing without virtual environment may mess up other python programs!

2.1.4 Optional - Running tests

To check everything is working, you may want to run the tests.

1. Install test dependencies:

```
python3 -m pip install --user -r _test/requirements-test.txt
```

2. Run the tests:

```
python3 -m pytest _test/*_test.py
```

2.1.5 Optional - Install Jupyter contrib extensions

For a better editing experience like having Table of contents and other things, do the following:

1. install the Jupyter contrib extensions²⁶ package:

If you have Anaconda:

```
conda install -c conda-forge jupyter_contrib_nbextensions
```

If you don't have Anaconda:

```
python3 -m pip install --user jupyter_contrib_nbextensions
```

2. Install in Jupyter:

```
jupyter contrib nbextension install --user
```

3. Enable extension:

For being able to view table of contents while editing notebooks, install toc2 extension:

```
jupyter nbextension enable toc2/main
```

For tocs to appear when in a document you will need to press a list button at the right-end of the toolbar.

(since Jupman 0.8 custom injected tocs are disabled by default)

4. For a nice GUI to install extensions, install the Jupyter Nbextensions configurator²⁷:

If you have Anaconda:

From Anaconda Prompt:

```
conda install -c conda-forge jupyter_nbextensions_configurator
```

If you don't have Anaconda:

2.1. Installation 7

²⁶ https://github.com/ipython-contrib/jupyter_contrib_nbextensions

²⁷ https://github.com/Jupyter-contrib/jupyter_nbextensions_configurator

```
python3 -m pip install --user jupyter_nbextensions_configurator
```

After installing, enable it:

```
jupyter nbextensions_configurator enable --user
```

and then start Jupyter, in file browser look for a Nbextensions tab

2.2 Configure

- 1. Edit conf.py²⁸ as needed, which is the configuration file for Sphinx. In particular, you **MUST** edit the sections marked with **TODO**
- 2. Try launching build:

```
python3 build.py
```

For more info, see related section

- 3. If everything works fine on your computer, push changes back to Github
- 4. Go back to ReadTheDocs and try to run a build. Hopefully your project will become available on something like *my-project.readthedocs.org*
- 5. If you want to grade exams, see *Exams management* page.

You should now be ready to create your notebooks by launching from the project root:

```
jupyter notebook
```

1. If you wish your notebooks to appear in the generated manual, you have to add them in the toc.rst file.

NOTE: the page toc-page.rst²⁹, which is set to be the master_doc of Sphinx, will just load the actual Table of Contents which is in toc.rst³⁰. It looks a bit convoluted because when it comes to indexes Sphinx is not much reliable, see this issue³¹. We strongly advise *not* to change these settings!

2. edit the home, which is in the index.ipynb³² file

2.3 Building the manual

For a quick build that only produces html:

```
python3 build.py -q
```

Site will be created in _build/ folder.

For help:

```
python3 build.py -h
```

²⁸ https://github.com/DavidLeoni/jupman/blob/master/conf.py

²⁹ https://github.com/DavidLeoni/jupman/blob/master/toc-page.rst

³⁰ https://github.com/DavidLeoni/jupman/blob/master/toc.rst

³¹ https://github.com/DavidLeoni/jupman/issues/11

³² https://github.com/DavidLeoni/jupman/blob/master/index.ipynb

To build everything (html + pdf + epub), go to the console and from the root directory run:

python3 build.py

NOTE: to generate PDFs you will need to install Latex environment

2.4 Publish

Just push to master and Github Actions / ReadTheDocs will do the rest, for example, for jupman is available at addresses:

- https://jupman.readthedocs.io/en/latest/
- https://davidleoni.github.io/jupman/en/latest/

IMPORTANT: ReadTheDocs WILL *NOT* execute Jupyter notebooks because of this bug³³

2.5 Further steps

See Editing worksheets, Customize and if needed Exams management pages

2.4. Publish 9

³³ https://github.com/DavidLeoni/softpython/issues/2

EDITING WORKSHEETS

Here we give an overview of how to edit worksheets. More info can be found in Jupman tests notebook

3.1 Common files

There are a bunch of files common to all worksheets and possibly website

You do not need to change them (except maybe my_lib.py)

File	Description	Jupyter	Website	Student zips
jupman_tools.py ³⁴	back end stuff	X	X	
jupman.py ³⁵	utilities for worksheets	X	X	X
my_lib.py ³⁶	custom utilities example	X	X	X
_static/js/jupman.js ³⁷	Javascript code	X	X	X
_static/css/jupman.css ³⁸	CSS	X	X	X
_static/css/jupman-web.css ³⁹	CSS		X	X

3.2 Running Jupyter

First of all, run Jupyter from the root directory:

jupyter notebook

 $^{^{34}\} https://github.com/DavidLeoni/jupman/blob/master/jupman_tools.py$

³⁵ https://github.com/DavidLeoni/jupman/blob/master/jupman.py

³⁶ https://github.com/DavidLeoni/jupman/blob/master/jupman.py

³⁷ https://github.com/DavidLeoni/jupman/blob/master/_static/js/jupman.js

³⁸ https://github.com/DavidLeoni/jupman/blob/master/_static/css/jupman.css

³⁹ https://github.com/DavidLeoni/jupman/blob/master/_static/css/jupman-web.css

3.3 Source code for chapters

Put chapters one per folder, in the root. Any folder which doesn't start with underscore _ or exam/ will be considered a chapter.

During build, each chapter gets automatically zipped and zip goes to _static/generated. So for example, python-example/ produces a zip called _static/generated/python-example.zip, which will have these contents:

```
python-example
   _static
    js
        jupman.js
        toc.js
    css
        jupman.css
    img
        cc-by.png
    python-example.ipynb
    lab.py
    lab_test.py
    lab_sol.py
    jupman.py
    my_lib.py
```

The zip folder structure will be a merge of chapter files and files shared by all chapters which are specified in <code>exer-cises_common_files</code> variable in conf.py. Since the root in the zip becomes the chapter itself, jupman will process .py and .ipynb files for fixing eventual relative imports. Markdown, HTML and CSS links in ipynb will also be adjusted.

Exercise files can be automatically generated from solutions, as we will see next.

3.4 Exercise types

There can be three kinds of exercises: exercises in Python files, exercises in Jupyter files and mixed jupyter and Python exercises.

You can automatically generate an exercise from a solution file by stripping text marked with special tags. You can inspect generated files in _build/jupman/directory

On the website, students will be able to see solutions by clicking on appropriate buttons.

In the zips to download, two versions of files will provided, one without solution and one with solutions (in exam modality of course no solution will be shipped)

3.4.1 Exercises in Python files

See python-example/python-example.ipynb

In this type of exercises, typically you have a Jupyter file (like python-example.ipynb) that describes the exercise and then the actual exercises are in Python files.

If there is a solution file FILE_sol.py ending in _sol.py but no corresponding exercise file FILE.py without the sol:

then Jupman will try to generate FILE.py one from FILE_sol.py. To do so, it will look for tags to strip inside the solution file.

If there is already an exercise file like this:

```
• python_intro.py
```

• python_intro_sol.py

Jupman will just copy the existing file.

3.4.2 Exercises in Jupyter files

See example: jupyter-example/jupyter-example-sol.ipynb

This type of exercises stay in a Jupyter notebook itself.

If there is a notebook ending in -sol.ipynb, the following applies (**WARNING**: for ipynb files we use dash -, *not* the underscore _):

- the notebook must contain tags to strip
- exercises derived will have 'EXERCISES' appended to the title (the word can be customized in conf.py you might need to translate it)

3.4.3 Mixed exercises in Jupyter and Python files

See jup-and-py-example/jup-and-py-example-sol.ipynb

3.4.4 Challenges

This is an experimental feature, current implementation is subject to change.

Challenges are solutions which remain unpublished and from which exercises are generated in the same original older where the solution resides (not only in the zip!). Challenge files can be both Jupyter notebooks or Python files, ending in -chal-sol.ipynb or _chal_sol.py.

The idea is that challenges solutions are gitignored, and exercises are manually generated by calling jupman. generate_exercise() inside a Jupyter notebook like this:

```
#jupman-purge
import sys; sys.path.append('../'); from conf import jm;
jm.generate_exercise('great_chal_sol.py')
#/jupman-purge
```

It is a bit laborious but the idea is that typically you will also want to run and see tests results in Jupyter notebook so you can do it in the same final cell, which you will also probably want to set in cell metadata "nbsphinx": "hidden"

• the solution notebook must contain tags to strip and have SOLUTIONS at the end of the title (the word can be customized in conf.py - you might need to translate it)

3.5 Solution tags

Presence of these tags marks a cell as a solution.

Start tags begin with a # while end tags begin with a #\

3.5.1 jupman-raise

Replaces code inside with an Exception (text is customizable in conf.py). Be careful to position the comment exactly with the indentation yuoi want the raise to appear. For example:

```
def add(x,y):
    #jupman-raise
    return x + y
    #/jupman-raise
```

becomes

```
def add(x,y):
    raise Exception('TODO IMPLEMENT ME !')
```

3.5.2 jupman-strip

Just strips code inside exercises

```
def f(x):
    print(x)

#jupman-strip
def help_func(x,y):
    return x - y
#/jupman-strip

def g(y):
    return y
```

becomes

```
def f(x):
    print(x)

def g(y):
    return y
```

3.5.3 write here

This special tag for python code erases whatever is found afterwards the # write here line

- you can put how many spaces you want in the comment
- phrase can be customized in conf.py

```
w = 5
# write here fast please
x = 5 + w
y = 2 + x
```

becomes

```
w = 5
# write here fast please
```

3.5.4 SOLUTION

In a code cell, if you put # SOLUTION at the beginning the whole cell cell content gets deleted (# SOLUTION string included).

• Word can be customized in conf.py

```
# SOLUTION

def f():
    print('hello')
```

becomes nothing:

[]:

3.5.5 QUESTION - ANSWER

In a markdown cell, everything in a cell with **ANSWER**: inside will be stripped.

• Markdown can be customized in conf.py

QUESTION: Describe why iPhone n+1 is better than iPhone n

ANSWER: it costs more

Becomes:

QUESTION: Describe why iPhone n+1 is better than iPhone n

[]:

3.5. Solution tags

3.6 Directive tags

Some tags change the preprocessor behaviour. They are applied before solution tags.

3.6.1 jupman-purge

Eliminate content both from exercises AND solutions. Can be helpful when you have code which creates expected output, like images or python data - the idea is to completely remove code so so students don't accidentally copy-paste it or uncomment it.

- jupman-purge-input: purges only cell source
- jupman-purge-output: purges only cell output
- jupman-purge-io: purges both input and output

jupman-purge purges only a span:

```
x=5
#jupman-purge
plt.savefig('expected_image.png')
jupman.save_py('expected_output_db.py', ['big', 'data', 'structure']*1000)
#/jupman-purge
x=6
```

becomes

```
x=5
x=6
```

3.6.2 jupman-preprocess

By default only notebooks solutions (ending in -sol.ipynb) are preprocessed before html conversion begins. If you want to force preprocessing on a particular non-solution notebook, add this in the first cell:

```
#jupman-preprocess
```

3.7 Hiding cells

A way to hide cells (like for example the import jupman code) is by clicking View->Cell toolbar -> Edit metadata and adding "nbsphinx": "hidden" to the JSON (see also original NBSphinx docs⁴⁰ and Togglable cells in Jupman tests).

NOTE 1: As of NBSphinx 2.17, it is not possible to hide only cell text but not the output.

⁴⁰ https://nbsphinx.readthedocs.io/en/0.2.14/hidden-cells.html#Hidden-Cells

3.8 Utilities

NOTE: not mandatory, it's mostly intended to tweak notebooks downloaded locally. Should be avoided in notebooks meant for students, as it's more likely it will mess their configurations - also, they might copy the notebooks without knowing they contain the custom js and use weird extensions which could generate conflicts (such as double toc)

For notebooks in the root folder:

```
import jupman
```

Worksheets in in subfolders can use sys.path to locate the module

```
import sys
sys.path.append('../')
import jupman
```

Some reason for this ugliness is reported in this issue⁴¹.

3.9 Launch unit tests

Inside worksheets you can run unittest tests.

To run all the tests of a test class, write like this

```
jupman.run(NameOfTheTestClass)
```

To run a single method, write like this:

```
jupman.run(NameOfTheTestClass.nameOfTheMethod)
```

3.10 Python Tutor

Among the various ways to embed Python Tutor, we decided to implement a special jupman.pytut() method. First you need to import the jupman module:

```
[1]: import sys
sys.path.append('../')
import jupman
```

Then you can put a call to jupman.pytut() at the end of a cell, and the cell code will magically appear in python tutor in the output (except the call to pytut() of course). To see Python Tutor you don't need to be online

```
[2]: x = [5,8,4]
y= {3:9}
z = [x]
jupman.pytut()

[2]: <IPython.core.display.HTML object>
```

3.8. Utilities 17

 $^{^{41}\} https://github.com/DavidLeoni/jupman/issues/12$

Beware of variables which were initialized in previous cells, they won't be available in Python Tutor and you will get an error:

```
[3]: w = 8

[4]: x = w + 5
    jupman.pytut()

Traceback (most recent call last):
    File "../jupman.py", line 2443, in _runscript
        self.run(script_str, user_globals, user_globals)
    File "/usr/lib/python3.7/bdb.py", line 578, in run
        exec(cmd, globals, locals)
    File "<string>", line 2, in <module>
    NameError: name 'w' is not defined

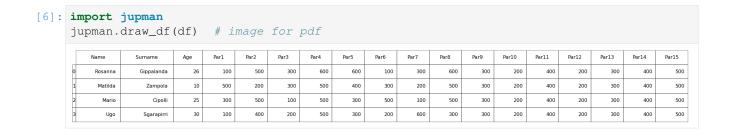
[4]: <IPython.core.display.HTML object>
```

3.11 Pandas

Correctly rendering pandas in PDFs is not so easy (see issue⁴²), so far we created this little function which sometimes is handy:

```
[5]: import pandas as pd
    lista = [['Rosanna', 'Gippalanda', 26, 100, 500, 300, 600, 600, 100, 300, 600, 300, __
    \Rightarrow200, 400, 200, 300, 400, 500],
            ['Matilda', 'Zampola',
                                     10, 500, 200, 300, 500, 400, 300, 200, 500, 300,
    \Rightarrow200, 400, 200, 300, 400, 500],
            ['Mario', 'Cipolli',
                                     25, 300, 500, 100, 500, 300, 500, 100, 500, 300,
    \Rightarrow200, 400, 200, 300, 400, 500],
            ['Ugo', 'Sgarapirri',
                                     30, 100, 400, 200, 500, 300, 200, 600, 300, 300,
    →200, 400, 200, 300, 400, 500]
           ]
    df = pd.DataFrame(lista, columns =['Name', 'Surname', 'Age', *['Par'+str(i) for i in_
    →range(1,16)]])
    df # web
[5]:
                  Surname Age Par1 Par2 Par3 Par4 Par5 Par6 Par7 Par8 \
    0 Rosanna Gippalanda 26 100 500 300 600 600 100 300
                                                                      600
    1 Matilda
                Zampola 10
                               500
                                      200 300 500
                                                      400
                                                             300 200 500
    2.
        Mario
                  Cipolli 25
                               300
                                      500
                                           100
                                                500
                                                      300 500 100 500
                                     400
         Ugo Sgarapirri 30
                               100
                                          200
                                                      300 200 600
                                                                        300
    3
                                                500
       Par9 Par10 Par11 Par12 Par13 Par14 Par15
       300
              200
                    400
                          200
                                 300
                                        400
        300
              200
                     400
                            200
                                  300
                                         400
                                                500
    1
                                  300
                                        400
    2
        300
              200
                     400
                           200
                                               500
    3
       300
              200
                     400
                           200
                                  300
                                         400
                                               500
```

⁴² https://github.com/DavidLeoni/jupman/issues/69



3.12 Showing function help

Python help is already quite good, but adds two useless extra lines and only works as a print, so we defined jupman.get_doc:

```
[7]: print(jupman.get_doc(jupman.get_doc))

def get_doc(fun):
    """ Returns the help of a function formatted in a faithful manner

    @since 3.3
"""
```

3.13 Custom js and css

If you need custom is and/or css in a notebook, you can inject it by running

```
jupman.init()
```

in the first cell, it will inject jupman.js and jupman.css

3.14 Show table of contents

Since 0.8, custom toc is disabled, try instead *installing toc2 extension*. If you still want the jupman toc (not recommended), execute

```
jupman.init(toc=True)
```

it will create the sidebar even when editing in Jupyter. To refresh the sidebar, just rerun the cell.

Note: hiding the jupman.init code cell will prevent the build system to embed the Javascript file jupman.js inside the page in the HTML website: this is still fine as it is fetched separately by settings in conf.py.

CHAPTER

FOUR

CUSTOMIZE

4.1 Change website theme

If you want to change site colors and other changes, copy/edit $_static/css/jupman-web.css^{43}$ and set it in conf html $_css_files$:

```
html_css_files = [
    'css/jupman.css',  # shared among jupyter and website
    'css/jupman-web.css',  # only on website
    #'css/softpython-theme.css',  # uncomment to activate
    #'css/scifi-theme.css',
]
```

4.2 Fonts

Fonts are a bit of a complex topic

TODO this part is just a collection of personal notes

- The missing guide to font formats⁴⁴
- https://docs.readthedocs.io/en/latest/guides/adding-custom-css.html
- RTD Code font issue on github⁴⁵

Tools:

Comprehensive article: https://www.useragentman.com/blog/2011/02/20/converting-font-face-fonts-quickly-in-any-os/and https://www.useragentman.com/blog/the-css3-font-converter/

https://github.com/zoltan-dulac/css3FontConverter

woff2

https://github.com/google/woff2

sfnt2woff

```
sudo apt-get install libbrotli-dev
sfnt2woff SomeFont.otf
```

mkeot

⁴³ https://github.com/DavidLeoni/jupman/blob/master/_static/css/jupman-web.css

⁴⁴ https://creativemarket.com/blog/the-missing-guide-to-font-formats

⁴⁵ https://github.com/readthedocs/sphinx_rtd_theme/issues/524

```
sudo apt-get install eot-utils
mkeot SomeFont.otf > SomeFont.eot
```

or https://github.com/wget/ttf2eot

FontForge (GUI and scriptable)

sudo apt-get install fontforge

4.3 font sizes

https://www.24a11y.com/2019/pixels-vs-relative-units-in-css-why-its-still-a-big-deal/

https://chiamakaikeanyi.dev/sizing-in-css-px-vs-em-vs-rem/

4.4 Warning about old versions

ReadTheDocs has a mechanism⁴⁶ to warn the user if he's looking at an old version of the site, but we found it doesn't work much for course-based documentation. So for versioning we think it's better to adopt a mixed git branch / tags devlopment model, and we added a template warning to show in old branches. To enable it in an old branch, just rename _templates/breadcrumbs.html.bak into _templates/breadcrumbs.html and edit as needed.

4.5 Fix nbsphinx to create rst files

Sometimes nbsphinx does not report properly RST conversion errors (see bug^{47}). As a hacky workaround, you might take the nbsphinx.py from ~/.local/lib/python3.5/site-packages/, make a copy of it in your project home and patch it like this 48 When you call sphinx, it will generate RST files in _build/jupman-rst/.

Of course, things can be cleaner using a virtual env with venv⁴⁹

4.6 Git performance notes

Current suggested setup for hosting on Github is creating branch gh-pages and using Github Actions to populate it with html, zips, pdf and epub files. While keeping all that stuff versioned may seem pretty inefficient, apparently git is pretty good⁵⁰ at compressing binary files

The size of .git repo for a 1000 pdf page project SoftPython with 300 commits and 100 MB of code is:

```
.git: 183 MB
```

By truncating gh-pages to last commit and garbage collecting, we get:

```
.git: 139 MB
```

If we completely remove gh-pages branch, we get:

 $^{^{46}\} https://docs.readthedocs.io/en/latest/versions.html$

⁴⁷ https://github.com/DavidLeoni/jupman/issues/9

⁴⁸ https://github.com/DavidLeoni/jupman/commit/0f332629ce4e2b0186c954c55aea7fa67992ace9#diff-bd3d9c4d2e80ed83fd2443d1301aa65bR649

⁴⁹ https://docs.python.org/3/library/venv.html

⁵⁰ https://stackoverflow.com/a/48305739

.git: 68.7 MB

So gh-pages size is:

one commit: 70.3 MB300 commits: 114.3 MB

which is not even double than source code git size.

If the repo gets really huge, in order to to shrink it some git knowledge is required.

If the repo is served from another server and you want to truncate that server git repo:

On that server console:

1. first make sure you are on gh-pages branch:

git checkout gh-pages

2. truncates previous commits:

```
git fetch --depth=1 origin gh-pages
```

3. removes various links around which may still point to old commits:

```
git reflog expire --expire-unreachable=now --all
```

4. actually deletes from disk old commits:

```
git gc --aggressive --prune=all
```

Note the result of truncation cannot be pushed back to origin as git would complain it is a shallow branch.

[]:

CHAPTER

FIVE

EXAMS MANAGEMENT

Jupman comes with a script to manage exams called exam.py, which allows to manage the full cycle of an exam.

5.1 What is an exam

Exam text is represented as Jupyter notebooks, which are taken from _templates/exam/solutions/exam-yyyy-mm-dd.ipynb

Exercises for students: they are supposed to be the exam notebook itself and / or plain python files (or the notebook itself) plus unittests and relative solutions.

Marks spreadsheet: By default there is also an LibreOffice spreadsheet to give marks, in case you need it.

When you initialize an exam with the init command, for example for date 2000-12-31, all the presets in _templates/exam/ are copied to private/2000-12-31/ and private/2000-12-31/solutions. Presets can be changed at will to suit your needs. When packaging, student zip is assembled in private/2000-12-31/student-zip

System is flexible enough so you can privately work on next exams in private/ folder and still being able to publish modifications to main website. After an exam, you can copy the private exam to the public folders in past-exams/.

5.2 Exam commands

To see the help:

```
python3 exam.py -h
```

To see help for a particular subcommand, like i.e. init, type the subcommand followed by -h:

```
python3 exam.py init -h
```

Running commands should be quite self-explanatory.

NOTE: as of today (Dec 2019) software may contain bugs, but at least we check for major misuses (like trying to overwrite existing exams).

In the file create-exam-example.sh there is a typical run of the script, which creates the example exam for date 2000-12-31. Notice it might ask you to delete the existing 2000-12-31 exam, if it does just follow the instructions. Here's the output:

```
> ./create-exam-example.sh
python3 exam.py init 2000-12-31
 Detected release from git: 3.2.0-3-g30a995c
No GOOGLE_ANALYTICS environment variable was found, skipping it
 You can now edit Python solutions, tests, exercises and exam notebook here
    _private/2000-12-31/solutions
 DONE.
python3 exam.py package 2000-12-31
 Detected release from git: 3.2.0-3-g30a995c
No GOOGLE_ANALYTICS environment variable was found, skipping it
 Cleaning _private/2000-12-31/server/jupman ...
 Copying exercises to _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-
\hookrightarrowLASTNAME-ID/
 Copying code
    from _private/2000-12-31/solutions
         _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-ID/
   Writing (patched) _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-
→LASTNAME-ID/exam-2000-12-31.ipynb
   Generating _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-
→ID/trees.py
   Writing _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-ID/
→example.txt
   Generating _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-
→ID/lists.pv
   Writing (patched) _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-
→LASTNAME-ID/trees_test.py
   Writing (patched) _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-
→LASTNAME-ID/lists_test.py
 Creating dir _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-
   Writing _private/2000-12-31/student-zip/jupman-2000-12-31-FIRSTNAME-LASTNAME-ID/
→img/mountains.jpg
 Building pdf ..
 Creating student exercises zip: _private/2000-12-31/server/jupman-2000-12-31-exam.
 Writing jupman.py
 Writing my_lib.py
 Writing _static/img/cc-by.png
 Writing _static/js/jupman.js
 Writing _static/css/jupman.css
 Writing _static/js/toc.js
 Writing _static/js/pytutor-embed.bundle.min.js
 Wrote _private/2000-12-31/server/jupman-2000-12-31-exam.zip
 DONE.
 ----- Simulating some shipped exams...
mkdir -p _private/2000-12-31/shipped/john-doe-112233
cp _templates/exam/solutions/lists_sol.py _templates/exam/solutions/lists_test.py _
→templates/exam/solutions/trees_sol.py _templates/exam/solutions/trees_test.py _
→private/2000-12-31/shipped/john-doe-112233
mkdir -p _private/2000-12-31/shipped/jane-doe-445566
```

(continues on next page)

(continued from previous page)

```
cp _templates/exam/solutions/lists_sol.py _templates/exam/solutions/lists_test.py _
→templates/exam/solutions/trees_sol.py _templates/exam/solutions/trees_test.py _
→private/2000-12-31/shipped/jane-doe-445566
----- Done with shipped exams simulation, time to grade ...
python3 exam.py grade 2000-12-31
 Detected release from git: 3.2.0-3-g30a995c
No GOOGLE_ANALYTICS environment variable was found, skipping it
 Copying Python files to execute and eventually grade in _private/2000-12-31/graded/
⇒john-doe-112233/graded
 Copying original shipped files (don't touch them!) in _private/2000-12-31/graded/
→john-doe-112233/shipped
 Copying Python files to execute and eventually grade in _private/2000-12-31/graded/
→jane-doe-445566/graded
 Copying original shipped files (don't touch them!) in _private/2000-12-31/graded/
⇒jane-doe-445566/shipped
 DONE.
python3 exam.py zip-grades 2000-12-31
 Detected release from git: 3.2.0-3-g30a995c
No GOOGLE_ANALYTICS environment variable was found, skipping it
 You can now find zips to send to students in _private/2000-12-31/graded
 DONE.
python3 exam.py publish 2000-12-31
 Detected release from git: 3.2.0-3-g30a995c
No GOOGLE_ANALYTICS environment variable was found, skipping it
 Copying solutions to exams/2000-12-31/solutions
 Copying exam PDF text
 Exam Python files copied.
 You can now manually build and run the following git instructions to publish the
   ./build.py
   git status # just to check everything is ok
   git commit -m 'published 2000-12-31 exam'
   git push
 DONE.
 Finished example exam run !!
```

CHAPTER

SIX

CHAPTER EXAMPLES

6.1 Python example

6.1.1 Download exercises zip

Browse files online⁵¹

Example of notebook for exercises in Python files

6.1.2 What to do

• unzip exercises in a folder, you should get something like this:

```
python-example
    python-example.ipynb
    lab1.py
    lab1_test.py
    lab1_sol.py
    lab2.py
    lab2_test.py
    lab2_test.py
    lab2_sol.py
    jupman.py
    my_lib.py
```

- open the editor of your choice (for example Visual Studio Code, Spyder or PyCharme), you will edit the files lab1.py and lab2.py
- Go on reading this notebook, and follow instuctions inside.

Let's begin

You are going to program a simulator of bouncing clowns. To do so, we are going to load this module:

```
[2]: import local

[3]: local.gimme(5)

It was a 5 indeed
```

 $^{^{51}\} https://github.com/DavidLeoni/jupman/tree/master/python-example$

Download test data

Local file:

- example.txt
- example.csv

6.1.3 Global image



6.1.4 Local exercise image



6.1.5 Python tutor

```
[4]: x = [1,2,3]
y = 6

jupman.pytut()

[4]: <IPython.core.display.HTML object>

[5]: y = [1,2,3]
jupman.pytut()

[5]: <IPython.core.display.HTML object>
```

Start editing lab1.py in text editor

```
[6]: from lab1_sol import *
```

6.1.6 add

Implement add function:

```
[7]: add(3,5)
[7]: 8
```

6.1.7 sub

Implement sub function

```
[8]: sub(7,4)
[8]: 3
```

6.2 Jupyter example

6.2.1 Download exercises zip

Browse files online⁵²

Example of notebook for exercises in Jupyter files.

For python files based example and more, see Python example

6.2.2 What to do

• unzip exercises in a folder, you should get something like this:

```
jupyter-example
  jupyter-example.ipynb
  jupyter-example-sol.ipynb
  jupman.py
  my_lib.py
```

WARNING: to correctly visualize the notebook, it MUST be in an unzipped folder!

- open Jupyter Notebook from that folder. Two things should open, first a console and then browser. The browser should show a file list: navigate the list and open the notebook jupyter-example/jupyter-example. ipynb
- · Go on reading that notebook, and follow instuctions inside.

⁵² https://github.com/DavidLeoni/jupman/tree/master/jupyter-example

Shortcut keys:

- to execute Python code inside a Jupyter cell, press Control + Enter
- to execute Python code inside a Jupyter cell AND select next cell, press Shift + Enter
- to execute Python code inside a Jupyter cell AND a create a new cell aftwerwards, press Alt + Enter
- If the notebooks look stuck, try to select Kernel -> Restart

6.2.3 Exercise 1

[4]: <IPython.core.display.HTML object>

Implement inc function:

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
def helper(x):
    return x + 1

def inc(x):
    return helper(x)
```

</div>

```
[5]:
    def inc(x):
        raise Exception('TODO IMPLEMENT ME !')
```

6.2.4 Exercise 2

Implement upper function

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
def helper2(x):
    return x.upper()

def upper(x):
    return helper2(x)
```

</div>

```
[6]: def upper(x):
    raise Exception('TODO IMPLEMENT ME !')
```

Exercise 3

Note everything after the 'write here' comment will be discarded. Note you can put how many spaces you want in the comment

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
[7]:
    w = 5
# write here

x = 5 + 6
y = 6.4
z = x / y
```

</div>

```
[7]:
w = 5
# write here
```

Exercise 4

Shows how to completely remove the content of a solution cell (including the solution comment)

EXERCISE: write a function that prints 'hello'

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
[8]: # SOLUTION

def f():
    print('hello')

</div>
[8]:
```

Exercise 5

Shows the QUESTION / ANSWER feature. All content in 'ANSWER:' cell will be stripped

QUESTION: Describe why iPhone n + 1 is better than iPhone n

Show answer<div class="jupman-sol jupman-sol-question" style="display:none">

ANSWER: it costs more

</div>

6.2.5 Conclusion

bla bla

Relative image test, Markdown format:



Relative image test, HTML img tag:



Relative link test, Markdown format:

Back to index

Relative link test, HTML a tag:

Back to index

[]

6.3 Jupyter example with custom js and css

6.3.1 Download exercises zip

Browse files online⁵³

Example of notebook for exercises in Jupyter files which calls jupman.init() for injecting jupman.js and jupman.css while editing in Jupyter.

Note calling init () is **not** mandatory. See also Common files⁵⁴ and Custom-js-and-css⁵⁵.

6.3.2 Exercise 1

Implement inc function:

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
def helper(x):
    return x + 1

def inc(x):
    return helper(x)
```

</div>

```
[2]:
    def inc(x):
        raise Exception('TODO IMPLEMENT ME !')
```

6.3.3 Exercise 2

Implement upper function

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
def helper2(x):
    return x.upper()

def upper(x):
    return helper2(x)
```

</div>

⁵³ https://github.com/DavidLeoni/jupman/tree/master/jupyter-js-example

⁵⁴ https://jupman.softpython.org/en/latest/manual/editing.html#Common-files

⁵⁵ https://jupman.softpython.org/en/latest/manual/editing.html#Custom-js-and-css

```
[3]: def upper(x):
    raise Exception('TODO IMPLEMENT ME !')
```

6.4 Jupyter and Python example

6.4.1 Download exercises zip

Browse files online⁵⁶

Most complex example of a notebook with exercises both in Jupyter and Python files, and 'advanced' features

6.4.2 What to do

• unzip exercises in a folder, you should get something like this:

```
jup-and-py-example
  jup-and-py-example.ipynb
  jup-and-py-example_sol.ipynb
  lab.py
  lab_test.py
  lab_sol.py
```

- open the editor of your choice (for example Visual Studio Code, Spyder or PyCharme), and edit lab.py file
- Go on reading this notebook, and follow instuctions inside.

Let's begin

You are going to program a simulator of bouncing clowns. To do so, we are going to load this module:

```
[2]: import local

[3]: local.gimme(5)

It was a 5 indeed
```

Download test data

Local file:

- · example.txt
- example.csv

 $^{^{56}\} https://github.com/DavidLeoni/jupman/tree/master/jup-and-py-example$

6.4.3 Global image



6.4.4 Local exercise image



6.4.5 Python tutor

```
[4]: x = 5
y = 6
z = x + y
jupman.pytut()
[4]: <IPython.core.display.HTML object>
```

6.4.6 Exercise in Jupyter

Implement this function:

Show solution<div class="jupman-sol jupman-sol-code" style="display:none">

```
[5]: def hello(s):
    return ['hello',s]*1000

hello_db = hello("Guybrush")
    (continues on next page)
```

(continued from previous page)

```
hello_db[:10]

['hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush']
```

</div>

```
[5]: def hello(s):
    raise Exception('TODO IMPLEMENT ME !')

hello_db = hello("Guybrush")

hello_db[:10]

[5]: ['hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush']
```

Full expected output is in file expected_output_db.py, if you can't manage to solve the exercise, as a last resort you can type: from expected_hello_db import * (DO NOT copy-paste file content, it would probably mess Jupyter up)

```
[6]: from expected_hello_db import *
    expected_hello_db[:10]

[6]: ['hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush',
    'hello',
    'Guybrush']
```

Other example:

```
[7]: hello_db2 = hello("Threepwood")
hello_db2[:10]
```

```
[7]: ['hello',
    'Threepwood',
    'hello',
    'Threepwood',
    'hello',
    'Threepwood',
    'hello',
    'Threepwood',
    'hello',
    'Threepwood',
    'hello',
    'Threepwood']
```

6.4.7 Exercise using previous output

Write some code which says hello 3 times using previous functionand

```
[8]: print(hello('Guybrush')[:6])
['hello', 'Guybrush', 'hello', 'Guybrush']
```

6.4.8 Question in Jupyter

QUESTION: Why learn coding?

Show answer<div class="jupman-sol jupman-sol-question" style="display:none">

ANSWER: So they pay me more

```
x + 1
```

Some other comment

```
Some nasty formatting

even more formatting
```

</div>

6.4.9 Exercise in Python

Start editing lab.py in text editor

```
[9]: from lab_sol import *
```

6.4.10 add

Implement add function:

```
[10]: add(3,5)
[10]: 8
```

6.4.11 sub

Implement sub function

```
[11]: sub(7,4)
[11]: 3
```

6.4.12 Fine grained purging

```
This cell input will be completely removed
```

```
[13]:

print("This cell output will be completely removed")
```

6.5 Big sub chapter 1

6.5.1 Big docs example 1

reasonable paragraph

Bla bla

reasonable sub paragraph

Bla bla

Reasonable subsub paragraph

Bla bla

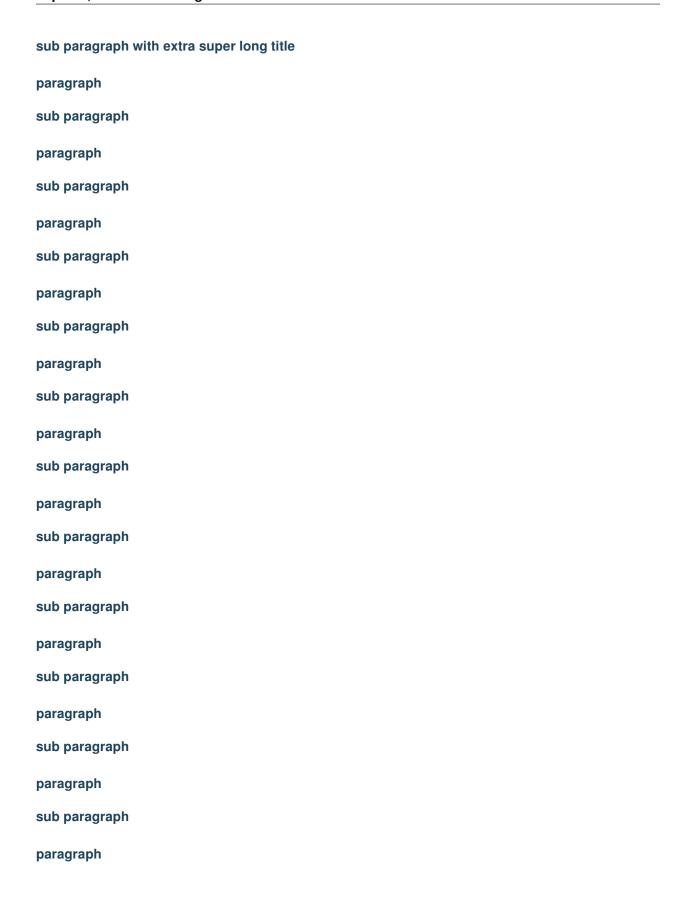
reasonable paragraph Bla bla reasonable sub paragraph Bla bla Reasonable subsub paragraph Bla bla paragraph with long title Bla bla sub paragraph with long title Bla bla paragraph with long title sub paragraph with long title subsub paragraph with long title paragraph with long title sub paragraph with long title paragraph with extra super long title sub paragraph with extra super long title sub sub paragraph with extra super long title paragraph with extra super long title

sub paragraph with extra super long title

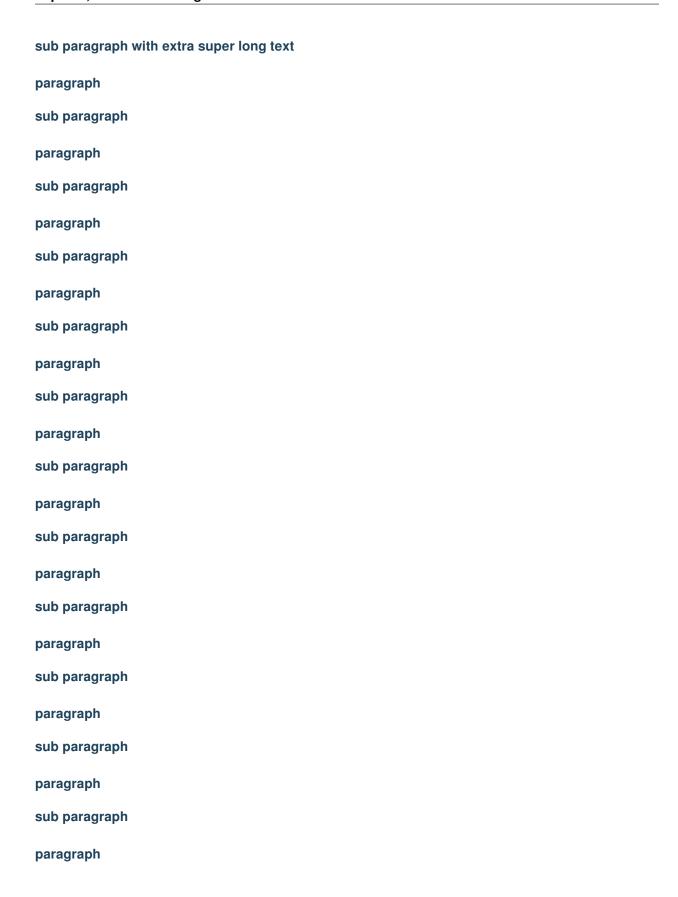
sub paragraph with extra super long title

paragraph with extra super long title

paragraph with extra super long title



sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text

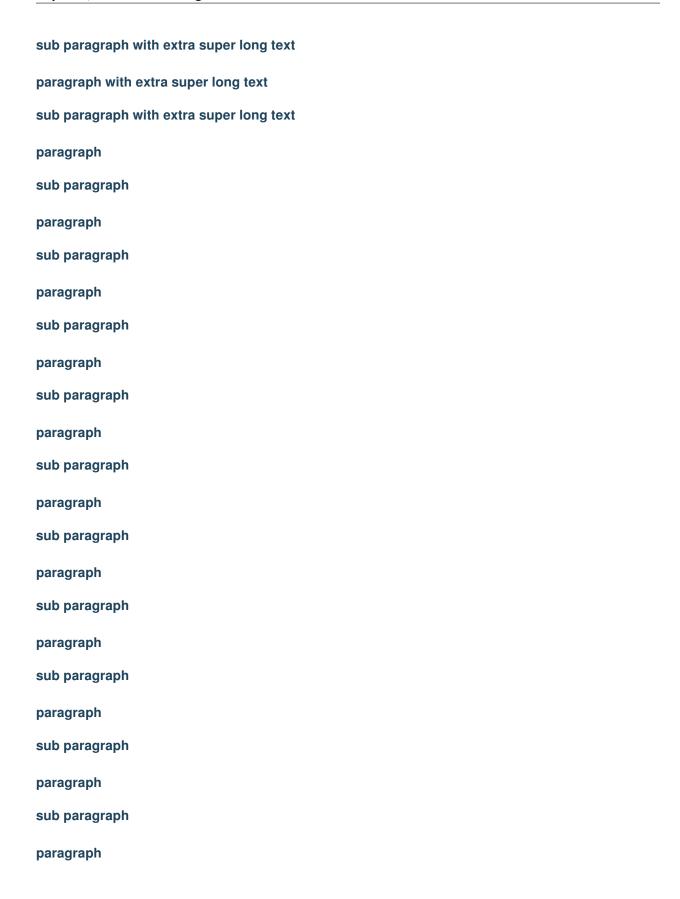


sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph

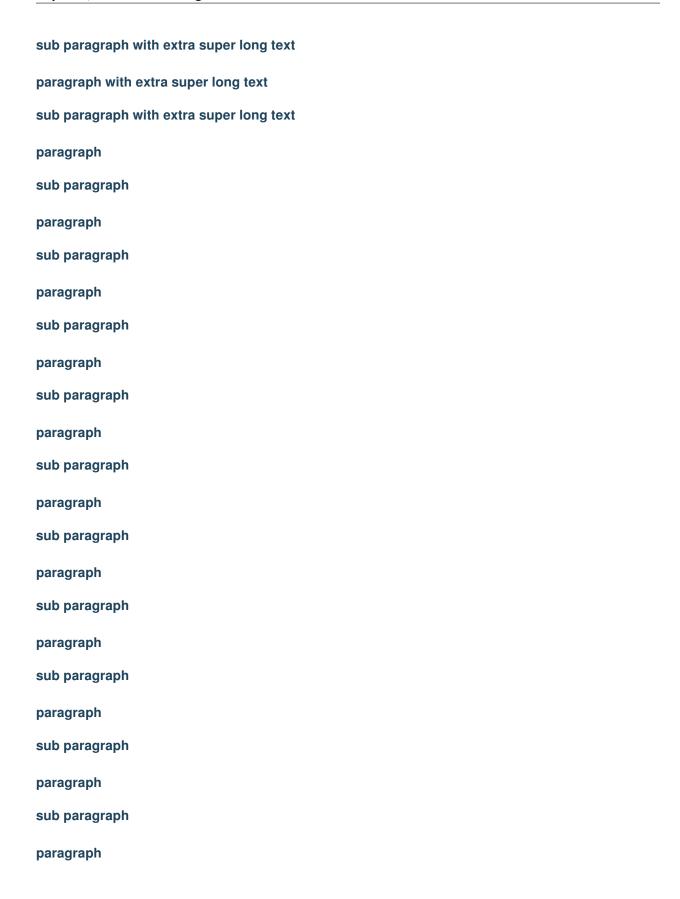
[]:

6.5.2 Big docs example 2

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
sub paragraph with extra super long text



sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text

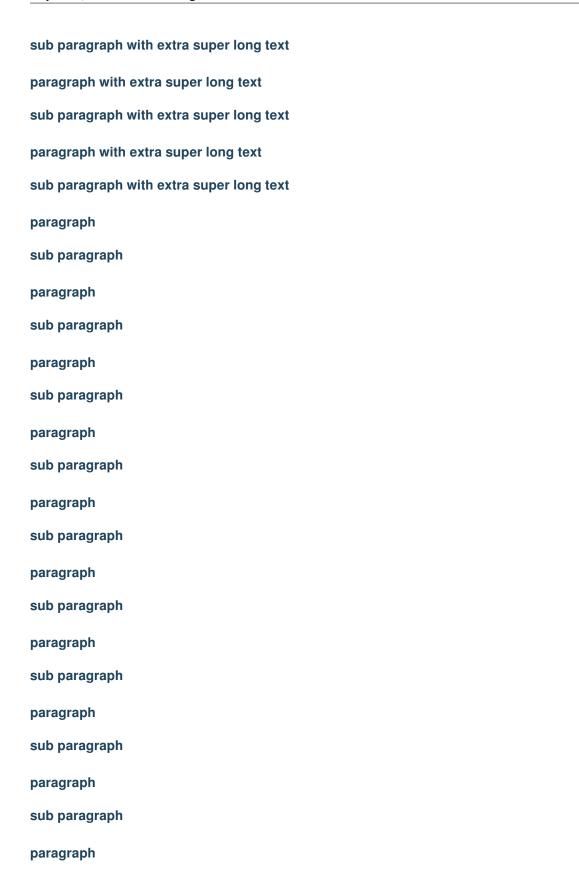


sub paragraph
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
sub paragraph

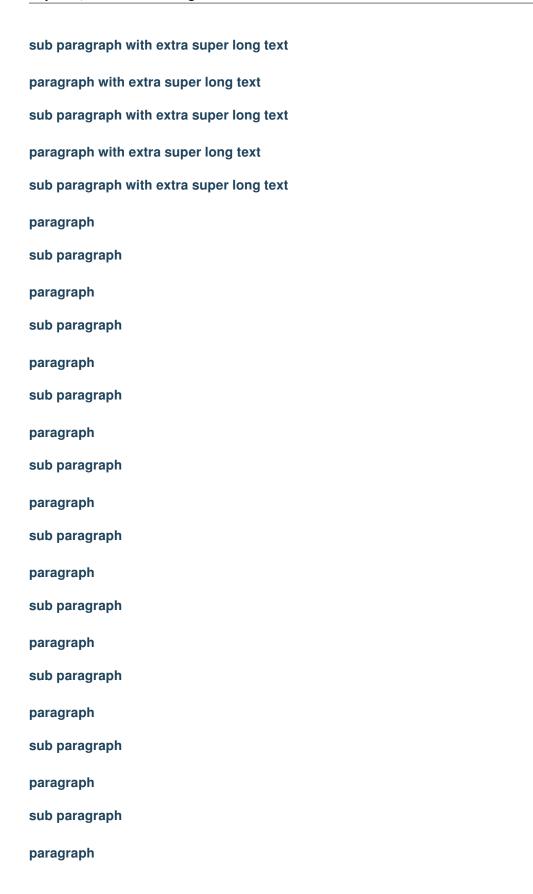
1.

6.5.3 Big docs example 3

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
sub paragraph with extra super long text
sub paragraph with extra super long text
paragraph with extra super long text



sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text



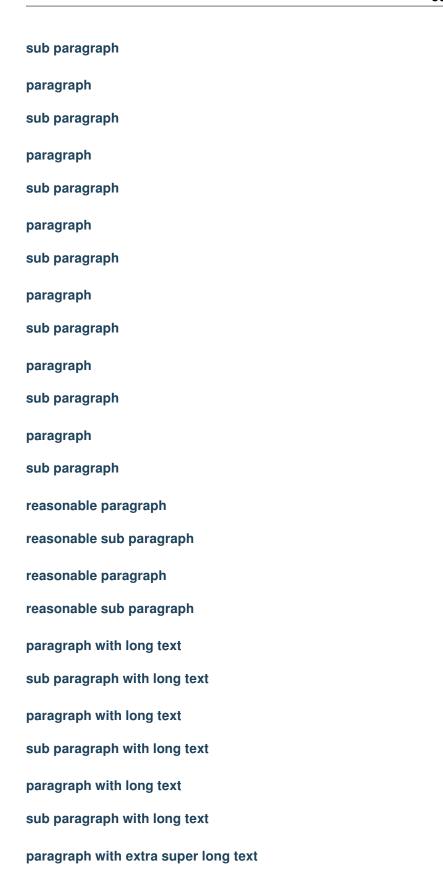
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph

[]:

6.5.4 Big docs example 4

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text

sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph
paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph

[]:

6.5.5 Big docs example 5

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
paragraph with long text

sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text

sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

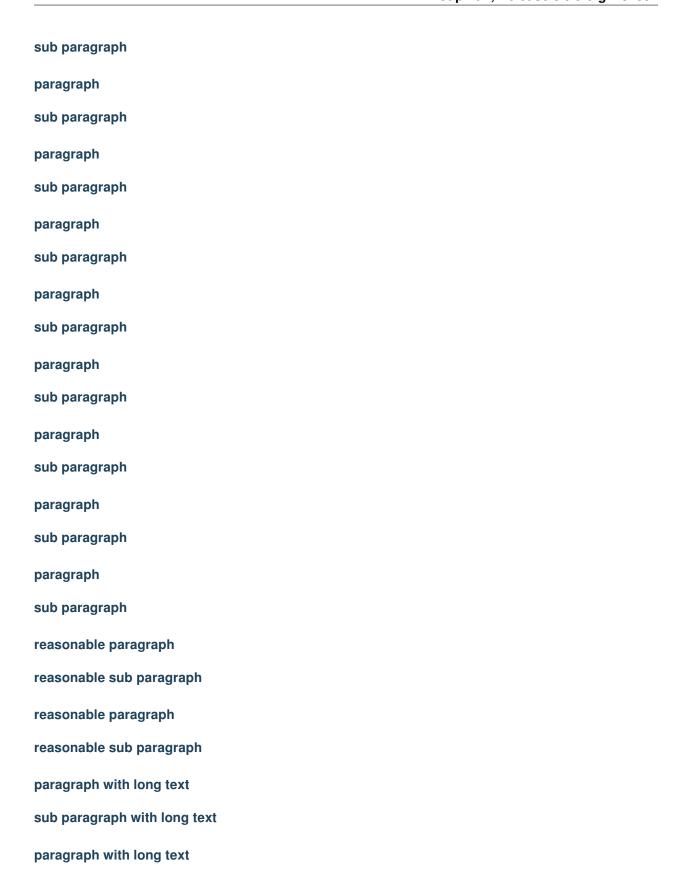
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph

]:

6.5.6 Big docs example 6

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text

sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph

[]:

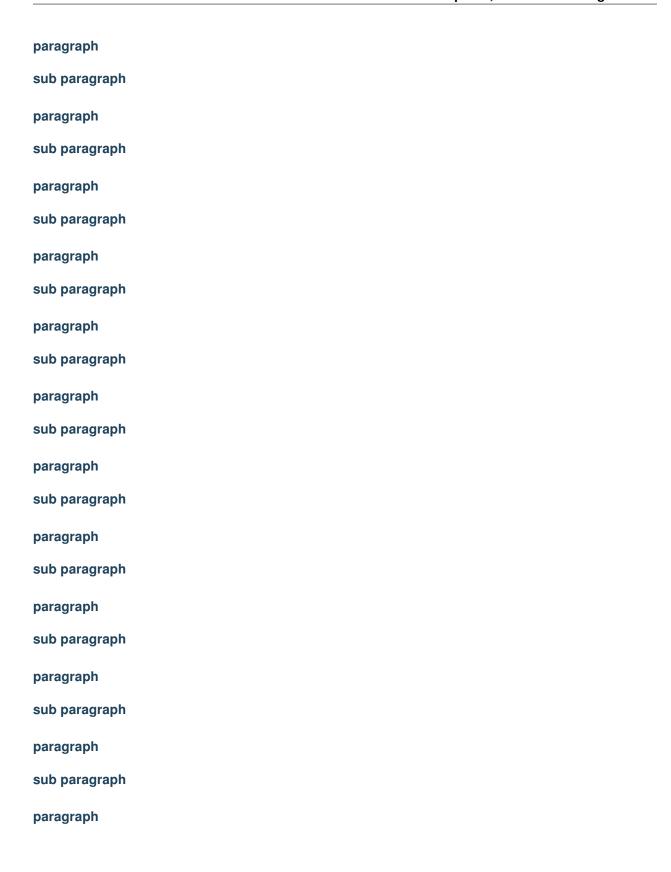
6.5.7 Big docs example 1

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text

sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph []:

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph



[]:

6.5.8 Big docs example 9

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph

sub paragraph

[]:

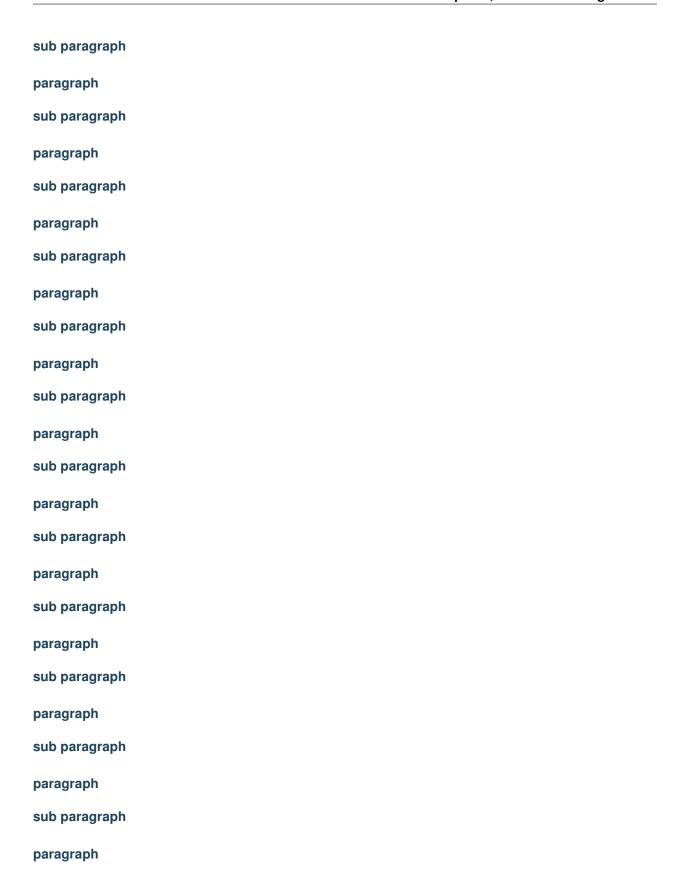
6.5.9 Big docs example 1

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph



sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text

paragraph



sub paragraph

paragraph

sub paragraph

paragraph

sub paragraph

f 1:

6.5.10 Big docs example 10

reasonable paragraph
reasonable sub paragraph
reasonable paragraph

reasonable sub paragraph

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text

paragraph with extra super long text

sub paragraph with extra super long text

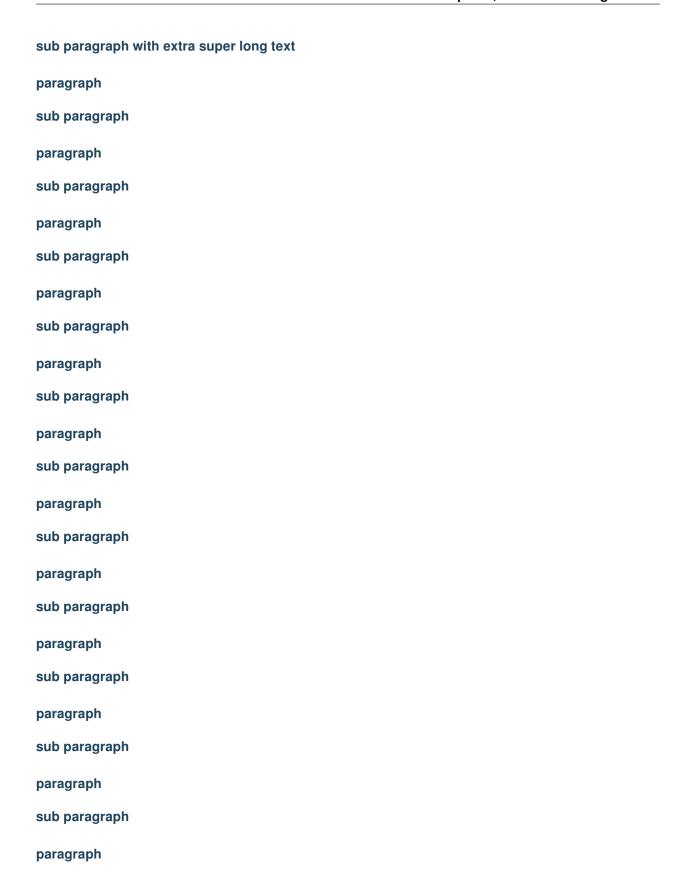
paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text



sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text



sub paragraph

paragraph

sub paragraph

paragraph

sub paragraph

paragraph

sub paragraph

[]:

6.6 Big sub chapter 2

6.6.1 Big docs example 1

reasonable paragraph

reasonable sub paragraph

reasonable paragraph

reasonable sub paragraph

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text

paragraph with long text

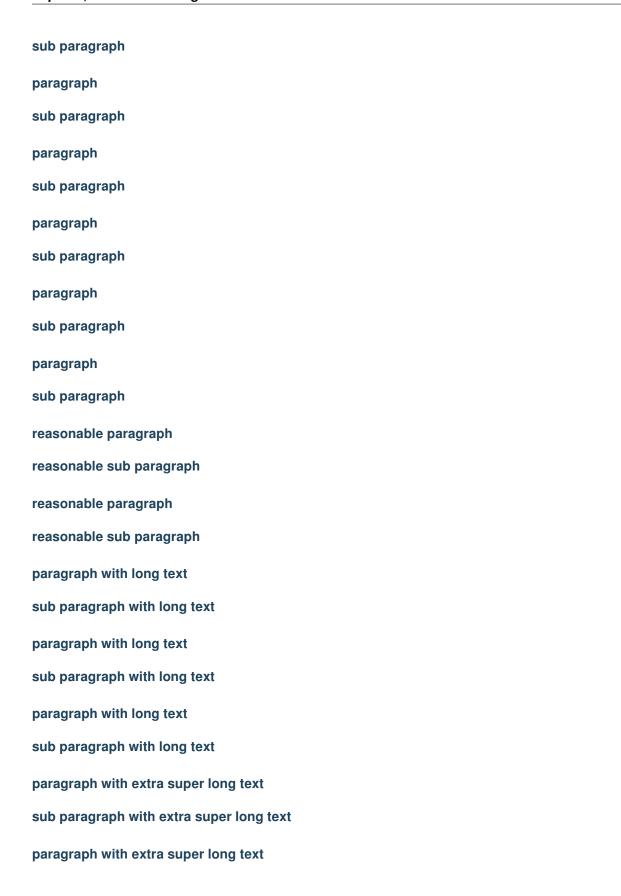
sub paragraph with long text

paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text

sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

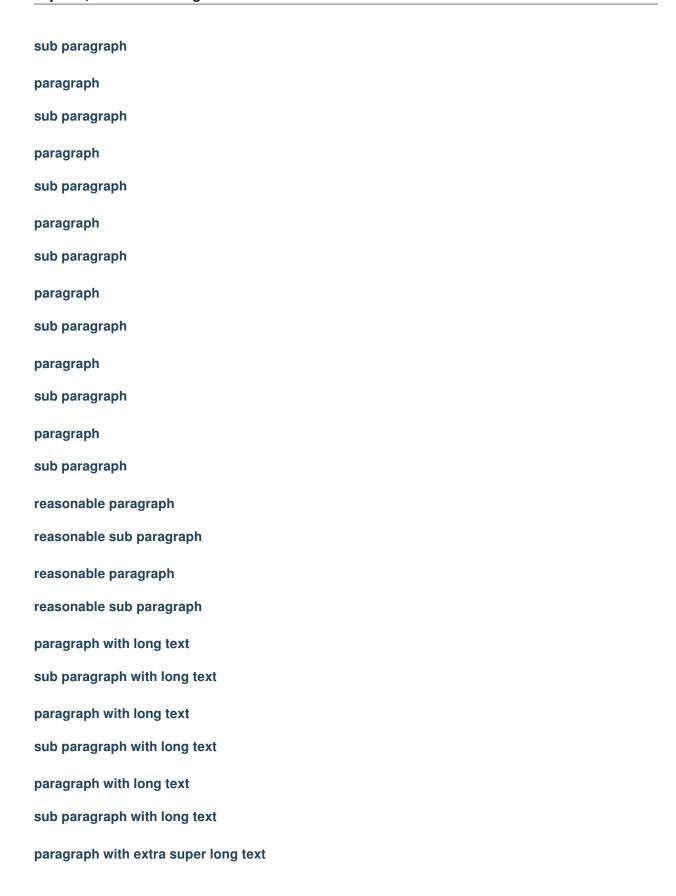
sub paragraph
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
paragraph
paragraph
sub paragraph
sub paragraph

]:

6.6.2 Big docs example 2

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text

sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph
paragraph
sub paragraph
paragraph
sub paragraph
paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
sub paragraph
paragraph

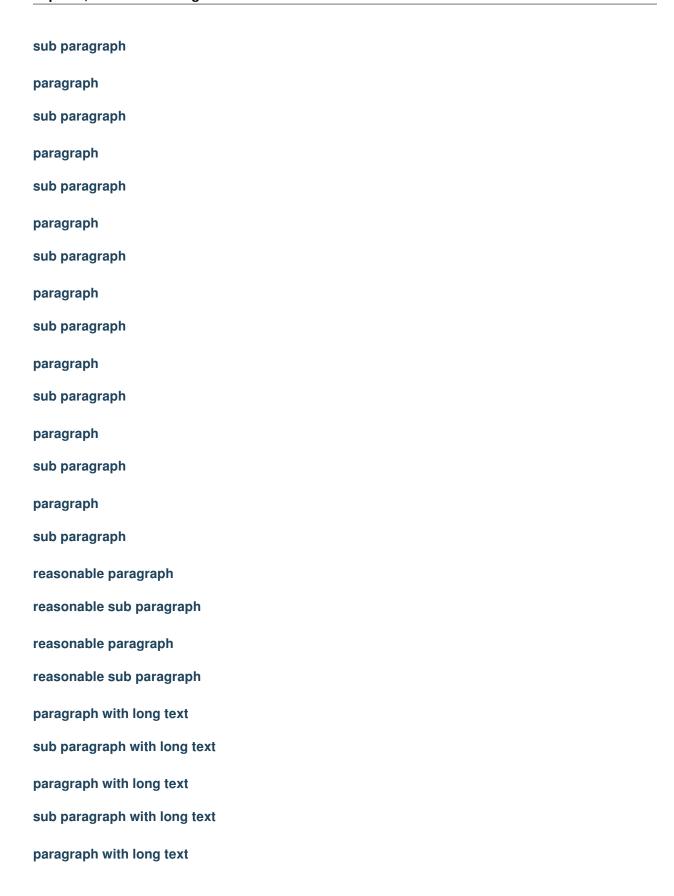
sub paragraph

[]:

6.6.3 Big docs example 3

reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text

sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph

1:

6.6.4 Big docs example 4

reasonable sub paragraph

reasonable paragraph

reasonable paragraph

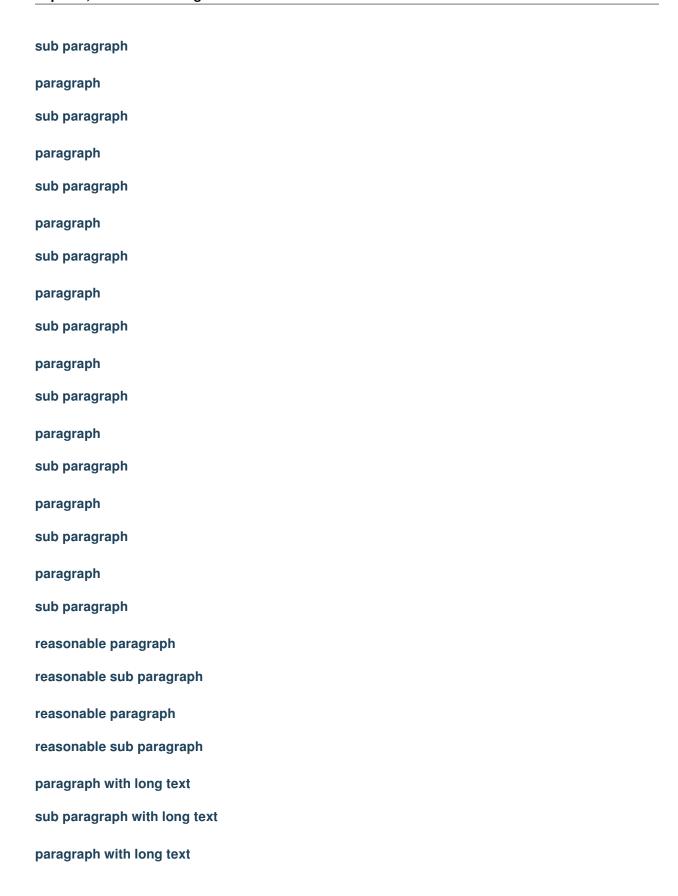
reasonable sub paragraph

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph

6.6.5 Big docs example 5

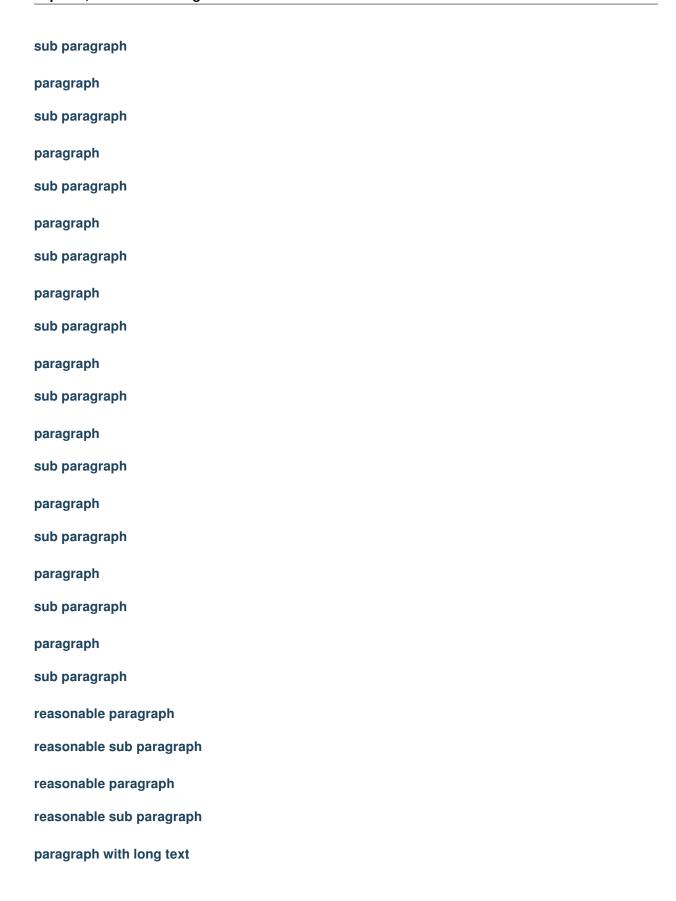
reasonable paragraph
reasonable sub paragraph

reasonable sub paragraph

reasonable paragraph

paragraph with long text

sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph



sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph

6.6.6 Big docs example 6

reasonable paragraph

reasonable sub paragraph

reasonable paragraph

reasonable sub paragraph

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text

paragraph with long text

sub paragraph with long text

paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text

sub paragraph with extra super long text

paragraph with extra super long text

sub paragraph with extra super long text

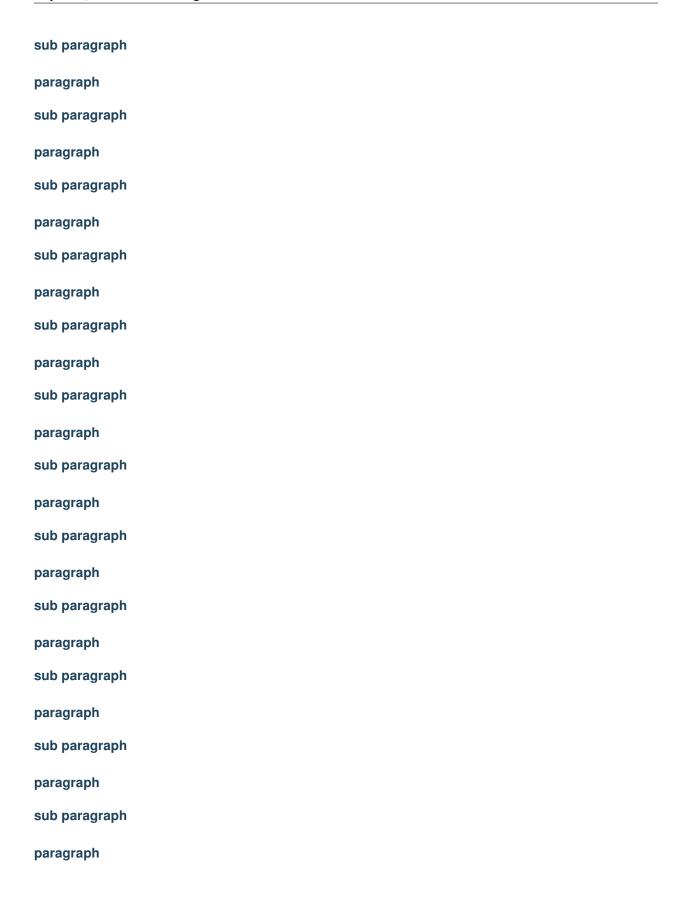
paragraph

sub paragraph

paragraph

sub paragraph

paragraph



sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph sub paragraph paragraph

sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	

[]:

6.6.7 Big docs example 1

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph paragraph

sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	
sub paragraph	
paragraph	

sub paragraph
paragraph
sub paragraph

[]:

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph sub paragraph

paragraph
sub paragraph
paragraph
sub paragraph

```
paragraph
sub paragraph
paragraph
sub paragraph
```

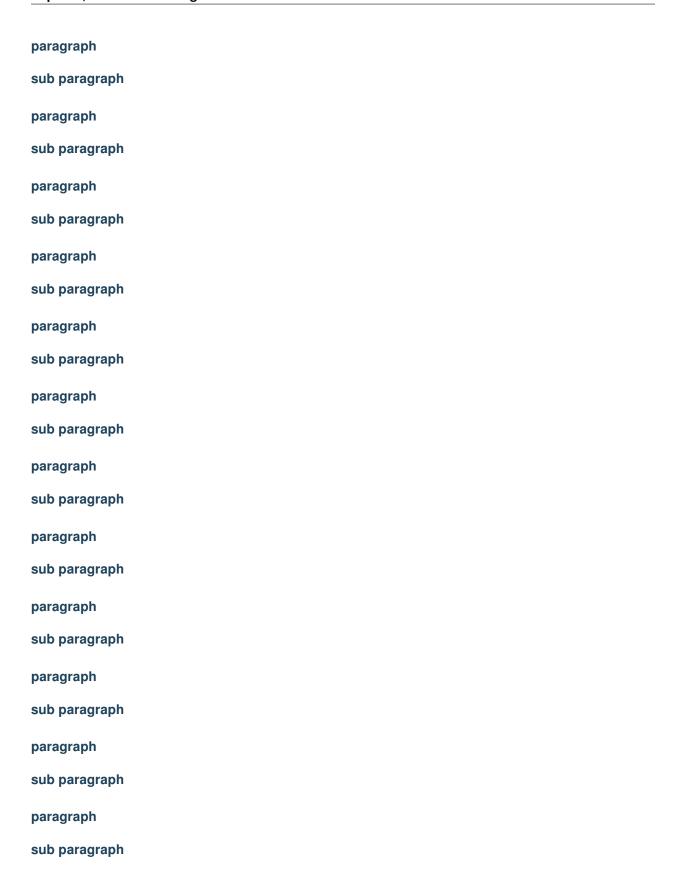
[]:

6.6.8 Big docs example 9

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text

paragraph
sub paragraph
paragraph
sub paragraph

paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text

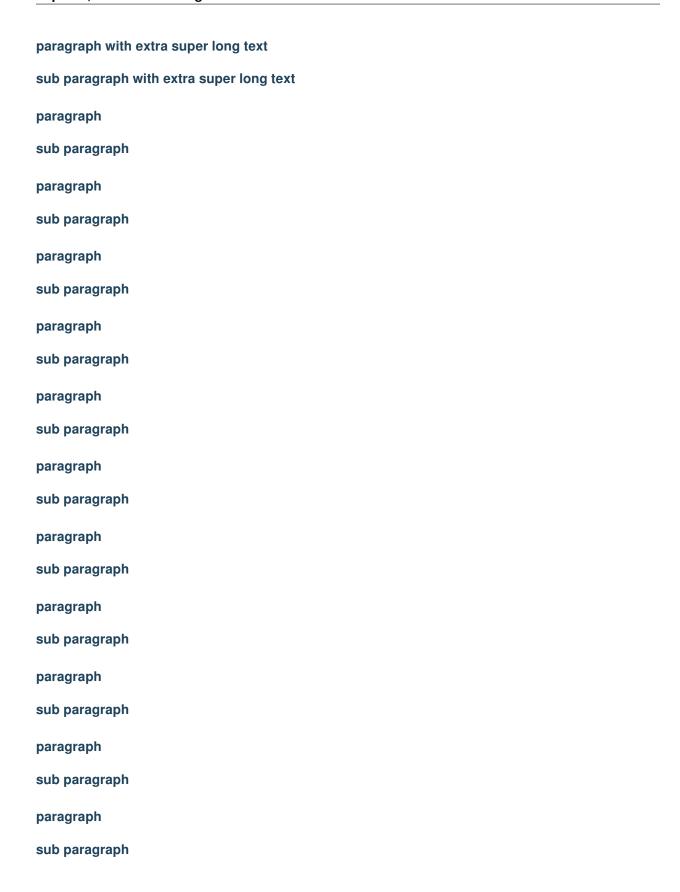


```
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
```

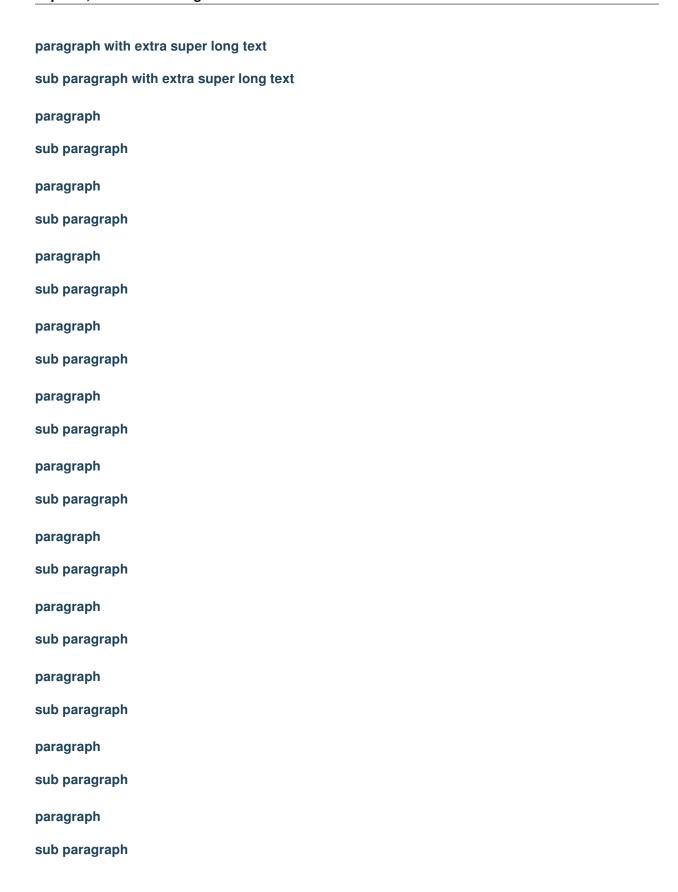
1:

6.6.9 Big docs example 1

reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text



paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text

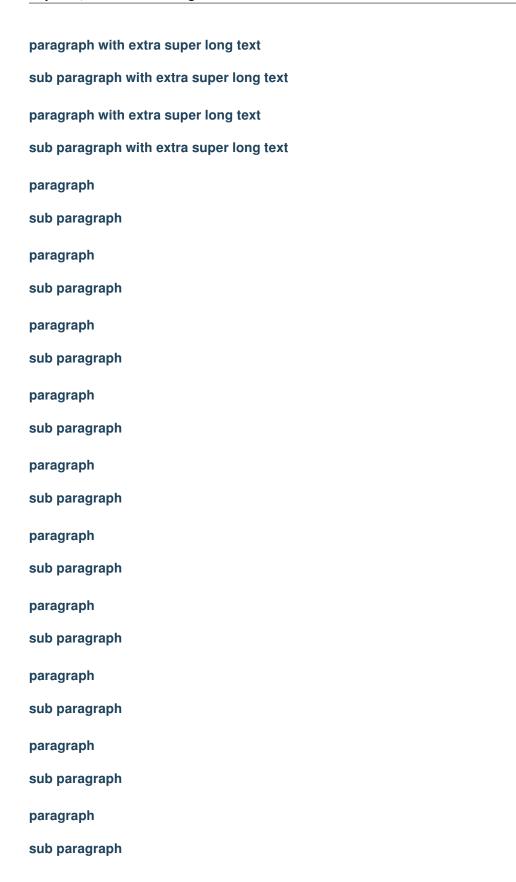


```
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
```

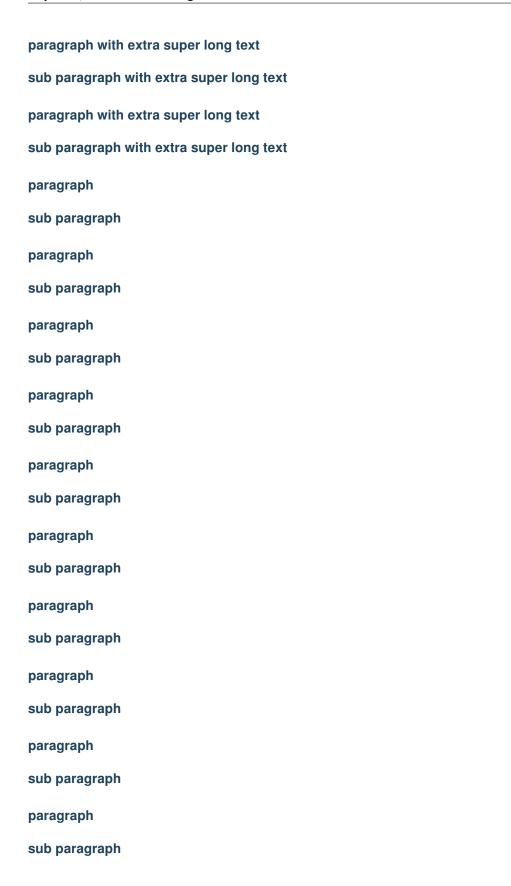
[]:

6.6.10 Big docs example 10

```
reasonable paragraph
reasonable sub paragraph
reasonable paragraph
reasonable sub paragraph
paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
paragraph with long text
sub paragraph with long text
sub paragraph with long text
sub paragraph with extra super long text
sub paragraph with extra super long text
paragraph with extra super long text
sub paragraph with extra super long text
```



paragraph sub paragraph reasonable paragraph reasonable sub paragraph reasonable paragraph reasonable sub paragraph paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with long text sub paragraph with long text paragraph with extra super long text sub paragraph with extra super long text paragraph with extra super long text sub paragraph with extra super long text



```
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
paragraph
sub paragraph
sub paragraph
sub paragraph
```

6.7 Example Challenge

6.7.1 Download exercises zip

Browse files online⁵⁷

This notebook has no solution!

We published solution on $Github^{58}$ only for example purposes, but normally all other files ending in -chal-sol or _chal_sol will be gitignored

```
[3]: from example_chal_sol import f
  f(3)
[3]: 3
[4]: def wow(x):
    raise Exception('TODO IMPLEMENT ME !')
```

⁵⁷ https://github.com/DavidLeoni/jupman/tree/master/challenge-example

⁵⁸ https://github.com/DavidLeoni/jupman/blob/master/challenge-example/example-chal-sol.ipynb

SEVEN

TEMPLATES

7.1 Changelog

Jupman Jupyter Manager

jupman.softpython.org⁵⁹

7.1.1 August 11th, 2022 - 3.5.4

- Python Tutor:
 - added credits #111
 - visualized vertical separator bar #110
 - fixed red arrow misalignment #105
 - stable ids #107
 - removed 'Customize visualization' #108
- Github actions: automatically build themed version #100
- SoftPython theme #92: various fixes
- Fixed relative html a, img with attributes in markdown not working in zip: #113

7.2 June 4th 2022 - 3.5

- generated html can be really used offline #96, also fixes wrong math symbols with local build #3
- automated testing on github actions #99
- virtual env install, pinpointed build dependencies #82
- fixed text overflow on smartphones #94, fixed softpython theme flag
- github actions: always reset html output #98

⁵⁹ https://jupman.softpython.org

7.2.1 April 28th 2022 - 3.4

- fixed softpython theme font size #92
- · restructured manual

7.2.2 February 25th 2022 - 3.3

Implemented:

- jupman-preprocess #64
- big docs support #77
- Challenges support (suboptimal but usable) #56
- jupman-purge #59
- jupman.draw_text for #66
- jupman.save_py function
- jupman.get_doc as nice way to print function documentation #68
- jupman.draw_text to show some ASCII characters in local build #66
- jupman.mem_limit for Linux #62
- Home link should point to index.html #71
- · optional parameter conf to jmt.init
- deterministic zip #60 (requires python 3.7)

Fixed:

- Notebook validation failed: Non-unique cell id error #78
- exam pdf build breaks when using images #61

Defined:

- how to use custom anchors #80
- how to have single pages like References at menu bottom #70

7.2.3 October 17th 2020 - 3.2

- · added optional build on Github Actions
- solutions are finally hidden on the website, with a click-to-show button!
- introduced generic jupman-togglable and specific jupman-sol CSS classes
- improved menu navigation
- added softpython theme
- · images are now shown centered in HTML
- · moved to jupman.softpython.org
- fixed write here tag not preserving the line
- deprecated jupman_tools.ignore_spaces in favor of tag_regex

- updated nbsphinx to 0.7.1
- updated sphinx_rtd_theme to 0.4.3
- updated sphinx to 2.3.1
- updated pygments to 2.7.1

7.2.4 January 16th 2020 - 3.1

- removed jupman.init root parameter
- bugfixes
- upgraded nbsphinx from 0.3.4 to 0.5.0
- upgraded sphinx_rtd_theme from 0.2.5b1 to 0.4.3
- upgraded sphinx from 1.7.6 to 2.3.1
- upgraded recommonmark from 0.4.0 to 0.6.0

7.2.5 December 29th 2019 - 3.0

- much simplified folder structure
 - Issue 33⁶⁰
- removed solutions from header requirement
 - Issue 32⁶¹
- introduced tests (pytest, hypothesis)
- · removed old_news in favor of changelog.md
- Latex:
 - much better PDF cover
 - using xelatex
 - set up unicode mappings
- · several fixes

7.2.6 September 24th 2018 - 2.0

• now using index.ipynb as home. Hurray!

⁶⁰ https://github.com/DavidLeoni/jupman/issues/33

⁶¹ https://github.com/DavidLeoni/jupman/issues/32

7.2.7 September 19th 2018 - 1.0

- · fixed build.py
- added html templates examples
- cleaned toc (was showing too much when loading)

7.2.8 August 26th 2018 - 0.9

- implemented generation of exercises from solutions [Issue 14(https://github.com/DavidLeoni/jupman/issues/14)
- reverted to old jupman.init() code Issue 12⁶²

7.2.9 August 12th 2018 - 0.8

- Prepended all functions in jupman.py with jupman_
- replaced index with proper homepage. see Issue 11⁶³
 - from now on you need home.ipynb file, because replacing index.rst is a nightmare!
 - new index.rst is just a placeholder which simply redirects to home.html. Do not modify it.
 - put the toctree in toc.rst
- exercises ipynb can now stay in exercises/ folder; when exercises are zipped, jupman automatically adds to the zip
 the required site files. see Issue 12⁶⁴
- Tried %run at beginning of notebooks, without much satisfaction (see discussion in Issue 12⁶⁵):
- disabled toc by default in html files. To enable it, in python use %run -i ../../jupman --toc
- · renamed past-exams directory from 'past-exams' to 'exams'
- created info, error, warn, fatal functions to conf.py
- introduced new variable exercise_common_files in conf.py for common files to be zipped
- added pages exam-project, markdown, project-ideas,
- added cc-by.png
- renamed changelog.txt to changelog.md
- now using templates with curly brackets in in templating, like _JM_{some_property}
- jupman.js: now when manually saving html in Jupyter, resulting html correctly hides cells
- Fixes https://github.com/DavidLeoni/jupman/issues/2: now toc is present in local build for pdfs

⁶² https://github.com/DavidLeoni/jupman/issues/12

⁶³ https://github.com/DavidLeoni/jupman/issues/11

⁶⁴ https://github.com/DavidLeoni/jupman/issues/12

 $^{^{65}\} https://github.com/DavidLeoni/jupman/issues/12$

7.2.10 August 3rd 2018 - 0.7

- added jupman.py pytut() for displaying Python tutor in the cells
- added jupman.py toc=False option to jupman.py init to disable toc
- · removed jupman.pyuseless networkx import from
- · fixed usage indentation
- · added changelog.txt

7.3 Past Exams

[]:

7.4 Exam project

For general (credits, attendance), see course description at section Evaluation and exams

Delivery times

Ideas for possible projects: See here

Last update: TODO

In short:

7.4.1 What to do

First of all: send by email to TODO@TODO.COM a brief description of the project, to decide what to do. I will create a Google doc to keep track of progresses and / or problems found.

Once the project is defined, go on like this:

1 - Download zip with template (view online files TODO⁶⁶)

After unzipped, you will find a folder named NAME-SURNAME-ID, with these files inside:

```
- NAME-SURNAME-ID
- project.ipynb
- markdown.ipynb
- requirements.txt
- img
- example.png
```

- 2 Rename the folder NAME-SURNAME-ID with your data
- 3 run Jupyter from the folder you just renamed
- 4 edit file project.ipynb, closely following the indications in the following technical requirements
- 5 Once done, send project by email to TODO@TODO.COM

7.3. Past Exams 129

⁶⁶ https://www.GITHUB.TODO

7.4.2 Technical requirements

Write in *Markdown*

Python code

requirements.txt file

Graphical interfaces

Be careful to

7.5 Project ideas

7.5.1 TODO

Last update: TODO

7.5.2 Introduction

1:

7.6 Jupman Project

PUT:

TITLE

NAME - ID

DATAE

7.6.1 Introduction

Bla bla

7.6.2 Data sources

Bla bla

7.6.3 Data cleaning and integration

Bla bla

7.6.4 Analysis

Bla bla

7.6.5 Problems found

Bla bla

7.6.6 Conclusioni

Bla bla

[]:

7.7 Markdown

Briefly explain why markdown is so great ..

[]:

7.7. Markdown 131

CHAPTER

EIGHT

TESTS

8.1 Tests

Tests and cornercases.

The page Title has one sharp, the Sections always have two sharps.

8.1.1 Section 1

bla bla

8.1.2 Section 2

Subsections always have three sharps

Subsection 1

bla bla

Subsection 2

bla bla

8.1.3 Python code

```
[2]: x = 4
lst = [1,2,4]
```

8.1.4 Python in Markdown

```
x = 4
lst = [1,2,4]
```

8.1.5 Other quotes

I'm quoted with > symbol on multiple lines. Am I readable?

Somebody asked

```
I'm quoted with **spaces**
on multiple lines
Am I readable?
```

```
Generic quote with backticks
Can you read it?
Is it nice?
```

```
# bash quote with backticks
cd foo
ls
cat wiz.txt
```

8.1.6 Download links

Files manually put in $_$ static:

- · Download trial.odt
- · Download trial.pdf

Files in arbitrary folder position:

· Download requirements.txt

NOTE: download links are messy, see issue 8⁶⁷

8.1.7 Links to HTML

• Link to trial.html

⁶⁷ https://github.com/DavidLeoni/jupman/issues/8

8.1.8 Info/Warning Boxes

Until there is an info/warning extension for Markdown/CommonMark, such boxes can be created by using HTML elements like this:

Note: This is an info, mixed bold!

Note: This is a warn, mixed bold!

Note: This is a long info, plain title, single paragraph

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

This is a long warn, plain title, single paragraph

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

Note: This is a long info, plain title, multiple paragraph

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

This is a long warn, plain title, multiple paragraph

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Note: This is a long info, bold title

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

8.1. Tests 135

This is a long warn, plain title!

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur.

For this to work reliably, you should obey the following guidelines:

- The class attribute has to be either "alert alert-info" or "alert alert-warning", other values will not be converted correctly.
- · No further attributes are allowed.
- For compatibility with CommonMark, you should add an empty line between the <div> start tag and the beginning
 of the content.

8.1.9 Math

For math stuff, see npshpinx docs⁶⁸

Here we put just some equation to show it behaves fine in Jupman

This is infinity: ∞

8.1.10 Unicode

Unicode characters should display an HTML, but with latex you might have problems, and need to manually map characters in conf.py

You should see a star in a black circle: \otimes

You should see a check: ✓

table characters: | | L -

8.1.11 Tables

A small table

Name	Surname	Age	Occupation
Augustus	Gloop	40	Software Engineer
Veruca	Salt	34	Writer
Violet	Beauregarde	37	Accountant

 $^{^{68}\} https://nbsphinx.readthedocs.io/en/0.2.14/markdown-cells.html \# Equations$

An extra large table

Why	do	you	need	SO	much	chocolate?	I don't	understand.
Oompa	Loompa,	do	ba	dee	doo			
I've	got	a	perfect	puzzle	for	you.		
Oompa	Loompa,	do	ba	dee	doo			
If	you	are	wise	you'll	listen	to	me.	
What	do	you	get	when	you	guzzle	down	sweets?

A table with extra large text

Who	do you	blame?
Gum chewing's fine when it's once in	It stops you from smoking and brightens	But it's repulsive revolting and
a while.	your smile.	wrong.
Blaming the kids is a lie and a shame	You know exactly who's to blame:	The mother and the father!

8.1.12 Lists

- Home
 - Contents
 - Revisions
- More
 - Quickstart
 - Configure
 - * Building the manual
 - * Publish
 - * Further steps
 - Installation

8.1.13 Image

SVG Images

SVG images work in notebook, but here it is commented since it breaks Latex, see issue⁶⁹

```
![An image](../img/cc-by.svg)
```

This one also doesn't work (and shows ugly code in the notebook anyway)

```
from IPython.display import SVG
SVG(filename='../img/cc-by.svg')
```

8.1. Tests 137

⁶⁹ https://github.com/DavidLeoni/jupman/issues/1

PNG Images



Inline images - pure markdown

Bla ![A PNG image md] (../_static/img/notebook_icon.png) bli blo



Bla

bli blo

Inline images - markdown and img

bla bli blo



bla

bli blo

Img class

If we pass a class, it will to be present in the website:

```
<img alt="markimg7325" class="jupman-inline-img" src="../_static/img/notebook_icon.png \rightarrow">
```



This

should be inline

8.1.14 Expressions list

Highlighting does work both in Jupyter and Sphinx

Three quotes, multiple lines - Careful: put exactly 4 spaces indentation

```
1. [2,3,1] != "[2,3,1]"
```

Three quotes, multiple lines, more compact - works in Jupyter, doesn't in Sphinx

```
1. python [2,3,1] != "[2,3,1]"
```

2. python
$$[4,8,12] == [2*2,"4*2",6*2]$$

Highlighting doesn't work in Jupyter neither in Sphinx:

Three quotes, single line

```
1. python [2,3,1] != ["2",3,1]
```

2. python
$$[4,8,12] == [2*2,"4*2",6*2]$$

Single quote, single line

```
1. python [2,3,1] != ["2",3,1]
```

2. python
$$[4,8,12] == [2*2,"4*2",6*2]$$

3. python [][:] == "[]"

8.1.15 Togglable cells

There are various ways to have togglable cells.

8.1. Tests 139

Show/hide exercises (PREFERRED)

If you need clickable show/hide buttons for exercise solutions, see here: Editing - Exercise types⁷⁰. It manages comprehensively use cases for display in website, student zips, exams, etc

If you have other needs, we report here some test we made, but keep in mind this sort of hacks tend to change behaviour with different versions of jupyter.

Toggling with Javascript

- · Works in MarkDown
- · Works while in Jupyter
- · Works in HTML
- Does not show in Latex (which might be a good point, if you intend to put somehow solutions at the end of the document)
- NOTE: after creating the text to see the results you have to run the initial cell with jupman.init (as for the toc)
- NOTE: you can't use Markdown block code since of Sept 2017 doesn't show well in HTML output

HTML details in Markdown, code tag

- Works while in Jupyter
- Doesn't work in HTML output
- as of Sept Oct 2017, not yet supported in Microsoft browsers

Click here to see the code

```
question = raw_input("What?")
answers = random.randint(1,8)
if question == "":
    sys.exit()
```

HTML details in Markdown, Markdown mixed code

- · Works while in Jupyter
- Doesn't work in HTML output
- as of Sept Oct 2017, not yet supported in Microsoft browsers

Click here to see the code

```
question = raw_input("What?")
answers = random.randint(1,8)
if question == "":
    sys.exit()
```

 $^{^{70}\} https://jupman.softpython.org/en/latest/manual/editing.html\#Exercise-types$

HTML details in HTML, raw NBConvert Format

- Doesn't work in Jupyter
- Works in HTML output
 - NOTE: as of Sept Oct 2017, not yet supported in Microsoft browsers
- · Doesn't show at all in PDF output

Some other Markdown cell afterwards

8.1.16 Files in templates

Since Dec 2019 they are not accessible see issue 10^{71} , but it is not a great problem, you can always put a link to Github, see for example exam-yyyy-mm-dd.ipynb⁷²

8.1.17 Stripping answers

For stripping answers examples, see jupyter-example/jupyter-example-sol. For explanation, see editing

8.1.18 Formatting problems

Characters per line

Python standard for code has limit to 79, many styles have 80 (see Wikipedia⁷³). We can keep 80 like this, which should **not** display a scrollbar:

Plain:

```
_____
```

As python markup:

This should **not** display a scrollbar:

```
[3]: '-'*78
[3]: '------'
```

This should **not** display a scrollbar:

On website this may display a scroll bar, because it will actually print ' apexes plus the dashes

[5]: '-'*80

8.1. Tests 141

⁷¹ https://github.com/DavidLeoni/jupman/issues/10

⁷² https://github.com/DavidLeoni/jupman/tree/master/_templates/exam/exam-yyyy-mm-dd.ipynb

⁷³ https://en.wikipedia.org/wiki/Characters_per_line

```
[5]:|'------
```

Note errors hold 75 dashes:

Plain:

```
Traceback (most recent call last)
ZeroDivisionError
<ipython-input-15-9e1622b385b6> in <module>()
---> 1 1/0
ZeroDivisionError: division by zero
```

As python markup:

```
ZeroDivisionError
                                          Traceback (most recent call last)
<ipython-input-15-9e1622b385b6> in <module>()
----> 1 1/0
ZeroDivisionError: division by zero
```

```
[6]: len('----
[6]: 75
```

Very large input

In Jupyter: default behaviour, show scrollbar

On the website: default behaviour, show scrollbar

```
[7]: # line with an exceedingly long comment line with an exceedingly long comment line...
     →with an exceedingly long comment line with an exceedingly long comment line with an_
     →exceedingly long comment line with an exceedingly long comment
     # line with an an out-of-this-world long comment line with an an out-of-this-world.
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an
     →an out-of-this-world long comment line with an an out-of-this-world long comment
     →line with an an out-of-this-world long comment line with an an out-of-this-world
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an
     →an out-of-this-world long comment line with an an out-of-this-world long comment
     →line with an an out-of-this-world long comment line with an an out-of-this-world
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an_
     →an out-of-this-world long comment line with an an out-of-this-world long comment.
     →line with an an out-of-this-world long comment line with an an out-of-this-world
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an_
     →an out-of-this-world long comment line with an an out-of-this-world long comment
     →line with an an out-of-this-world long comment line with an an out-of-this-world
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an_
     →an out-of-this-world long comment line with an an out-of-this-world long comment.
     →line with an an out-of-this-world long comment line with an an out-of-this-world
     → long comment line with an an out-of-this-world long comment line with a(continues on mextipage)
     \hookrightarrow this-world long comment line with an an out-of-this-world long comment line with an \Box
       an out-of-this-world long comment line with an an out-of-this-world long
    142 ine with an an out-of-this-world long comment line with an an out-of-Chapter 8 details
     →long comment line with an an out-of-this-world long comment line with an an out-of-
     →this-world long comment line with an an out-of-this-world long comment line with an_
     \hookrightarrowan out-of-this-world long comment line with an an out-of-this-world long comment.
```

Very long HTML (and long code line)

Should expand in vertical as much as it wants.

Very long output

In Jupyter: by clicking, you can collapse

On the website: a scrollbar should appear

```
[9]: for x in range(150):
         print('long output ...', x)
     long output ... 0
     long output ... 1
     long output ... 2
     long output ... 3
     long output ... 4
     long output ... 5
     long output ... 6
     long output ... 7
     long output ... 8
     long output ... 9
     long output ... 10
     long output ... 11
     long output ... 12
     long output ... 13
     long output ... 14
     long output ... 15
     long output ... 16
     long output ... 17
     long output ... 18
     long output ... 19
     long output ... 20
     long output ... 21
     long output ... 22
     long output ... 23
     long output ... 24
     long output ... 25
     long output ... 26
                                                                                          (continues on next page)
```

8.1. Tests 143

```
long output ... 27
long output ... 28
long output ... 29
long output ... 30
long output ... 31
long output ... 32
long output ... 33
long output ... 34
long output ... 35
long output ... 36
long output ... 37
long output ... 38
long output ... 39
long output ... 40
long output ... 41
long output ... 42
long output ... 43
long output ... 44
long output ... 45
long output ... 46
long output ... 47
long output ... 48
long output ... 49
long output ... 50
long output ... 51
long output ... 52
long output ... 53
long output ... 54
long output ... 55
long output ... 56
long output ... 57
long output ... 58
long output ... 59
long output ... 60
long output ... 61
long output ... 62
long output ... 63
long output ... 64
long output ... 65
long output ... 66
long output ... 67
long output ... 68
long output ... 69
long output ... 70
long output ... 71
long output ... 72
long output ... 73
long output ... 74
long output ... 75
long output ... 76
long output ... 77
long output ... 78
long output ... 79
long output ... 80
long output ... 81
long output ... 82
long output ... 83
```

(continues on next page)

```
long output ... 84
long output ... 85
long output ... 86
long output ... 87
long output ... 88
long output ... 89
long output ... 90
long output ... 91
long output ... 92
long output ... 93
long output ... 94
long output ... 95
long output ... 96
long output ... 97
long output ... 98
long output ... 99
long output ... 100
long output ... 101
long output ... 102
long output ... 103
long output ... 104
long output ... 105
long output ... 106
long output ... 107
long output ... 108
long output ... 109
long output ... 110
long output ... 111
long output ... 112
long output ... 113
long output ... 114
long output ... 115
long output ... 116
long output ... 117
long output ... 118
long output ... 119
long output ... 120
long output ... 121
long output ... 122
long output ... 123
long output ... 124
long output ... 125
long output ... 126
long output ... 127
long output ... 128
long output ... 129
long output ... 130
long output ... 131
long output ... 132
long output ... 133
long output ... 134
long output ... 135
long output ... 136
long output ... 137
long output ... 138
long output ... 139
long output ... 140
                                                                                     (continues on next page)
```

8.1. Tests 145

```
long output ... 141
long output ... 142
long output ... 143
long output ... 144
long output ... 145
long output ... 146
long output ... 147
long output ... 148
long output ... 149
```

8.1.19 Python tutor tests

See Pytutor page

8.2 Python tutor tests

There are various ways to embed Python tutor, first we put the recommended one.

RECOMMENDED: You can put a call to jupman.pytut() at the end of a cell, and the cell code will magically appear in python tutor in the output (except the call to pytut() of course). Does not need internet connection.

```
[1]: import sys
    sys.path.append('../')
    import jupman

[2]: x = [5,8,4,10,30,20,40,50,60,70,20,30]
    y= {3:9}
    z = [x]
    jupman.pytut()

[2]: <IPython.core.display.HTML object>
```

8.2.1 scope

BEWARE of variables which were initialized in previous cells, they WILL NOT be available in Python Tutor:

```
[3]: w = 8

[4]: x = w + 5
    jupman.pytut()

Traceback (most recent call last):
    File "../jupman.py", line 2434, in _runscript
        self.run(script_str, user_globals, user_globals)
    File "/usr/lib/python3.7/bdb.py", line 578, in run
        exec(cmd, globals, locals)
    File "<string>", line 2, in <module>
    NameError: name 'w' is not defined

[4]: <IPython.core.display.HTML object>
```

8.2.2 window overflow

When too much right space is taken, it might be difficult to scroll:

8.2.3 pytut execution

Some cells might execute in Jupyter but not so well in Python Tutor, due to its inherent limitations⁷⁴:

8.2.4 infinite loops

Since execution occurs first in Jupyter and then in Python tutor, if you have an infinite loop no Python Tutor instance will be spawned:

```
while True:
    pass
jupman.pytut()
```

⁷⁴ https://github.com/pgbovine/OnlinePythonTutor/blob/master/unsupported-features.md

8.2.5 resizability

Long vertical and horizontal expansion should work:

```
[8]: x = {0:'a'}
    for i in range(1,30):
        x[i] = x[i-1]+str(i*10000)
        jupman.pytut()

[8]: <IPython.core.display.HTML object>
```

8.2.6 cross arrows

With multiple visualizations, arrows shouldn't cross from one to the other even if underlying script is loaded multiple times (relates to visualizerIdOverride)

```
[9]: x = [1,2,3]

jupman.pytut()

[9]: <IPython.core.display.HTML object>
```

8.2.7 print output

With only one line of print, Print output panel shouldn't be too short:

```
[10]: print("hello")
    jupman.pytut()
    hello
[10]: <IPython.core.display.HTML object>

[11]: y = [1,2,3,4]
    jupman.pytut()
[11]: <IPython.core.display.HTML object>
```

8.2.8 Alignment test 1

Test for https://github.com/DavidLeoni/jupman/issues/105

Before fixing red arrow was badly misaligned and went way past the function call.

```
f([[2,3,4]], 0)
jupman.pytut()

[12]: <IPython.core.display.HTML object>
```

8.2.9 Alignment test 2

Check the arrow stays on right line even in long code.

```
[13]: def f(mat, i):
         """ Do something else"""
         row = []
         row.append(mat[i])
         return row
     m = [
           ['a','b'],
           ['c','d'],
           ['a','e'],
     ]
     \#row = f(m, 0)
     x = 3
     if x == x:
        m = [
             [1,2,3],
             [4,3]
         ]
     print("zim")
     y = 1
     if y == y:
         m = [
             [1,2,3],
             [4,3]
     print("zam")
     jupman.pytut()
     zam
[13]: <IPython.core.display.HTML object>
```

8.2.10 Alternative: HTML magics

Another option is to directly paste Python Tutor iframe in the cells, and use Jupyter %%HTML magics command.

HTML should be available both in notebook and website - of course, requires an internet connection.

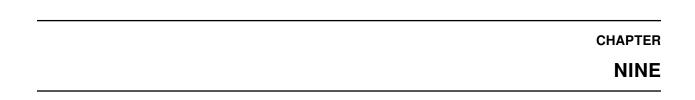
Beware: you need the HTTPS!

8.2.11 Alternative: NBTutor

To show Python Tutor in notebooks, there is already a jupyter extension called NBTutor 75 , afterwards you can use magic %%nbtutor to show the interpreter.

Unfortunately, it doesn't show in the generated HTML:-/

 $^{^{75}\} https://github.com/lgpage/nbtutor$



REFERENCES

Shows how to put a single page at the bottom of the sidebar, visible without being inside a section. See this issue⁷⁶

⁷⁶ https://github.com/DavidLeoni/jupman/issues/70