UAV Workshop

Python Tools

Prof. Venki Muthukumar, Ph.D.

Anaconda

Anaconda Python/R Distribution

anaconda.com/distribution/

ANACONDA.

Products Why Anaconda? Solutions Resources Company Contact Us Download

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

jupyter spyder NumPy SciPy Numba pandas DASK Bokeh HoloViews Datashader matplotlib scikit-learn $H_2O.ai$ TensorFlow CONDA

Windows | macOS | Linux

This website uses cookies to ensure you get the best experience on our website. Privacy Policy

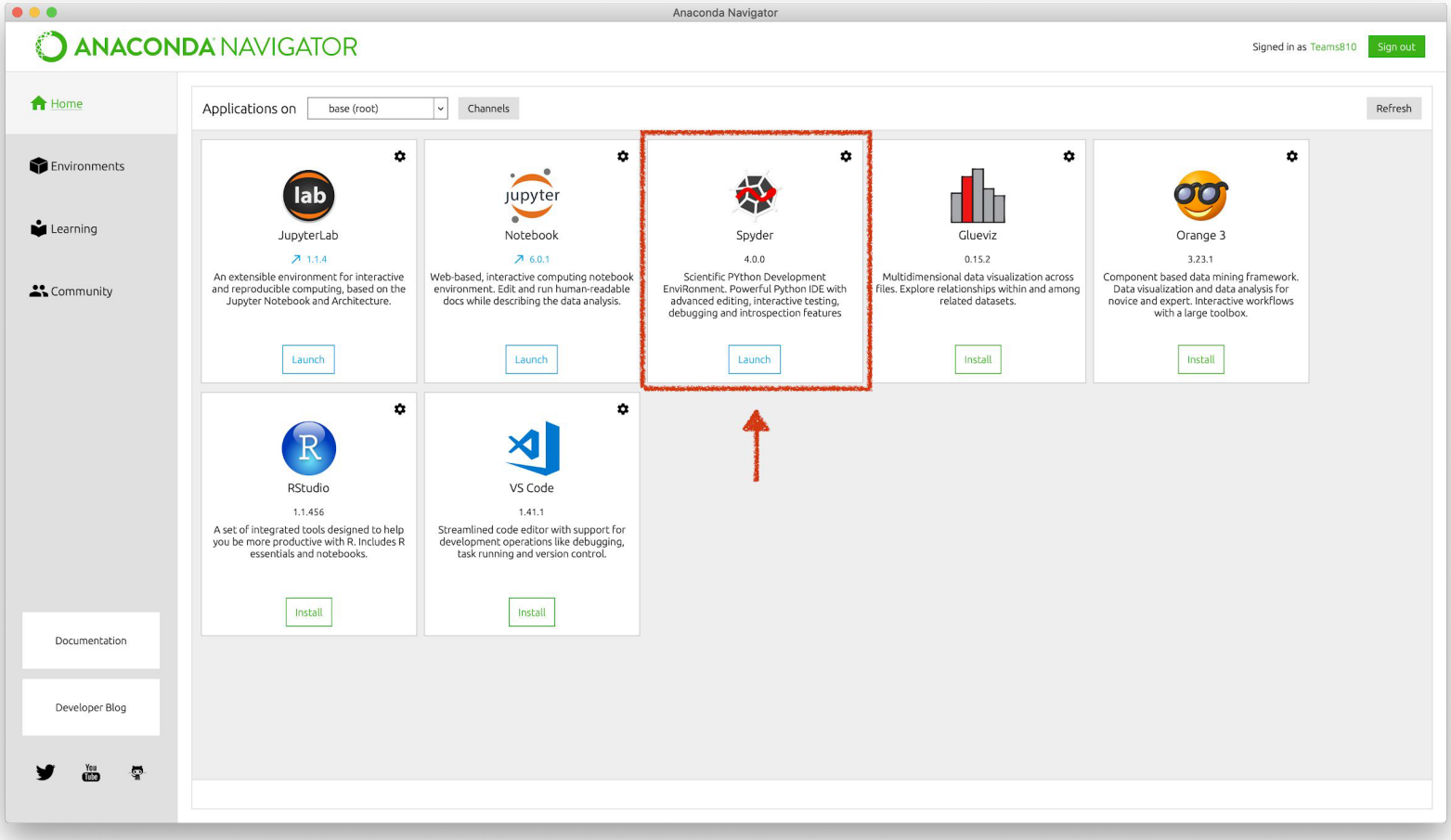
ACCEPT

Other Distributions & IDE

- Python Distributions
  - Active Python, Anaconda, Enthought Canopy, WinPython, python(x,y), etc.
- IDEs
  - NINJA, Spyder, PyCharm, eclipse, Django, pyscripter, BeeWare, Atom, Sublime, VisualCode,

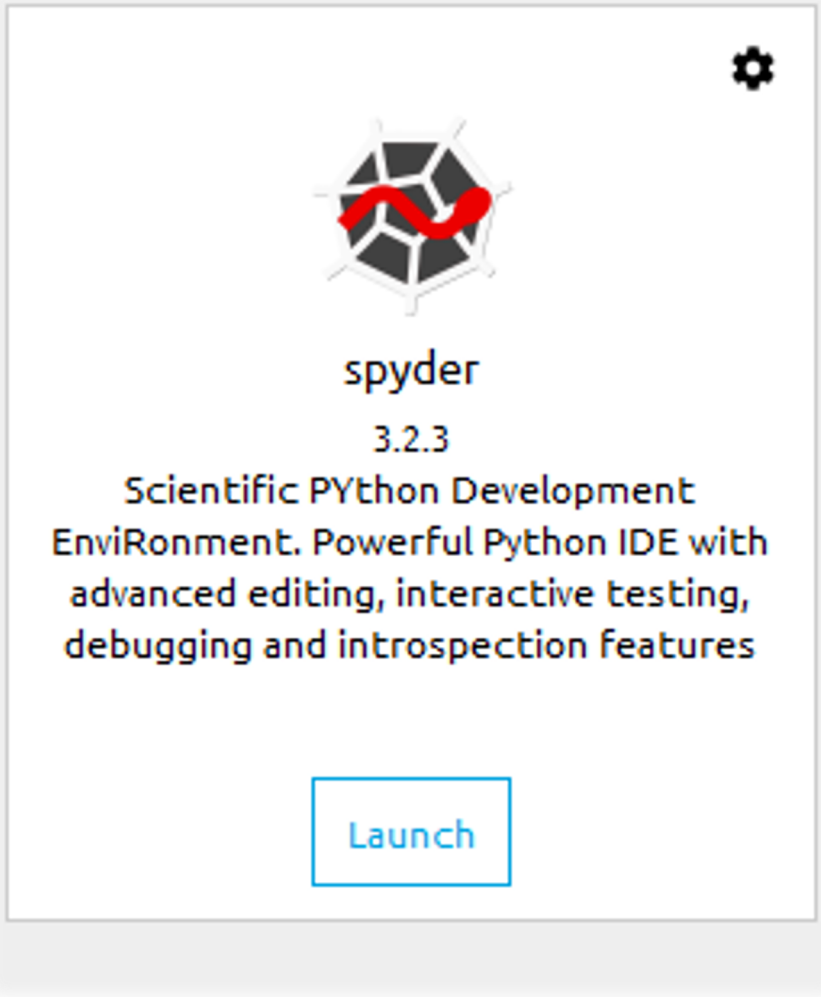
Anaconda Navigator

UAV Workshop



SPYDER IDE

SPYDER



SPYDER IDE

UAV Workshop

What is Spyder?

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language

SPYDER IDE

UAV Workshop

Spyder (Python 3.7)

/Users/victor/.spyder-py3/temp.py

temp.py

```
538
539     @contextmanager
540     def block_update_properties(self):
541         self._update_properties = False
542         yield
543         self._update_properties = True
544
545     def _update_coordinates(self):
546         """Insert the cursor position into the correct position in the
547         tuple of indices and update the cursor coordinates.
548         """
549         coords = list(self.dims.indices)
550         for d, p in zip(self.dims.displayed, self.position):
551             coords[d] = p
552         self.coordinates = tuple(coords)
553         self._value = self.get_value()
554         self.status = self.get_message()
555
556     def get_message(self):
557         """Generate a status message based on the coordinates and value
558
559         Returns
560         ----------
561         msg : string
562             String containing a message that can be used as a status update.
563         """
564         full_scale = np.multiply(self.scale, self._scale_view)
565         full_translate = np.add(self.translate, self._translate_view)
566
567         if any(
568             isinstance(n, slice) for n in self.coordinates
569         ):  # fixes bug where any element of self.coordinates is updated to a non-nu
570             self.coordinates = (0,) * len(self.coordinates)
571         full_coord = np.round(
572             np.multiply(self.coordinates, full_scale) + full_translate
573         ).astype(int)
574
575         msg = f'{self.name} {full_coord}'
576
577         value = self._value
578
579         if value is not None and not np.all(value == (None, None)):
580             msg += ': '
581             if type(value) == tuple:
582                 msg += status_format(value[0])
583                 if value[1] is not None:
```

/Users/victor

Source | Console | Object

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.11.1 -- An enhanced Interactive Python.

In [1]:
```

IPython console | History

conda: base (Python 3.7.3) | Line 556, Col 27 | ASCII | LF | RW | Mem 53%

Spyder plugins

UAV Workshop

spydernotebook | GuitarModeling

Edit View Run Kernel

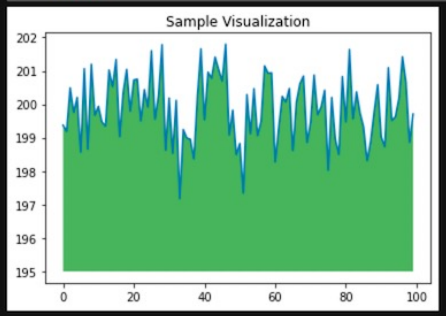
Code | Python 3 (Spyder)

[4]:
```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Sample Visualization")
plt.show()
```



[6]:
```
#----------------------Parameters of the Simulation----------------------
n = 1
L = 0.6477  # String's length (m
d = 0.000309  # String's density E4 (kg/m)
#d = 0.00578 # String's density E2 (kg/m)
F = 56.34
#F = 65.9 # Tension (N)
c = np.sqrt(F/d) # Speed of waves in the string
frequency = (n*c)/(2*L)
Cn = 0.000005
t_0 = 0 #Initial time
deltaT = 0.0001
print(frequency)
```

Editor | Notebook

/Users/juanitagomez/.spyder-py3

Name	Type	Size	Value
c	float64	1	427.00128464371005
Cn	float	1	5e-06
d	float	1	0.000309
deltaT	float	1	0.0001
F	float	1	56.34
frequency	float64	1	329.6289058543384
L	float	1	0.6477
n	int	1	1
t_0	int	1	0
x	list	100	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...]
ys	Array of float64	(100,)	[199.3743578 199.19178079 200.49207342 ... 200.65184634 198.86255419 ...

Variable explorer | Help | Plots | Files

Console 1/A | GuitarModeling/A

```
In [7]: runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')

In [8]: runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')

In [9]: runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')

In [9]:

In [10]: print(d)
0.000309

In [11]: ys
Out[11]:
array([199.3743578 , 199.19178079, 200.49207342, 199.75735424,
       200.20556248, 198.57028675, 201.06828343, 198.65799977,
       201.19817929, 199.67374033, 199.93721577, 199.47208686,
       199.34621786, 201.02758643, 200.52863798, 201.34186833,
       199.03202734, 200.35505824, 201.04438294, 199.79334461,
       200.72666158, 200.75225926, 199.50278834, 200.43751099,
       199.91829586, 201.60293197, 199.55369949, 200.26563137,
       201.78436574, 198.62443275, 200.18715755, 198.5347672 ,
       200.11933579, 197.17840466, 199.24536681, 198.99098274,
       198.95218458, 198.37014096, 200.24552493, 201.65915333,
       199.53841793, 200.96105668, 200.77946815, 201.41185544,
       201.04232715, 200.69797569, 201.80026229, 199.08296063,
       199.82492053, 198.4945493 , 198.82823873, 197.34196339,
```

IPython console | History

LSP Python: ready | conda: spyder-dev-4 (Python 3.9.5) | Line 1, Col 1 | UTF-8 | LF | RW | Mem 65%

Installing Spyder Plugins

- Try
  - conda install spyder-notebook -c spyder-ide
  - conda install spyder-terminal -c spyder-ide

- else try
  - conda install spyder-notebook -c conda-forge
  - conda install -c conda-forge spyder-terminal

The Notebook

A web-based interactive computing interface and platform that combines code, equations, text and visualisations.

jupyter

http://www.jupyter.org

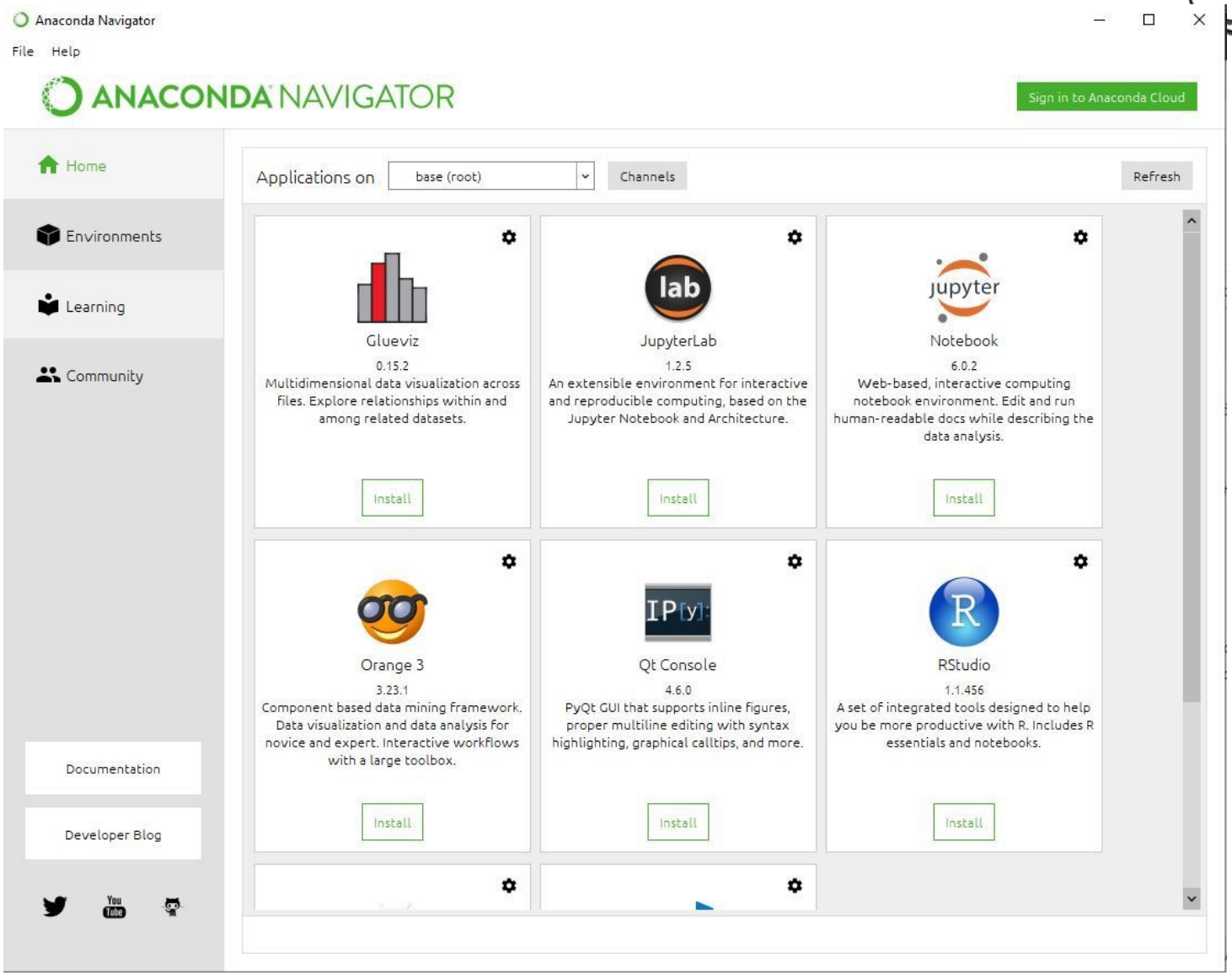
Many supported languages: Python, Haskell, Julia, R … One generally speaks about a "kernel" for a specific language

No excuses possible when it comes to describe all steps in an analysis!

In a nutshell: an "interactive shell opened within the browser"

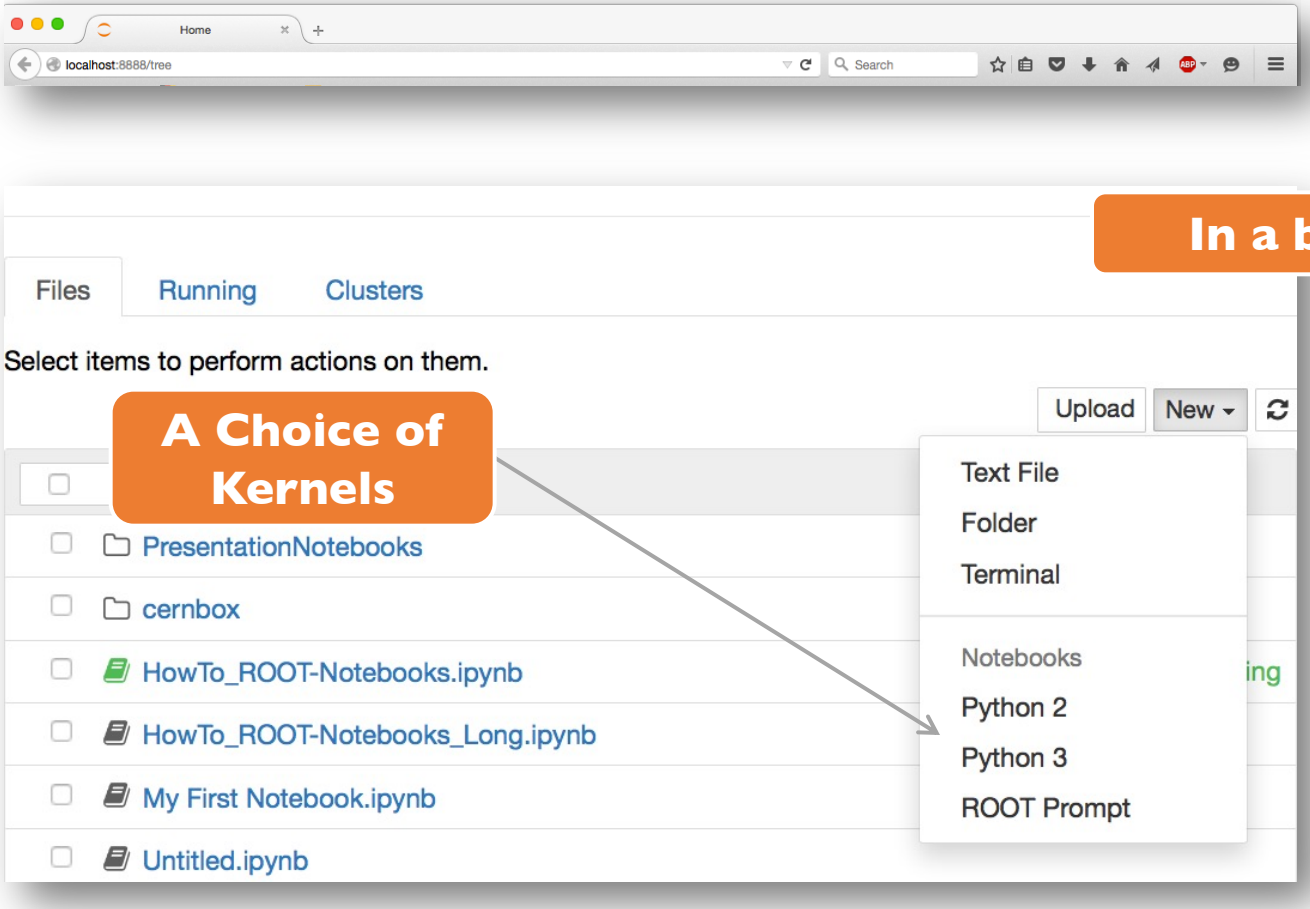
Also called:
"Jupyter Notebook" or "IPython Notebook"

Anaconda - Ipython

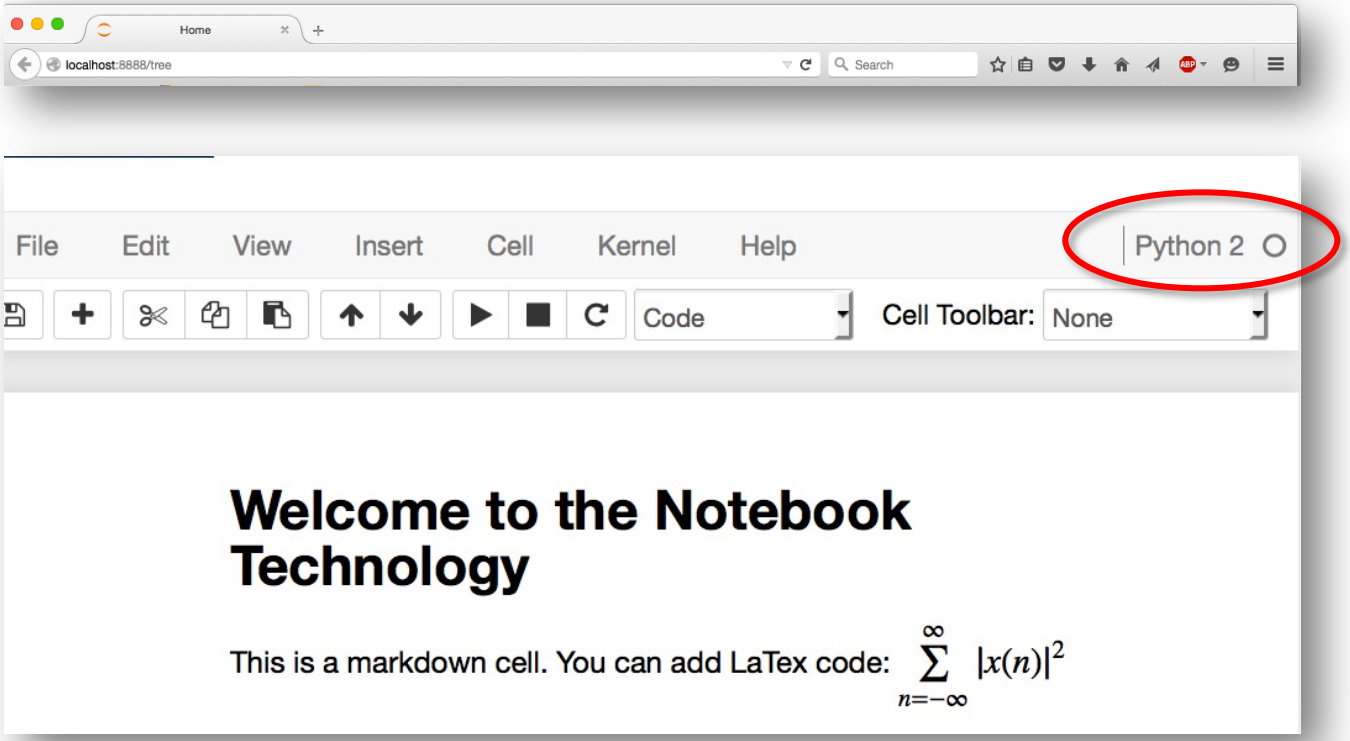


UAV Workshop

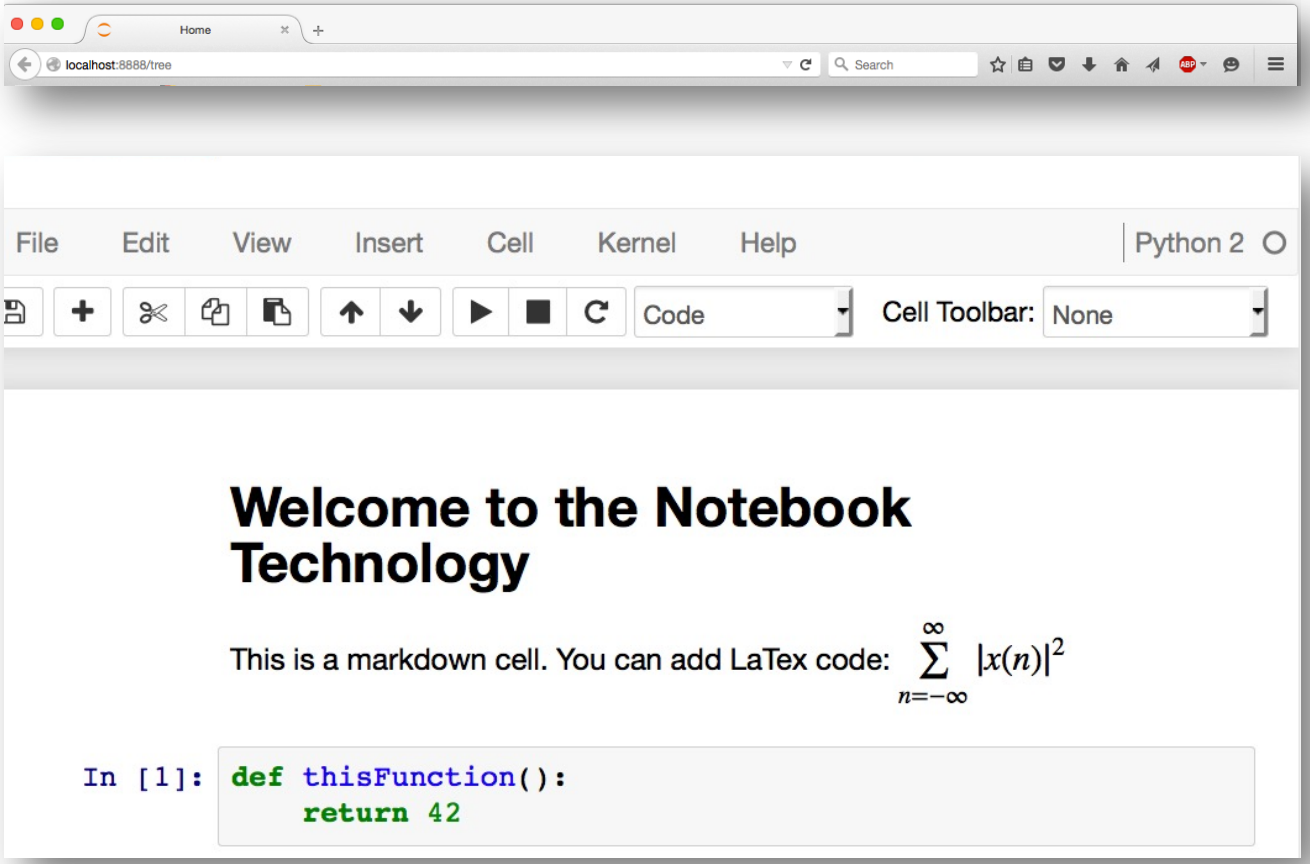
Jupyter Notebook



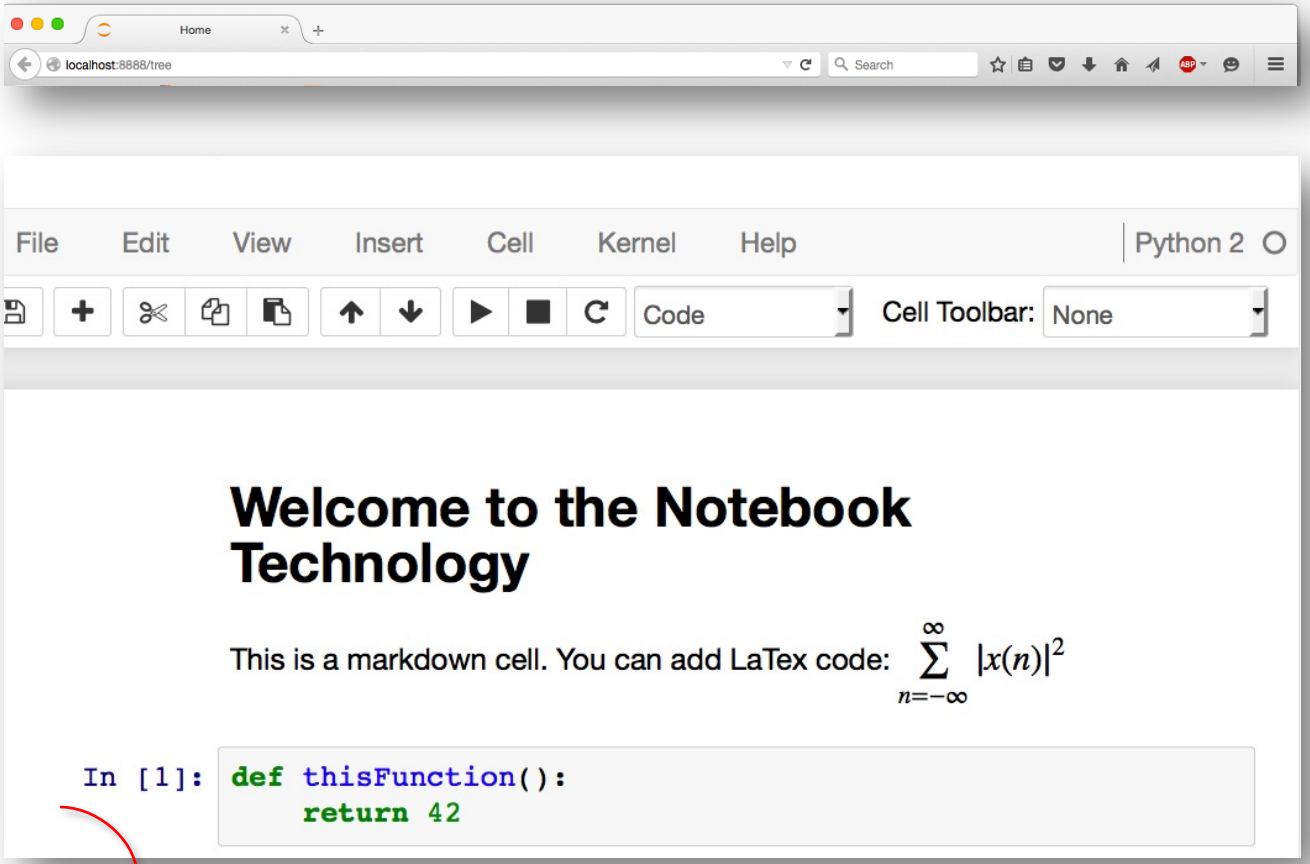
Kernels are processes that run interactive code in a particular programming language and return output to the user. Kernels also respond to tab completion and introspection requests.



Text and Formulas

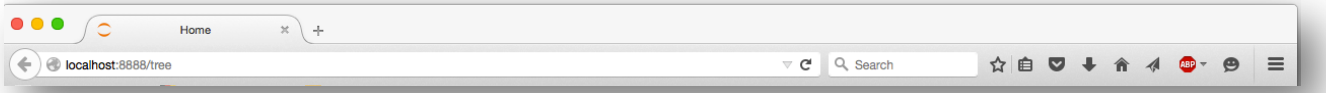

Home
localhost:8888/tree
Search
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar:
None
Welcome to the Notebook Technology
This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$
In [1]:
def thisFunction():
return 42

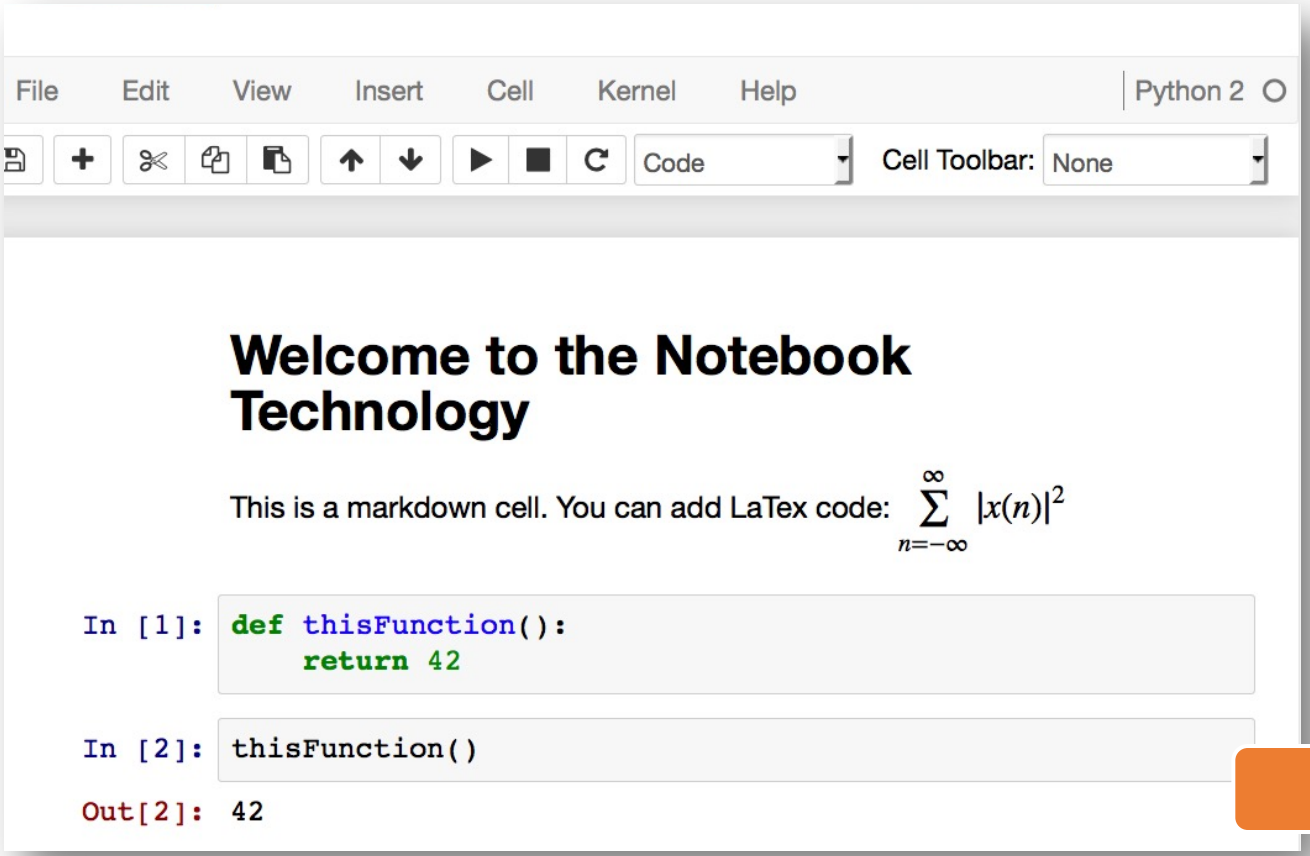
Code



This is a notebook in Python

Code


Home
localhost:8888/tree
Search


File Edit View Insert Cell Kernel Help
Python 2
Code
Cell Toolbar: None
Welcome to the Notebook Technology
This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$
In [1]:
def thisFunction():
return 42
In [2]:
thisFunction()
Out[2]:
42

Code

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

In [1]:
```
def thisFunction():
    return 42
```

In [2]:
```
thisFunction()
```

Out[2]: 42

In [3]:
```
%%bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
> SF.jpg
```

Shell Commands

We can invoke commands in the shell…

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
            return 42

In [2]: thisFunction()
Out[2]: 42

In [3]: %%bash
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
        > SF.jpg
```

```
  % Total    % Received % Xferd  Average Speed   Time
  Time     Time  Current
                                 Dload  Upload   Total
  Spent    Left  Speed
100  128k  100  128k    0     0  2731k      0 --:--:--
--:--:-- --:--:-- 2787k
```

Shell Commands

… And capture their output

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: $\sum_{n=-\infty}^{\infty} |x(n)|^2$

In [1]:
```
def thisFunction():
    return 42
```

In [2]:
```
thisFunction()
```

Out[2]:
```
42
```

In [3]:
```
%%bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
> SF.jpg
```

```
  % Total    % Received % Xferd  Average Speed   Time
  Time     Time  Current
                                 Dload  Upload   Total
  Spent    Left  Speed
100  128k  100  128k    0     0  2731k      0 --:--:--
--:--:-- --:--:-- 2787k
```

In [4]:
```
from IPython.display import Image
Image(filename="./SF.jpg",width=225)
```

In [3]:
```
%%bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
> SF.jpg
```

```
  % Total    % Received % Xferd  Average Speed   Time
  Time     Time  Current
                                 Dload  Upload   Total
  Spent    Left  Speed
100  128k  100  128k    0     0  2731k      0 --:--:--
--:--:-- --:--:-- 2787k
```

In [4]:
```
from IPython.display import Image
Image(filename="./SF.jpg",width=225)
```

Out[4]:

Images

In a browser

Text and Formulas

```
In [2]: thisFunction()

Out[2]: 42
```

Code

```
> SF.jpg
  % Total    % Received % Xferd  Average Speed   Time
  Time     Time  Current
                                 Dload  Upload   Total
  Spent    Left  Speed
100  128k  100  128k    0     0  2731k      0 --:--:--
--:--:-- --:--:-- 2787k
```

Shell Commands

```
In [4]: from IPython.display import Image
        Image(filename="./SF.jpg",width=225)

Out[4]:
```

Images

UAV Workshop

JupyterLab

- JupyterLab is the next generation of the Jupyter Notebook.
- JupyterLab offers a general framework for interactive computing and data science in the browser, using Python, Julia, R, or one of many other languages.