Chapter1-Preliminaries

April 1, 2021

Scientific Computation (MKP3303)

> R.U.Gobithaasan (2021). Scientific Computing, Lectures for Undergraduate Degree Program B.Sc (Applied Mathematics), Faculty of Ocean Engineering Technology, University Malaysia Terengganu. https://sites.google.com/site/gobithaasan/LearnTeach

Chapter 1: Preliminaries
1. Command line reference 2. Using Notebook 3. Markdown Cells 4. Scientific Python Ecosystem 5. Documentation and Help files 6. Input / Output caching

References: - w3schools Online Materials - SciPi Lecture Notes - Robert Johansson, Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib (2019, Apress). - Donaldson Toby,Python: Visual QuickStart Guide (2008, Peachpit Press). - Tony Gaddis-Starting Out with Python,(2018,Global Edition-Pearson Education) - Robert Johansson August, Introduction to Scientific Computing in Python Continuum Analytics, (2015)

```
$ jupyter nbconvert --to html NOTEBOOK-NAME.ipynb
```

1 Command line reference

Click on this link to see all the commands available: - Windows commands - Mac

2 Using Jupyter Notebook

some shortcuts:

- b: Create a new cell below the currently selected cell.
- a: Create a new cell above the currently selected cell.
- d-d Delete the currently selected cell.
- 1 to 6: Heading cell of level 1 to 6.
- x: Cut currently selected cell.
- c: Copy currently selected cell.
- v: Paste cell from the clipboard.
- m: Convert a cell to a markdown cell.
- y: Convert a cell to a code cell.

- Up: Select previous cell.
- Down: Select next cell.
- Enter: Enter edit mode.
- Escape: Exit edit mode.
- Shift-Enter: Run the cell.
- h: Display a help window with a list of all available keyboard shortcuts.
- 0-0: Restart the kernel.
- i-i: Interrupt an executing cell.
- s: Save the notebook.

3 Markdown Cells

Summary of Markdown Syntax for Jupyter Notebook Markdown Cells

Fonts:

- italics: *text*
- bold: **text**
- stike-through: ~~text~~
- fixed-width: `text`
- url: URL text
- Vertatim(with tab):

```
def func(x):
    return x ** 2
```

- New paragraph: with an empty line.
- Types of headers:

[39]:
```
'''
# Level 1 heading
## Level 2 heading
### Level 3 heading
'''
```

[39]:
```
' \n# Level 1 heading \n## Level 2 heading \n### Level 3 heading\n'
```

- Block quote: > Text here is indented and offset > from the main text body.
- Unordered list (use - or *):
- Item one
- Item two
- Item three
- Ordered list:

1. Item one

2. Item two
3. Item three

Table:

A	B	C
1	2	3
4	5	6

- image: local machine

- image: internet

[40]:
```
#![Alternative text](https://www.python.org/static/img/python-logo.png)
```

- Inline LateX equation:

$f2(x, y, z) = x^2 + y^3 + \sqrt{z}$

- Displayed LateX equation. See some examples at Latex Cookbook

$$\begin{aligned} y &= x^4 + 4 \\ &= (x^2 + 2)^2 - 4x^2 \\ &\leq (x^2 + 2)^2 \end{aligned} \tag{1}$$

4 Scientific Python Ecosystem

run command: > pip install *module*

where *module* is the name of the module you want to install. For example to install a module called **pandas**: > pip install pandas

Core numeric libraries - **Numpy**: numerical computing with powerful numerical arrays objects, and routines to manip- ulate them. Numpy

- **Scipy**: high-level numerical routines. Optimization, regression, interpolation, etc. Scipy
- **Matplotlib**: 2-D visualization, "publication-ready" plots Matplotlib

source: SciPi Lecture Notes

4.1 Needed Modules / Libraries

Pip install the modules you need. You may browse for modules at PyPI · The Python Package Index

[41]:
```
!python3 --version
!pip freeze |(grep 'matplotlib\|numpy\|jupyter\|scipy')
```

```
Python 3.8.3
jupyter==1.0.0
jupyter-client==6.1.3
jupyter-console==6.1.0
jupyter-core==4.6.3
jupyter-packaging==0.7.12
jupyter-server==1.5.1
jupyterlab==3.0.12
jupyterlab-server==2.3.0
jupyterthemes==0.20.0
matplotlib==3.2.2
numpy==1.19.4
scipy==1.5.2
```

4.1.1 Loaded Modules

[42]:
```
import sys
sys.modules.keys();
```

[43]:
```
print(dir())
```

```
['In', 'Out', '_', '_18', '_20', '_22', '_31', '_33', '_35', '_37', '_39', '_5',
'_7', '_9', '__', '___', '__builtin__', '__builtins__', '__doc__', '__loader__',
'__name__', '__package__', '__spec__', '_dh', '_exit_code', '_i', '_i1', '_i10',
'_i11', '_i12', '_i13', '_i14', '_i15', '_i16', '_i17', '_i18', '_i19', '_i2',
'_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27', '_i28', '_i29',
'_i3', '_i30', '_i31', '_i32', '_i33', '_i34', '_i35', '_i36', '_i37', '_i38',
'_i39', '_i4', '_i40', '_i41', '_i42', '_i43', '_i5', '_i6', '_i7', '_i8',
'_i9', '_ih', '_ii', '_iii', '_oh', 'a', 'exit', 'get_ipython', 'importlib',
'm', 'math', 'myModule', 'quit', 'sys']
```

[44]:
```
import myModule
print(dir())
```

```
['In', 'Out', '_', '_18', '_20', '_22', '_31', '_33', '_35', '_37', '_39', '_5',
'_7', '_9', '__', '___', '__builtin__', '__builtins__', '__doc__', '__loader__',
'__name__', '__package__', '__spec__', '_dh', '_exit_code', '_i', '_i1', '_i10',
'_i11', '_i12', '_i13', '_i14', '_i15', '_i16', '_i17', '_i18', '_i19', '_i2',
'_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27', '_i28', '_i29',
```

```
'_i3', '_i30', '_i31', '_i32', '_i33', '_i34', '_i35', '_i36', '_i37', '_i38',
'_i39', '_i4', '_i40', '_i41', '_i42', '_i43', '_i44', '_i5', '_i6', '_i7',
'_i8', '_i9', '_ih', '_ii', '_iii', '_oh', 'a', 'exit', 'get_ipython',
'importlib', 'm', 'math', 'myModule', 'quit', 'sys']
```

[45]:
```
del myModule
```

[46]:
```
print(dir())
```

```
['In', 'Out', '_', '_18', '_20', '_22', '_31', '_33', '_35', '_37', '_39', '_5',
'_7', '_9', '__', '___', '__builtin__', '__builtins__', '__doc__', '__loader__',
'__name__', '__package__', '__spec__', '_dh', '_exit_code', '_i', '_i1', '_i10',
'_i11', '_i12', '_i13', '_i14', '_i15', '_i16', '_i17', '_i18', '_i19', '_i2',
'_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27', '_i28', '_i29',
'_i3', '_i30', '_i31', '_i32', '_i33', '_i34', '_i35', '_i36', '_i37', '_i38',
'_i39', '_i4', '_i40', '_i41', '_i42', '_i43', '_i44', '_i45', '_i46', '_i5',
'_i6', '_i7', '_i8', '_i9', '_ih', '_ii', '_iii', '_oh', 'a', 'exit',
'get_ipython', 'importlib', 'm', 'math', 'quit', 'sys']
```

[47]:
```
a=2
```

[48]:
```
a
```

[48]:
```
2
```

[49]:
```
print(dir())
```

```
['In', 'Out', '_', '_18', '_20', '_22', '_31', '_33', '_35', '_37', '_39',
'_48', '_5', '_7', '_9', '__', '___', '__builtin__', '__builtins__', '__doc__',
'__loader__', '__name__', '__package__', '__spec__', '_dh', '_exit_code', '_i',
'_i1', '_i10', '_i11', '_i12', '_i13', '_i14', '_i15', '_i16', '_i17', '_i18',
'_i19', '_i2', '_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27',
'_i28', '_i29', '_i3', '_i30', '_i31', '_i32', '_i33', '_i34', '_i35', '_i36',
'_i37', '_i38', '_i39', '_i4', '_i40', '_i41', '_i42', '_i43', '_i44', '_i45',
'_i46', '_i47', '_i48', '_i49', '_i5', '_i6', '_i7', '_i8', '_i9', '_ih', '_ii',
'_iii', '_oh', 'a', 'exit', 'get_ipython', 'importlib', 'm', 'math', 'quit',
'sys']
```

[50]:
```
import myModule
import importlib
importlib.reload(myModule)
```

[50]:
```
<module 'myModule' from '/Volumes/GoogleDrive/My Drive/0teaching/2021-2020/Sem-2
/2020MKP3303/ScientificComputingWithPython/NotebookLectures/myModule.py'>
```

5 Documentation & Help files

[51]:
```
import math
```

[52]:
```
math.pow(2,4)
```

[52]:
```
16.0
```

[53]:
```
import math as m # importing a module
```

[54]:
```
m.pow(2,5)
```

[54]:
```
32.0
```

[55]:
```
 #m. #tab after the dot operator to see all available functions
```

[56]:
```
help(m.pow)
```

```
Help on built-in function pow in module math:

pow(x, y, /)
    Return x**y (x to the power of y).
```

[57]:
```
# import a math module
import math
```

[58]:
```
help(math.log10)
```

```
Help on built-in function log10 in module math:

log10(x, /)
    Return the base 10 logarithm of x.
```

[59]:
```
?math.log10 # (For jupyter notebook) prompt a new window to show the info
```

```
Object `math.log10 # (For jupyter notebook) prompt a new window to show the info
` not found.
```

[60]:
```
help(math.log10)
```

```
Help on built-in function log10 in module math:

log10(x, /)
    Return the base 10 logarithm of x.
```

6 Input output Caching

[61]:
```
5+9
```

[61]:
```
14
```

[62]:
```
In[27] # previous input
```

[62]:
```
'import myModule\nprint(dir())'
```

[63]:
```
In[26]
```

[63]:
```
'print(dir())'
```

[65]:
```
Out[61] # previous output
```

[65]:
```
14
```