

# Welcome! Please do the following...

1. Sit close to the front
2. If you have not done the prerequisites, do them now! (see link [1])
3. On your computer, open...
  1. A normal command prompt (Windows)/terminal (linux/unix)
  2. An Anaconda prompt
4. Download all files from link [1] (or clone entire repo)

silent mode,  
please!



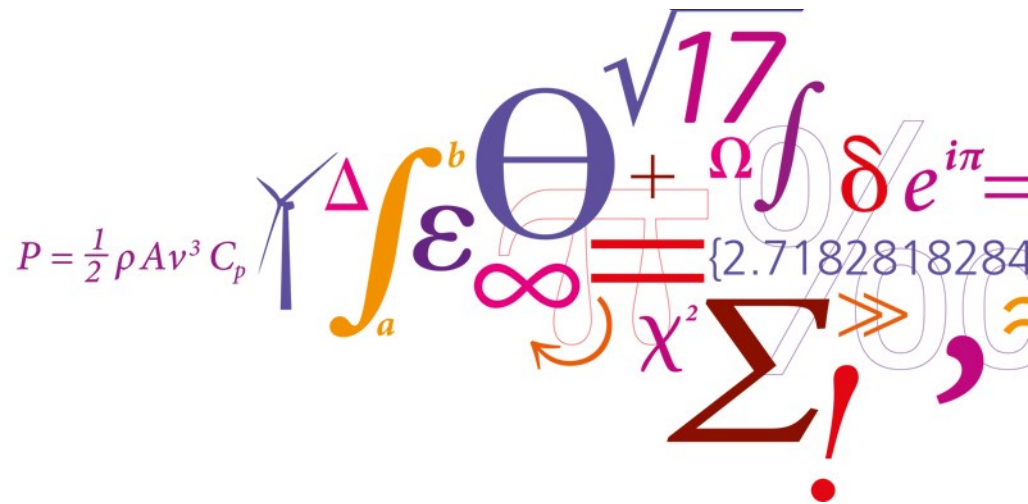
[1] <https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

# Scientific Python Workshop #1:

## Introduction to Anaconda

September 27, 2017  
DTU Risø – H. H. Koch

Jenni Rinker



# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up

# Workshop outline

## **13:00 – 13:05**

## **Preliminaries**

13:05 – 13:20

What is Python/Anaconda

13:20 – 13:40

Anaconda Prompt

13:40 – 14:00

Terminal Python

14:00 – 14:10

Break

14:10 – 14:30

Spyder

14:30 – 14:50

Jupyter

14:50 – 15:00

Wrap-Up

# I am...

Jenni Rinker  
Postdoctoral Researcher  
Loads and Controls

and...

# I am...

Jenni Rinker  
Postdoctoral Researcher  
Loads and Controls

and...

**\*\*\*NOT AN EXPERT\*\*\***

# Workshop history: born of a summer school, seek to “unite” researchers here at DTU

*“[scientists] spend far too much time writing deficient code and reinventing the wheel”*



<https://python.g-node.org/wiki/>

## Rough workshop outline (*subject to change*)

1. Introduction to Anaconda – 27. sep 13:00
2. Getting started with Python – 4. okt 10:00
3. Collaborating with Python– 17. okt 10:00
4. How to speed up your code – 2. nov 10:00
5. Test-driven development and documentation – 16. nov 13:00
6. Python tools/resources here at DTU – 21. nov 10:00

# Workshop history: born of a summer school, seek to “unite” researchers here at DTU

*“[scientists] spend far too much time writing deficient code and reinventing the wheel”*



## Rough workshop outline (subject to change)

1. Introduction to Anaconda – 27. sep 13:00
2. Getting started with Python – 4. okt 10:00
3. Collaborating with Python– 17. okt 10:00
4. How to speed up your code – 2. nov 10:00
5. Test-driven development and documentation – 16. nov 13:00
6. Python tools/resources here at DTU – 21. nov 10:00

Objective: get you familiar with what Anaconda is and how to use it (so no coding today!)

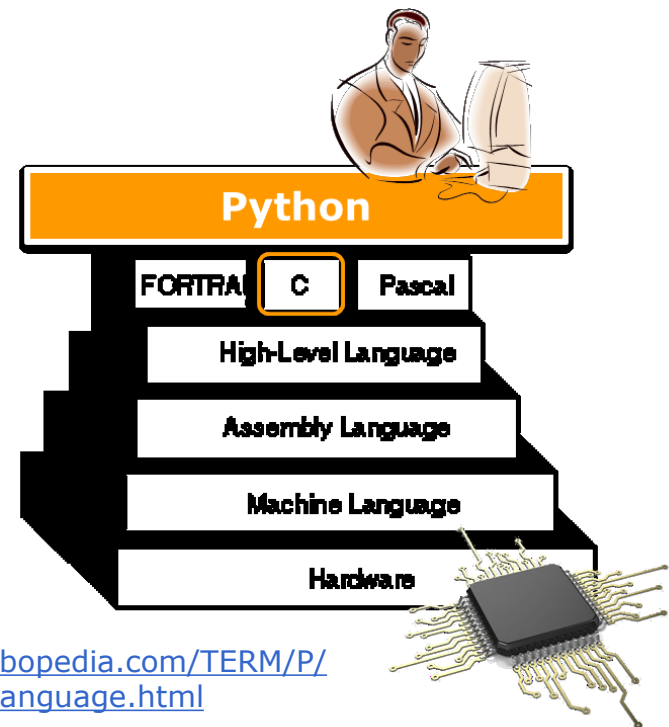


# Workshop outline

13:00 – 13:05	Preliminaries
<b>13:05 – 13:20</b>	<b>What is Python/Anaconda</b>
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up

# Python is a programming language known for simplicity, readability, and collaborative code-sharing philosophy

- Interpreted
- Object-oriented
- Dynamic typing
- Dynamic binding



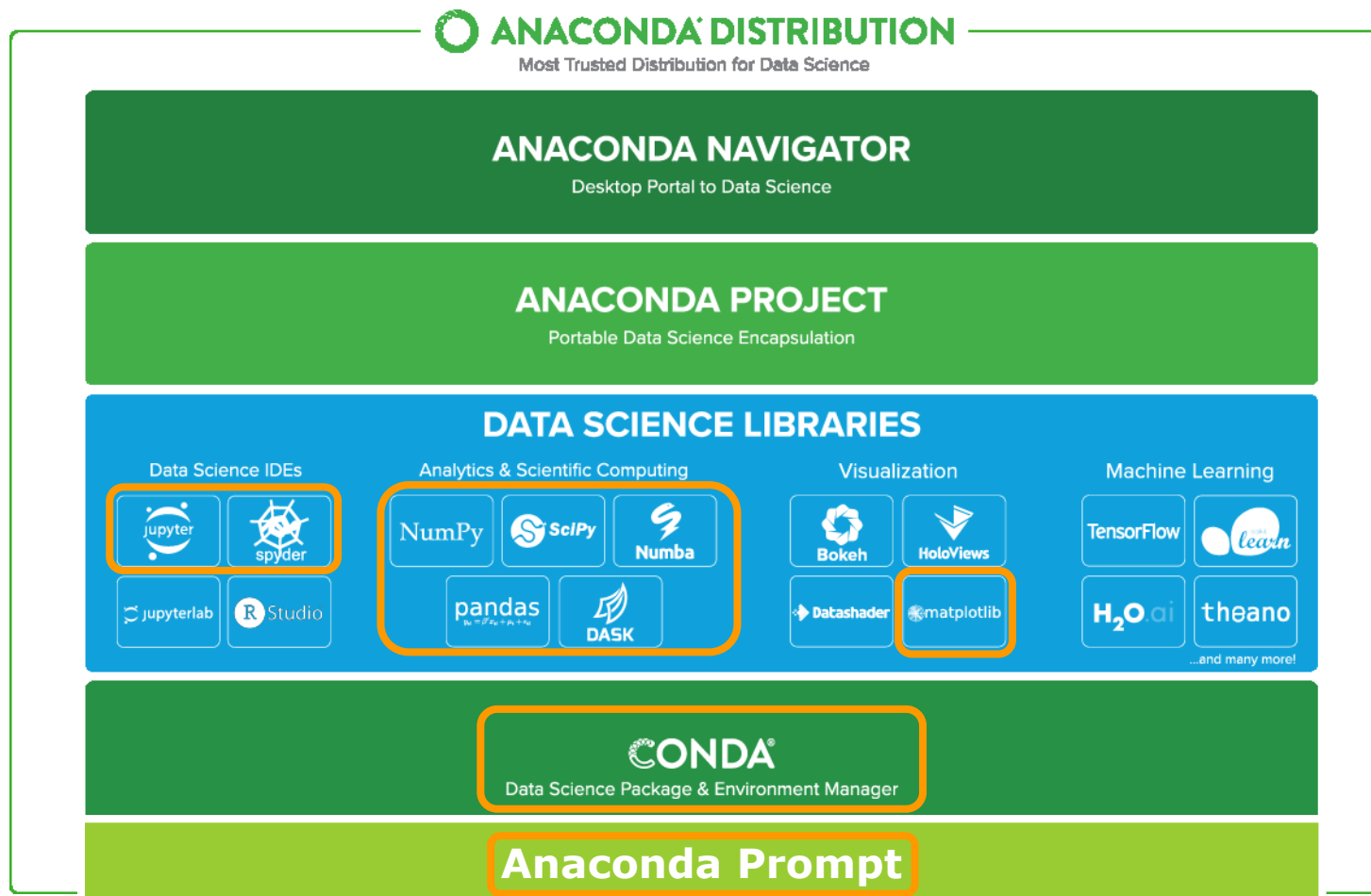
[http://www.webopedia.com/TERM/P/programming\\_language.html](http://www.webopedia.com/TERM/P/programming_language.html)

# Anaconda is a cross-platform distribution of Python

- Good for science
- Good for Windows
- Can do a light install with miniconda
- [And many other reasons...](#)

# Anaconda is a cross-platform distribution of Python

<https://www.anaconda.com/distribution/>



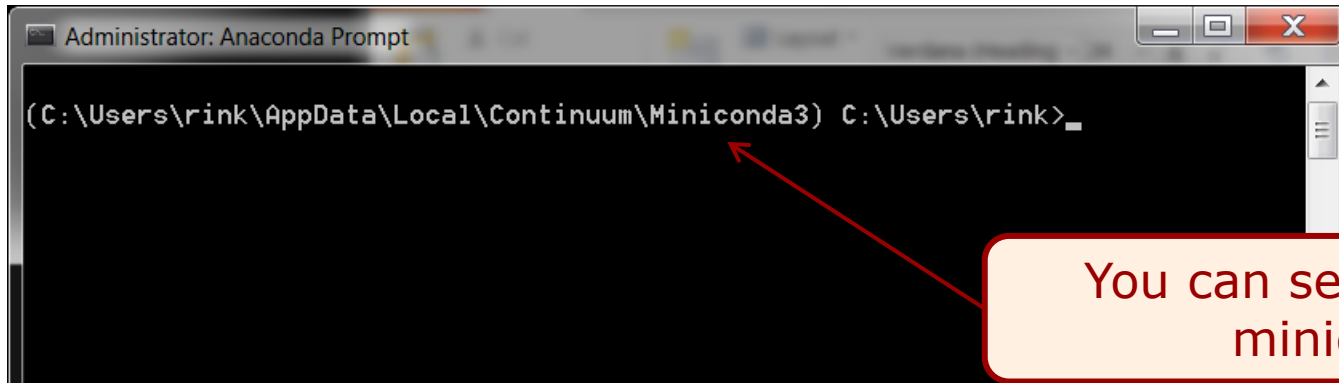
# Status update request



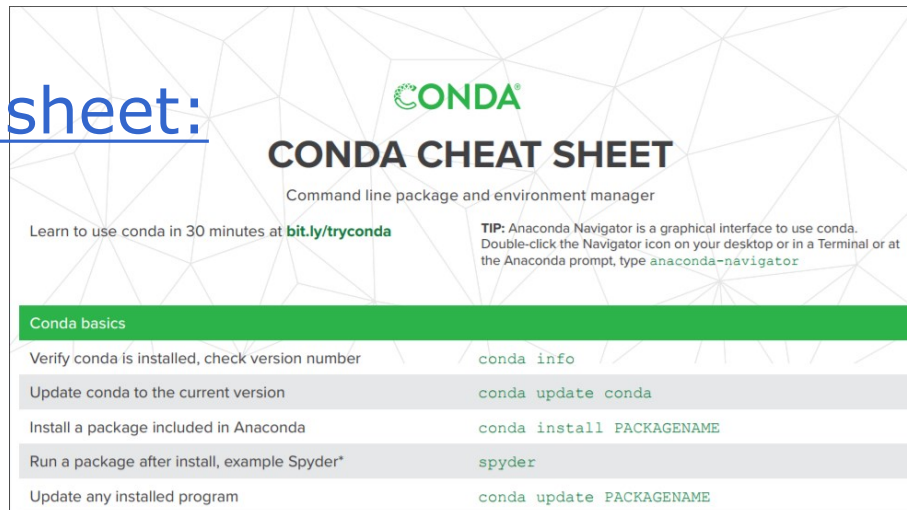
# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
<b>13:20 – 13:40</b>	<b>Anaconda Prompt</b>
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up

# Go to your Anaconda Prompt



## conda cheat sheet:



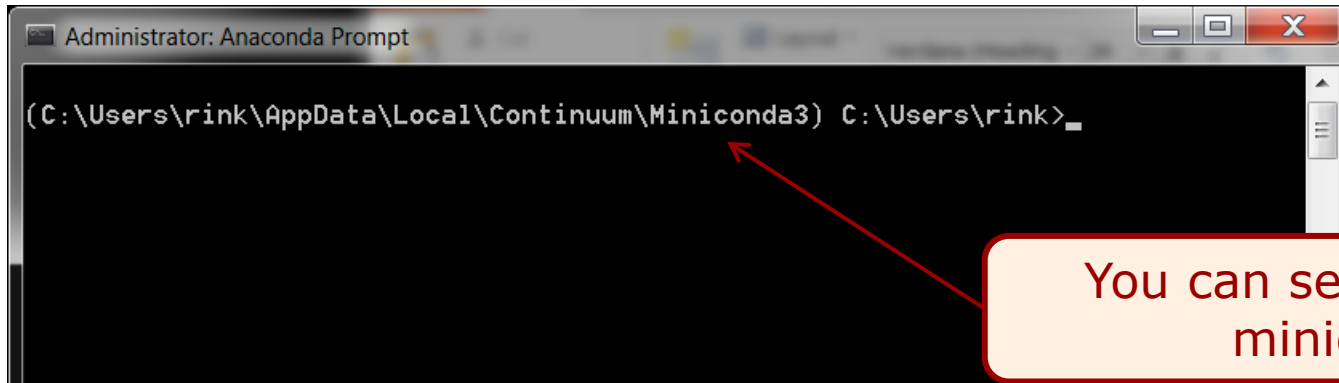
**CONDA**  
**CONDA CHEAT SHEET**  
 Command line package and environment manager

Learn to use conda in 30 minutes at [bit.ly/tryconda](https://bit.ly/tryconda)

**TIP:** Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type `anaconda-navigator`

Conda basics	
Verify conda is installed, check version number	<code>conda info</code>
Update conda to the current version	<code>conda update conda</code>
Install a package included in Anaconda	<code>conda install PACKAGENAME</code>
Run a package after install, example Spyder*	<code>spyder</code>
Update any installed program	<code>conda update PACKAGENAME</code>

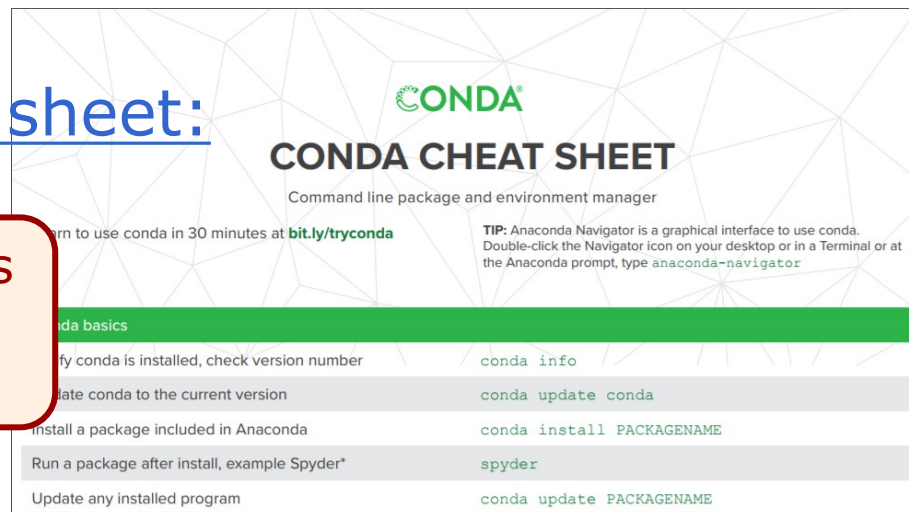
# Go to your Anaconda Prompt



You can see I installed miniconda

## conda cheat sheet:

I could go through this with you step-by-step...but I won't





# Exercise: creating an environment

## SCENARIO



"joe\_bob.py"

Python 2.7

NumPy 1.11

<https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

# Exercise: creating an environment

## SCENARIO



"joe\_bob.py"

Python 2.7

NumPy 1.11

<https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

Quick exercise: What  
do you think  
joe\_bob.py does?

# Exercise: creating an environment

## SCENARIO



"joe\_bob.py"

Python 2.7

NumPy 1.11

<https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

### **Someone try this:**

In your Anaconda prompt, navigate to folder with .py file and type:

```
python joe_bob.py
```

What happens?

# Exercise: creating an environment

## SCENARIO



"joe\_bob.py"

Python 2.7

NumPy 1.11

<https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

### Someone try this:

In your Anaconda prompt, navigate to folder with .py file and type:

```
python joe_bob.py
```

What happens? **SYNTAX ERROR!**

# Exercise: creating an environment

## SCENARIO



"joe\_bob.py"

Python 2.7

NumPy 1.11

<https://gitlab.windenergy.dtu.dk/python-at-risoe/scientific-python-workshops/1-intro-to-conda>

How do we handle this?  
Through a conda  
***environment*** with installed  
***packages***

### Someone try this:

In your Anaconda prompt, navigate to folder with .py file and type:

```
python joe_bob.py
```

What happens? **SYNTAX ERROR!**

# But first...a few definitions

## Python Package

- Essentially a collection of functions/scripts/etc. that are bundled and distributed together
- Think “Matlab toolbox”, except everyone can make one

## Conda Environment

- A mini-world within conda with a specific Python version and a collection of packages
- Think a specific instantiation of a Matlab version and relevant packages

# Exercise: creating a custom environment

1. Create environment with needed python version
2. Add necessary packages to environment
3. Activate that environment
4. Run `joe_bob.py` (`python joe_bob.py`)
  - DOES IT WORK? What's the output?
5. (deactivate environment)

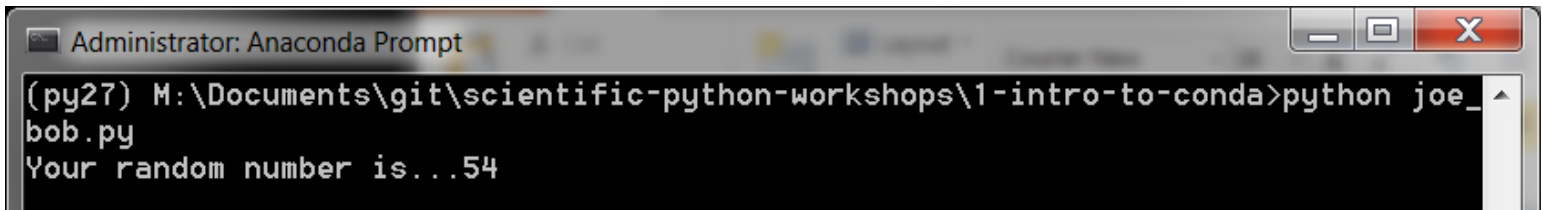
[conda cheat sheet:](#)

***Do this with to the  
person next to you!  
Use the cheat sheet!***

# Exercise: creating a custom environment

## Solution:

1. `conda create --name py27 python=2.7`
2. `activate py27`
3. `conda list`
4. `conda install numpy=1.11`
5. `python joe_bob.py`



Administrator: Anaconda Prompt

```
(py27) M:\Documents\git\scientific-python-workshops\1-intro-to-conda>python joe_bob.py
Your random number is...54
```



# Status update request



# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
<b>13:40 – 14:00</b>	<b>Terminal Python</b>
14:00 – 14:10	Break
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up

# Three ways to run python from the terminal

1. Call python interpreter on the script

```
> python joe_bob.py
```

activate py27  
python joe\_bob.py

2. Enter interpreter interactively

```
> python (command prompt or Anaconda prompt)
```

```
> ipython (Anaconda prompt)
```

3. Make script executable

```
> joe_bob.py
```

Requires some set-up on  
Windows...you'll have to  
Google it

# IPython vs. python interpreter

- IPython is a console with a read-eval-print loop (REPL)
- IPython has tab completion

IPython is  
python...plus stuff

```
Python 3.6.1 [Continuum Analytics, Inc.] (default, May 11 2017, 13:25:24) [MSC v
.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import num
         numbers  numpy
         numexpr  numpydoc
```

# Exercise: “Hello, world!” in python, ipython, and called by interpreter

1. In a python interpreter, figure out how to print “Hello, world!” in Python 3
2. Do Step 1 but for an IPython interpreter
3. Now create a file called “hello\_world.py” that can be called by `python hello_world.py` and prints “Hello, world!”

***Do this with to the person next to you!***

# Exercise: “Hello, world!” in python, ipython, and called by interpreter

## SOLUTIONS

1. `print('Hello, world!')`

2. `print('Hello, world!')`

3. `print('Hello, world!')` (in file `hello_world.py`)

# Workshop outline



13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
<b>14:00 – 14:10</b>	<b>Break</b>
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up

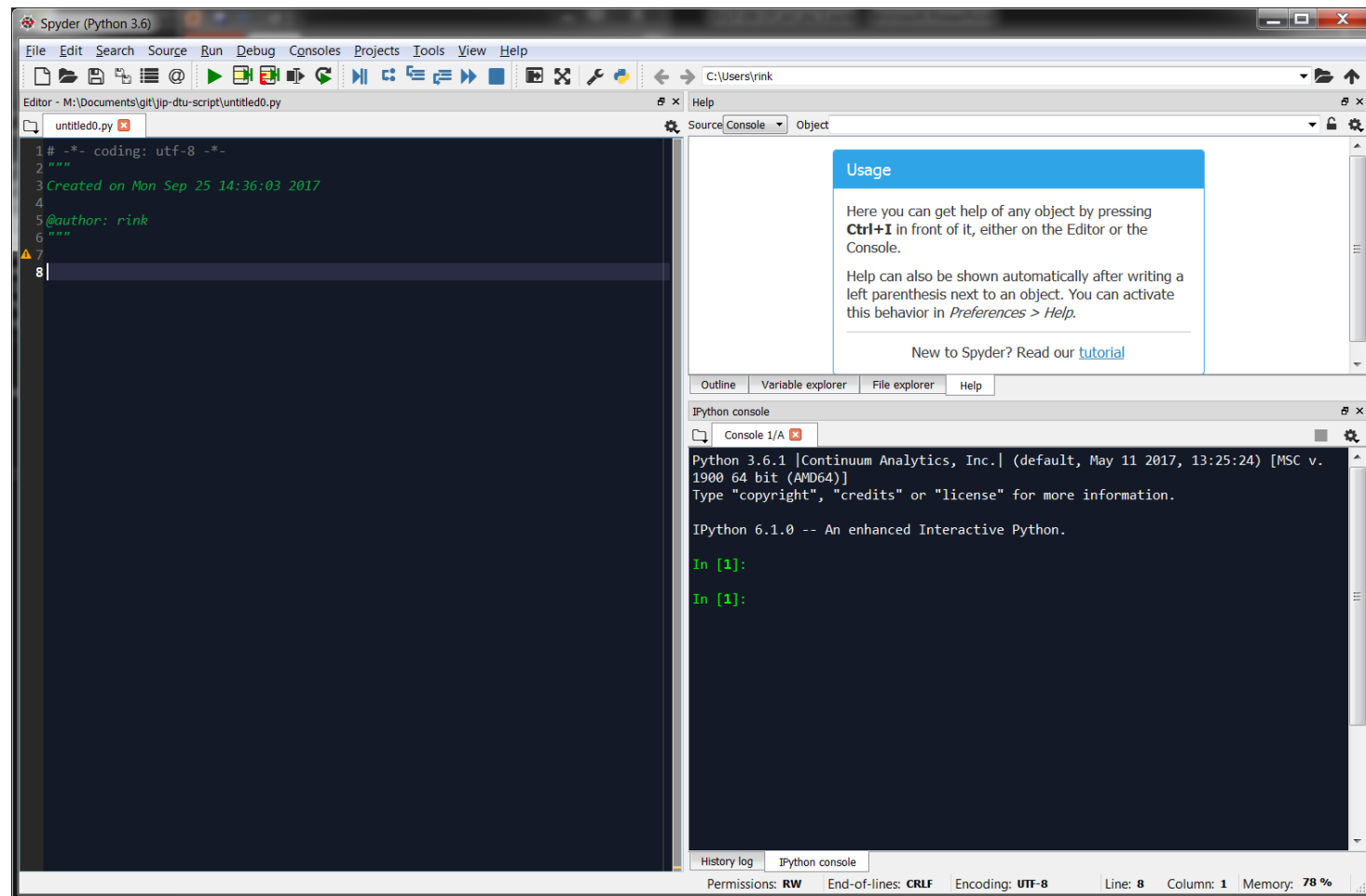


# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
<b>14:10 – 14:30</b>	<b>Spyder</b>
14:30 – 14:50	Jupyter
14:50 – 15:00	Wrap-Up



# Spyder is basically Matlab





# A few random tips

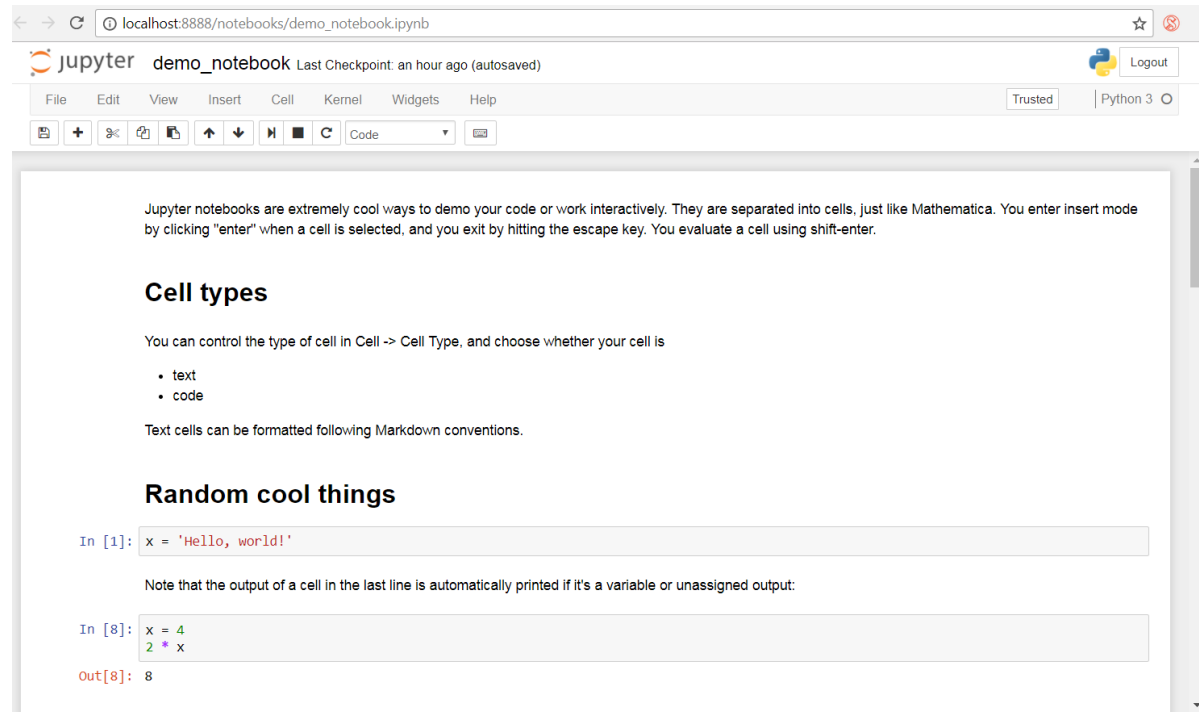
- Launch by clicking on icon, through Anaconda Navigator, or `spyder` in Anaconda prompt
- Be sure to update conda/spyder fairly frequently (`conda update conda, conda update spyder` from Anaconda prompt)
- Add places to path using PYTHONPATH manager
- Syntax coloring: Tools -> Preferences -> Syntax coloring
  - Or “preferences” button in toolbar (looks like wrench)
- PEP8 flags: Preferences -> Editor -> Real-time code style analysis
- Code annotations (BUG, TODO, FIXME, etc...)
- Drag-and-drop windows
- Debugging just like in Matlab
- Run script with F5 (shortcuts set in preferences)
- Run cell with shift + enter
  - Define cells using “# %%” at beginning of line

# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
14:10 – 14:30	Spyder
<b>14:30 – 14:50</b>	<b>Jupyter</b>
14:50 – 15:00	Wrap-Up

# If Spyder is Matlab...then Jupyter is Mathematica. But better.

- Launch a notebook by clicking icon in Anaconda Navigator or `jupyter notebook` in Anaconda prompt (might need to `conda install jupyter` first)
- (demo of notebook and exercise)



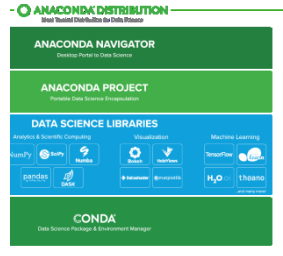
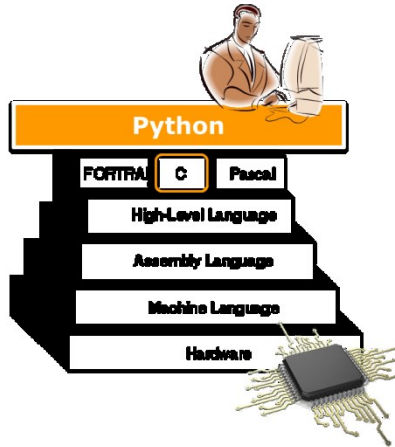
# Status update request



# Workshop outline

13:00 – 13:05	Preliminaries
13:05 – 13:20	What is Python/Anaconda
13:20 – 13:40	Anaconda Prompt
13:40 – 14:00	Terminal Python
14:00 – 14:10	Break
14:10 – 14:30	Spyder
14:30 – 14:50	Jupyter
<b>14:50 – 15:00</b>	<b>Wrap-Up</b>

# Today we learned...



1. What Python and Anaconda are



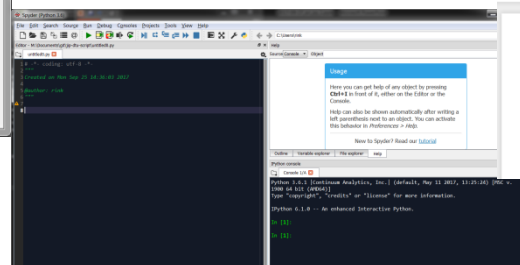
2. How to create environments in conda

3. How to run code from the terminal...

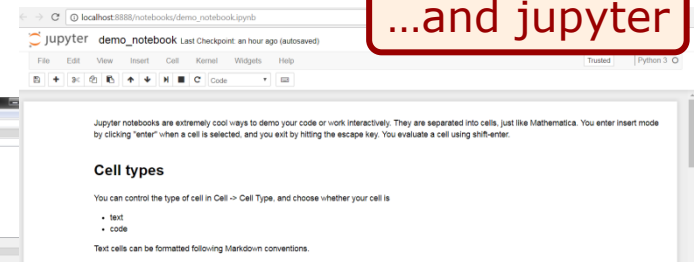
```
Python 3.6.1 [Continuum Analytics, Inc.] (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import num
        numbers numpy
        numexpr  numexpr
```

...spyder...



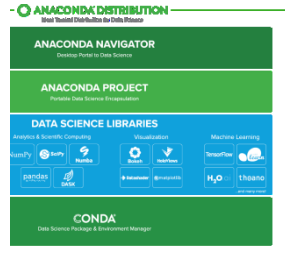
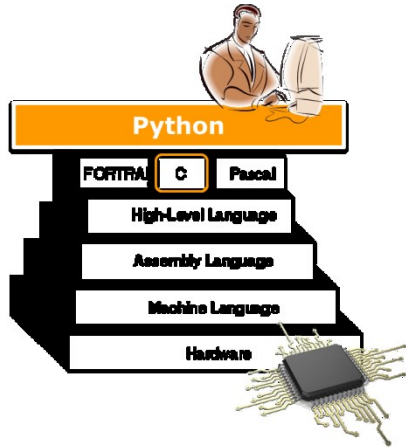
...and jupyter





# Looking forward...what's next?

- Next workshop: **Getting Started with Python**
  - A.k.a., how to actually write code in Python
  - 4. okt 10:00
- Google is your friend
- But so are your colleagues!
  - Python@Risø group on GitLab:  
<https://gitlab.windenergy.dtu.dk/python-at-risoe>
  - [rink@dtu.dk](mailto:rink@dtu.dk)
- Expect a survey from me – please fill it out!



Thanks!

```
Python 3.6.1 [Continuum Analytics, Inc.] (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import num
        numbers numpy
        numexpr  numpydoc
```

