Introduction

Probability and Statistics, Spring 2017

Helsinki Metropolia University of Applied Sciences

The Environment

Python, Anaconda and jupyter Notebook

MetCat

Metropolia

Numerical Python

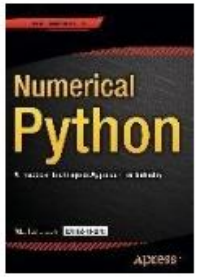
Tarkennettu haku

Haku: Numerical Python

Kirjaston kokoelmat | Kansainväliset e-aineistot

Näytetään **1 - 3** / 3

Järjestä Relevanssi ▼ Tuloksia sivulla 20 ▼



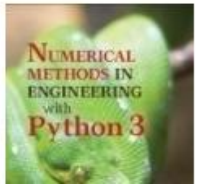
Numerical Python

E-kirja
Johansson, Robert
2015

Ei saatavuustietoja

Verkossa saatavilla:
SFX
SpringerLink Books Professional And Applied Computing 2015

Katso saatavuus kokotekstinä tai muissa kirjastoissa (sfx)



Numerical Methods in Engineering with Python 3, Third Edition

E-kirja
Kiusalaas, Jaan
2013

Ei saatavuustietoja

Rajaa hakua

- [ ] Verkossa saatavilla 2

Aineistotyyppi

- Kirja 3
  - E-kirja 2
  - Kirja 1

Tekijä

Vuosi

Kieli

https://www.continuum.io/downloads

DOWNLOAD ANACONDA NOW

Download for 

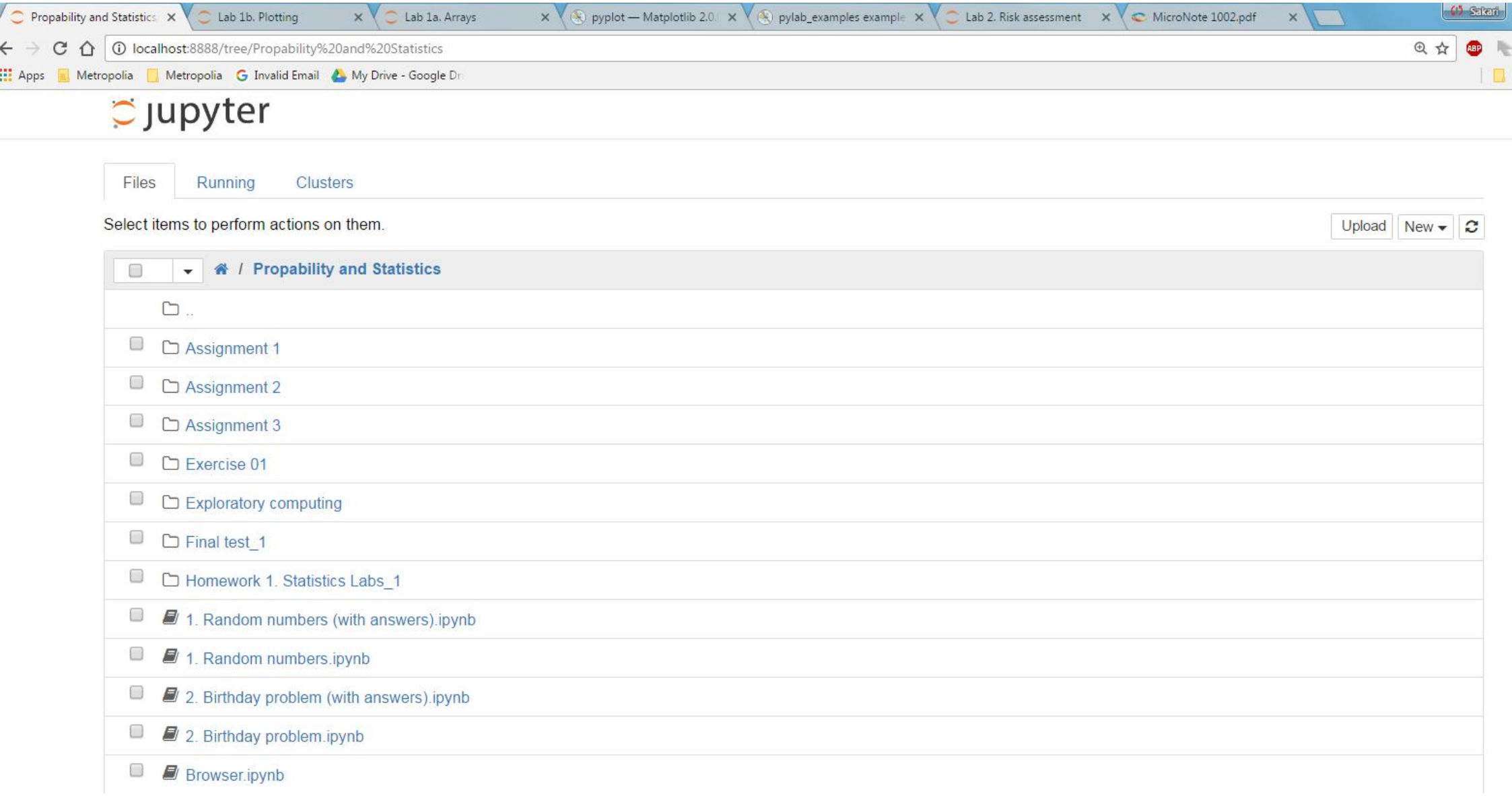
GET SUPERPOWERS WITH ANACONDA

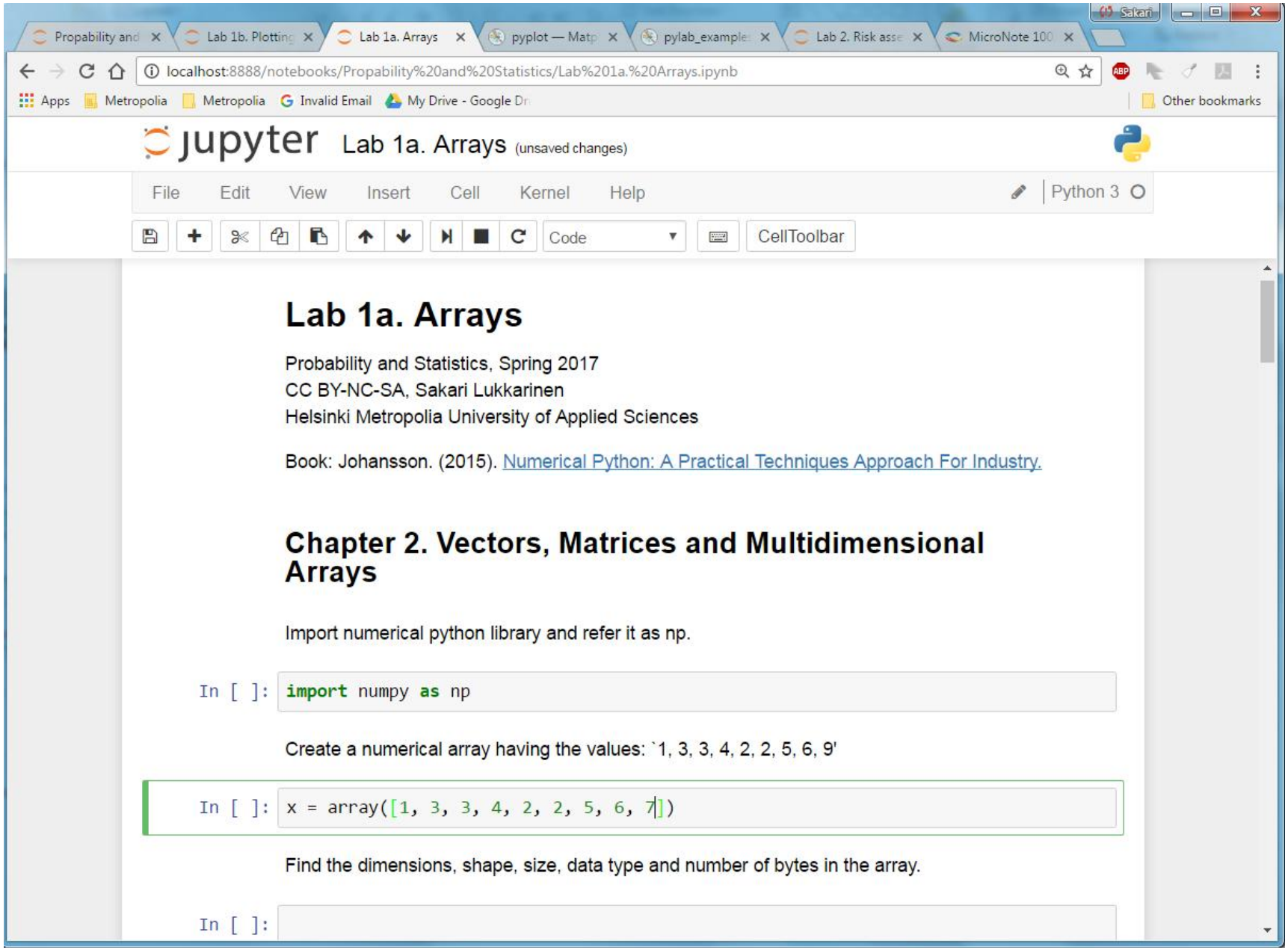
Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular Python, R and Scala packages for data science.

Additionally, you'll have access to over 720 packages that can easily be installed with conda, our renowned package, dependency and environment manager,

Which version should I download and install?
With Anaconda you can run multiple versions of Python in isolated environments, so choose the download with the Python version that you use more often, as that will be your default Python version.

If you don't have time or disk space for the entire distribution, try Miniconda which contains only conda and Python. Then install just the individual


Propability and Statistics
Lab 1b. Plotting
Lab 1a. Arrays
pyplot — Matplotlib 2.0
pylab_examples example
Lab 2. Risk assessment
MicroNote 1002.pdf
Safari
localhost:8888/tree/Propability%20and%20Statistics
Apps
Metropolia
Metropolia
Invalid Email
My Drive - Google Dr
jupyter
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
/ Propability and Statistics
..
Assignment 1
Assignment 2
Assignment 3
Exercise 01
Exploratory computing
Final test_1
Homework 1. Statistics Labs_1
1. Random numbers (with answers).ipynb
1. Random numbers.ipynb
2. Birthday problem (with answers).ipynb
2. Birthday problem.ipynb
Browser.ipynb

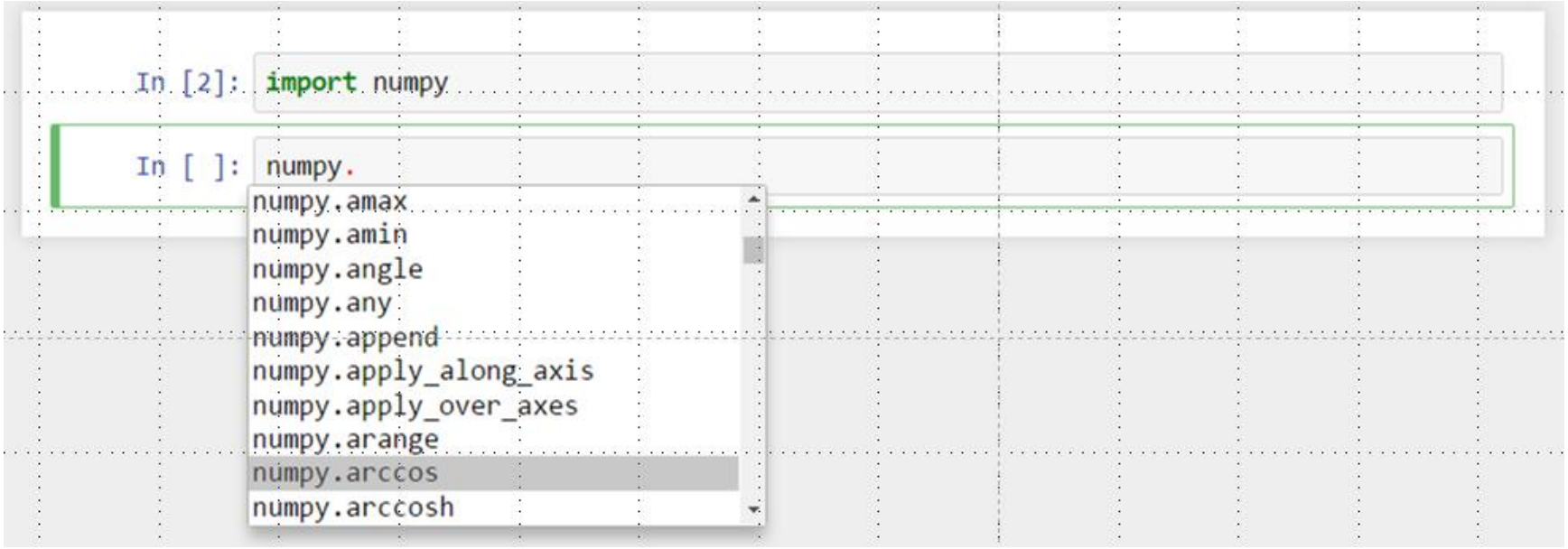

Propability and
Lab 1b. Plotting
Lab 1a. Arrays
pyplot — Matp
pylab_example
Lab 2. Risk asse
MicroNote 100
Sakari
localhost:8888/notebooks/Propability%20and%20Statistics/Lab%201a.%20Arrays.ipynb
Apps
Metropolia
Metropolia
Invalid Email
My Drive - Google Dr
Other bookmarks
jupyter
Lab 1a. Arrays
(unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar
Lab 1a. Arrays
Probability and Statistics, Spring 2017
CC BY-NC-SA, Sakari Lukkarinen
Helsinki Metropolia University of Applied Sciences
Book: Johansson. (2015). Numerical Python: A Practical Techniques Approach For Industry.
Chapter 2. Vectors, Matrices and Multidimensional Arrays
Import numerical python library and refer it as np.
In [ ]:
import numpy as np
Create a numerical array having the values: `1, 3, 3, 4, 2, 2, 5, 6, 9'
In [ ]:
x = array([1, 3, 3, 4, 2, 2, 5, 6, 7])
Find the dimensions, shape, size, data type and number of bytes in the array.
In [ ]:

Behind the scene

Figure 1-2. *An overview of the components and layers in the scientific computing environment for Python, from a user's perspective, from top to bottom. Users typically only interact with the top three layers, but the bottom layer constitutes a very important part of the software stack. An example of specific software components from each layer in the stack is shown in the right part of the figure*

Notebook: Input and output cells

Autocompletion



Documentation

```
In [6]: numpy.cos?
```

```
Type:            ufunc
String form:     <ufunc 'cos'>
File:            c:\anaconda3\lib\site-packages\numpy\__init__.py
Docstring:
cos(x[, out])

Cosine element-wise.

Parameters
----------
x : array_like
    Input array in radians.
out : ndarray, optional
    Output array of same shape as `x`.
```

Magic commands

```
In [7]: %pylab inline

        Populating the interactive namespace from numpy and matplotlib

In [8]: %lsmagic

Out[8]: Available line magics:
        %alias  %alias_magic  %autocall  %automagic  %autosave  %bookmark  %cd  %clear
        %cls  %colors  %config  %connect_info  %copy  %ddir  %debug  %dhist  %dirs  %do
        ctest_mode  %echo  %ed  %edit  %env  %gui  %hist  %history  %install_default_co
        nfig  %install_ext  %install_profiles  %killbgscripts  %ldir  %less  %load  %lo
        ad_ext  %loadpy  %logoff  %logon  %logstart  %logstate  %logstop  %ls  %lsmagic
          %macro  %magic  %matplotlib  %mkdir  %more  %notebook  %page  %pastebin  %pdb
          %pdef  %pdoc  %pfile  %pinfo  %pinfo2  %popd  %pprint  %precision  %profile
         %prun  %psearch  %psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickref
          %recall  %rehashx  %reload_ext  %ren  %rep  %rerun  %reset  %reset_selective
        %rmdir  %run  %save  %sc  %set_env  %store  %sx  %system  %tb  %time  %timeit
         %unalias  %unload_ext  %who  %who_ls  %whos  %xdel  %xmode

        Available cell magics:
        %%!  %%HTML  %%SVG  %%bash  %%capture  %%cmd  %%debug  %%file  %%html  %%javasc
        ript  %%latex  %%perl  %%prun  %%pypy  %%python  %%python2  %%python3  %%ruby
         %%script  %%sh  %%svg  %%sx  %%system  %%time  %%timeit  %%writefile

        Automagic is ON, % prefix IS NOT needed for line magics.
```

Defining functions

In [10]:
```
def fib(n):
    """
    Return a list of Fibonacci numbers
    """
    f0, f1 = 0, 1
    f = [1] * n

    # Loop from 1 to n-1
    for i in range(1, n):
        f[i] = f0 + f1
        f0, f1 = f1, f[i]

    return f

print(fib(10))
```

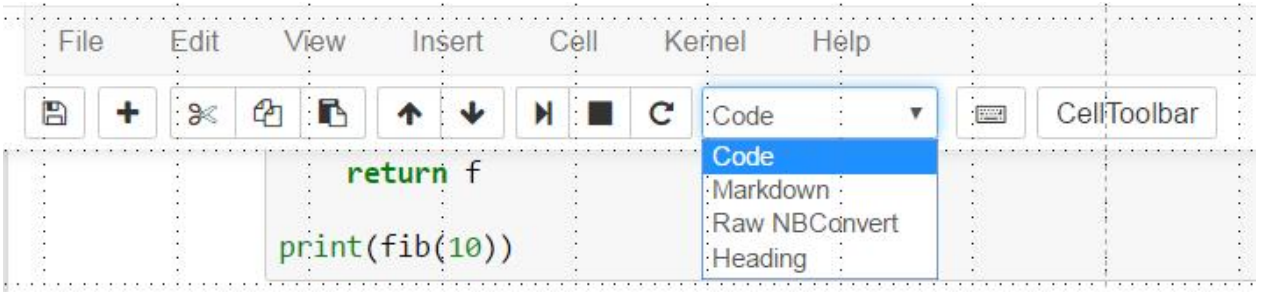
```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

Watch timing the code

```
In [11]: %timeit fib(1000)
         1000 loops, best of 3: 227 µs per loop

In [12]: %time results = fib(10000)
         Wall time: 13 ms

In [13]: %prun fib(10000)
```

```
         4 function calls in 0.019 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.018    0.018    0.018    0.018 <ipython-input-10-c3b554856af0>:1(fib)
        1    0.001    0.001    0.019    0.019 <string>:1(<module>)
        1    0.000    0.000    0.019    0.019 {built-in method builtins.exec}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Cell types



Code

Any Python code. Press Shift+Enter to send the code to the kernel. The results are sent back to the browser

Markdown

Contains marked-up plain text, which is interpreted using Markdown Language and HTML (and Latex)

Raw

A raw text cell, displayed without any interpretation

Headings

Heading cell, from level 1 to 6 (#, ##, ###, ….)

Markdown cells

Function	Syntax by example
Italics	`*text*`
Bold	`**text**`
Strike-through	`~~text~~`
Fixed-width font	`` `text` ``
URL	`[URL text](http://www.example.com)`
New paragraph	Separate the text of two paragraphs with an empty line.

Verbatim	Lines that start with four blank spaces are displayed as-is, without any further processing, using a fixed-width font. This is useful for code-like text segments.
	`␣␣␣␣def func(x):`
	`␣␣␣␣ return x ** 2`

Table	`| A | B | C |`
	`|---|---|---|`
	`| 1 | 2 | 3 |`
	`| 4 | 5 | 6 |`

Horizontal line	A line containing three dashes is rendered as a horizontal line separator: `---`

Heading	`# Level 1 heading`
	`## Level 2 heading`
	`### Level 3 heading`
	`...`

Block quote	`Lines that start with a '>' are rendered as a block quote.`
	`> Text here is indented and offset`
	`> from the main text body.`

Unordered list	`* Item one`
	`* Item two`
	`* Item three`

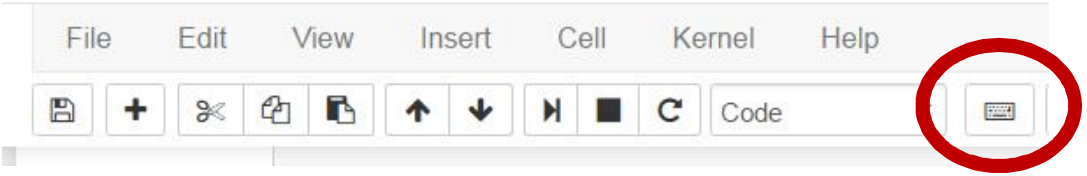
Markdown cells (continued)

Function	Syntax by example

Ordered list	`1. Item one`
	`2. Item two`
	`3. Item three`

Image	`![Alternative text](image-file.png)`[9] or `![Alternative text](http://www.example.com/image.png)`
Inline LaTeX equation	`$\LaTeX$`
Displayed LaTeX equation (centered, and on a new line)	`$$\LaTeX$$` or `\begin{env}...\end{env}` where `env` can be a LaTeX environment such as `equation`, `eqnarray`, `align`, etc.

Keyboard shortcuts



a – Create new cell above

b – Create new cell below

c – Copy cell

x – Cut cell

v – Paste cell

m – Convert to markdown

y – Convert to code

h – Help

s – Save notebook

i – i - Interrupt kernel

0 – 0 – Restart kernel

ENTER – Enter edit mode

Esc – Exit edit mode

UP – Previous cell

DOWN – Next cell

Laboratory exercises

Week 1

- Lab 1a. Arrays
- Lab 1b. Plotting and Visualization
- Lab 2. Discrete Random Numbers

Home Assignment 1. Birthday Problem – Random numbers version