

Python Bootcamp Toolboxes

PhD Bruno C. Quint

bquint@ctio.noao.edu

Resident Astronomer at SOAR Telescope

<https://github.com/b1quint/PythonBootcamp2017>



Table of Contents

- iPython
- Jupyter
- PyCharm



Python Vs iPython

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> █
```

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: █
```



Python Vs iPython

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> █
```

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: █
```



Python Vs iPython

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> █
```

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
```

In [1]: █



Python Vs iPython

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> █
```

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.
```

IPython 5.1.0 -- An enhanced Interactive Python.

```
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: █
```



Python Vs iPython

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> x = 2
>>> y = 5
>>> print x + y
7
>>> █
```



Python Vs iPython

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: x = 2

In [2]: y = 5

In [3]: print x + y
7
In [4]: █
```

Colors!!



Python Vs iPython

Tab Completion

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> x = 2
>>> y = 5
>>> print x + y
7
>>> import numpy
>>> num
```



Tab pressed 3x



Python Vs iPython

Tab Completion

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> x = 2
>>> y = 5
>>> print x + y
7
>>> import numpy
>>> num
```



Tab pressed 3x

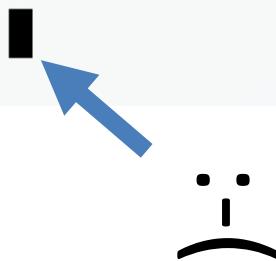


Python Vs iPython

Tab Completion

```
bquint@soarbr3 [~] $ python
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> x = 2
>>> y = 5
>>> print x + y
7
>>> import numpy
>>> num
```

Tab pressed 3x





Python Vs iPython

Tab Completion

```
bquint@soarbr3 [~] $ ipython
Python 2.7.12 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: x = 2

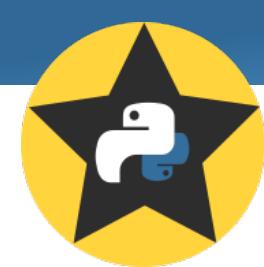
In [2]: y = 5

In [3]: print x + y
7

In [4]: import numpy
In [5]: numpy
```

:::
Tab pressed 1x



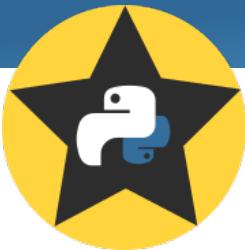


Python Vs iPython

Tab Completion

```
In [6]: import numpy
```

```
In [7]: numpy.■
```



Python Vs iPython

Tab Completion

In [6]: `import numpy`

In [7]: `numpy.`█

<code>numpy.abs</code>	<code>numpy.arange</code>
<code>numpy.absolute</code>	<code>numpy.arccos</code>
<code>numpy.absolute_import</code>	<code>numpy.arccosh</code>
<code>numpy.add</code>	<code>numpy.arcsin</code>
<code>numpy.add_docstring</code>	<code>numpy.arcsinh</code>
<code>numpy.add_newdoc</code>	<code>numpy.arctan</code>
<code>numpy.add_newdoc_ufunc</code>	<code>numpy.arctan2</code>
<code>numpy.add_newdocs</code>	<code>numpy.arctanh</code>
<code>numpy.alen</code>	<code>numpy.argmax</code>
<code>numpy.all</code>	<code>numpy.argmin</code>
<code>numpy.allclose</code>	<code>numpy.argpartition</code>
<code>numpy.ALLOW_THREADS</code>	<code>numpy.argsort</code>
<code>numpy.alltrue</code>	<code>numpy.argwhere</code>
<code>numpy.alterdot</code>	<code>numpy.around</code>
<code>numpyamax</code>	<code>numpy.array</code>
<code>numpy.amin</code>	<code>numpy.array2string</code>
<code>numpy.angle</code>	<code>numpy.array_equal</code>



Python Vs iPython

Help with (i)Python

```
>>> help(numpy.sin)
```

```
In [7]: help(numpy.sin)
```



Python Vs iPython

Help with Python

Help on ufunc object:

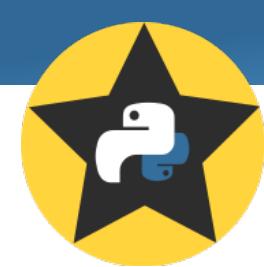
```
sin = class ufunc(__builtin__.object)
| Functions that operate element by element on whole arrays.

| To see the documentation for a specific ufunc, use np.info(). For
| example, np.info(np.sin). Because ufuncs are written in C
| (for speed) and linked into Python with NumPy's ufunc facility,
| Python's help() function finds this page whenever help() is called
| on a ufunc.

| A detailed explanation of ufuncs can be found in the "ufuncs.rst"
| file in the NumPy reference guide.
```

Unary ufuncs:

```
op(X, out=None)
| Apply op to X elementwise
```



Python Vs iPython

Help with iPython

[In 8]: numpy.sin?



Python Vs iPython

Help with iPython

[In [8]: numpy.sin?

Type: ufunc
String form: <ufunc 'sin'>
File: ~/anaconda/lib/python2.7/site-packages/numpy/__init__.py
Docstring:
sin(x[, out])

Trigonometric sine, element-wise.

Parameters

x : array_like

Angle, in radians (:math:`2 \pi` rad equals 360 degrees).

Returns

y : array_like

The sine of each element of x.



Python Vs iPython

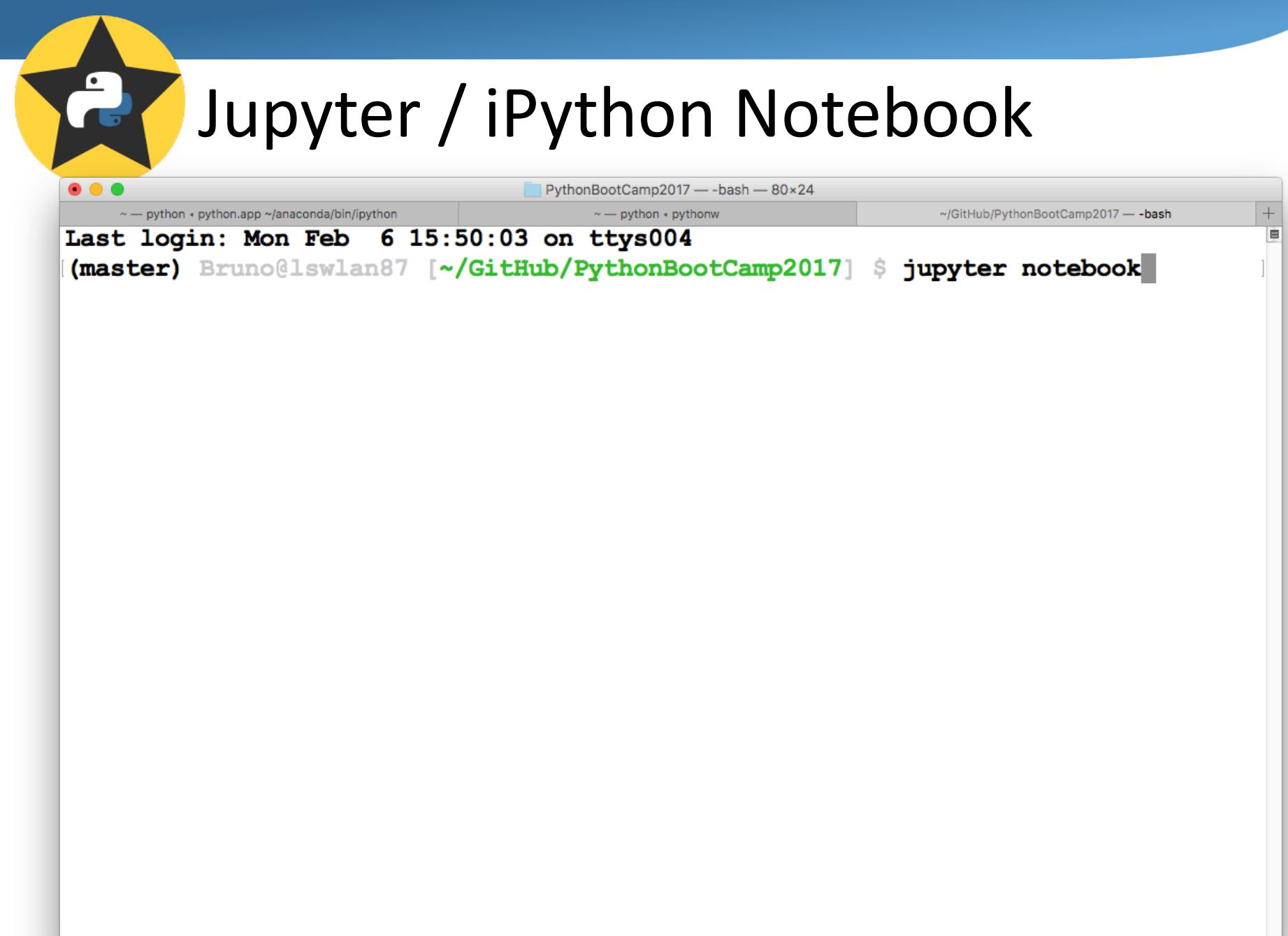
Magic Functions

```
[In 14]: %log
%logoff    %logon     %logstart   %logstate   %logstop
```

```
[In 14]: %logstart?
```

```
[In 16]: %%time
....: for i in range(100):
....:     x = 0
....:
```

```
CPU times: user 120 µs, sys: 48 µs, total: 168 µs
Wall time: 128 µs
```





Jupyter / iPython Notebook

```
PythonBootCamp2017 — jupyter-notebook — 77x21
~ — python · python.app ~/anaconda/bin/ipython           ~ — python · pythonw          ~/GitHub/PythonBootCamp2017 — jupyter-notebook + []
Last login: Mon Feb  6 15:50:03 on ttys004
[master] Bruno@lswlan87 [~/GitHub/PythonBootCamp2017] $ jupyter notebook
[W 16:00:21.442 NotebookApp] Unrecognized JSON config file version, assuming
version 1
[I 16:00:26.878 NotebookApp] [nb_conda_kernels] enabled, 3 kernels found
[I 16:00:27.093 NotebookApp] The port 8888 is already in use, trying another
port.
[I 16:00:27.400 NotebookApp] ✓ nbpresent HTML export ENABLED
[W 16:00:27.400 NotebookApp] ✗ nbpresent PDF export DISABLED: No module named
nbbrowserpdf.exporters.pdf
[I 16:00:27.419 NotebookApp] [nb_conda] enabled
[I 16:00:27.889 NotebookApp] [nb_anacondacloud] enabled
[I 16:00:27.988 NotebookApp] Serving notebooks from local directory: /Users/B
runo/GitHub/PythonBootCamp2017
[I 16:00:27.989 NotebookApp] 0 active kernels
[I 16:00:27.989 NotebookApp] The Jupyter Notebook is running at: http://local
host:8889/
[I 16:00:27.989 NotebookApp] Use Control-C to stop this server and shut down
all kernels (twice to skip confirmation).
```



Jupyter / iPython Notebook

A screenshot of a web browser window displaying the Jupyter Notebook interface. The title bar shows the URL `localhost:8889/tree#notebooks`. The browser's toolbar includes icons for search, refresh, and various extensions. The main content area is titled "jupyter". Below the title, there are tabs for "Files", "Running", "Clusters", and "Conda". A message says "Select items to perform actions on them." with buttons for "Upload", "New", and a refresh icon. The file list shows:

- [Notebooks](#)
- [Presentations](#)
- [README.md](#)



Jupyter / iPython Notebook

A screenshot of a Jupyter Notebook interface running in a web browser. The browser window has three tabs: "Notebooks/", "Untitled", and "Python BootCamp 2017 - Alex". The main content area shows the Jupyter logo and the title "Untitled (unsaved changes)". Below the title is a toolbar with various icons for file operations like saving, opening, and deleting, as well as cell controls like running, executing, and inserting. A large input cell is visible, labeled "In []:", with a cursor inside it. The browser's address bar shows the URL "localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]". The top right corner of the browser window shows the name "Bruno".



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the browser is on localhost:8889, showing tabs for 'Notebooks/' (active), 'Untitled', and 'Python BootCamp 2017 - Alex'. The address bar shows the URL. The main window displays a Jupyter notebook titled 'Untitled' with '(unsaved changes)' in parentheses. The toolbar includes standard file operations (New, Open, Save, Print) and a 'Code' dropdown. Below the toolbar is a code cell labeled 'In [1]' containing the following Python code:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Untitled" and the tab bar shows "localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]". The main area displays a cell labeled "In [1]" containing Python code:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```

Below this cell is another cell labeled "In []:" with a cursor, indicating it is ready for input. A red box highlights the following text in the bottom right corner:

<ENTER>: New line
<SHIFT> + <ENTER>: Execute cell



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the browser is on localhost:8889, showing three tabs: 'Notebooks/' (active), 'Untitled', and 'Python BootCamp 2017 - Alex'. The address bar shows the URL. The toolbar includes standard browser controls (back, forward, search) and various Jupyter-specific icons (file operations, cell controls, etc.). The main area displays two code cells:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```

```
In [ ]: print "Hello World"
```



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The browser window has tabs for 'Notebooks/' (active), 'Untitled', and 'Python BootCamp 2017 - Alex'. The address bar shows the URL `localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]`. The notebook itself has a title 'jupyter Untitled (unsaved changes)'. The toolbar includes buttons for file operations (New, Open, Save, etc.), cell selection (Cell, Kernel, Help), and a CellToolbar. Below the toolbar, two code cells are visible:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```

```
In [2]: print "Hello World"  
Hello World
```

A third cell, In [], is currently selected and ready for input.



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Untitled" and the tab bar shows "localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]". The browser's address bar also displays this URL. The main content area shows two code cells:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```

```
In [2]: print "Hello World"  
Hello World
```

A blue arrow points from the text "Command Mode" to the input field of the second code cell. The input field for the second cell is labeled "In []:".

Toolbar icons include: file, new, cut, copy, paste, up, down, left, right, cell, code, cell toolbar, cloud, gift, and video.

Kernel: Python [Root]

File Edit View Insert Cell Kernel Help

Python BootCamp 2017 - Alex Bruno

localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]

Apps Entertainment Work For reading Save to Mendeley ADS Search Lushprojects.com - SOAR - Work time f... Other Bookmarks

jupyter Untitled (unsaved changes)

In [1]: import numpy
x = numpy.arange(100) * 0.01
y = numpy.exp(-x)

In [2]: print "Hello World"

Hello World

Command Mode

In []:



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Untitled" and is part of a tab group for "Python BootCamp 2017 - Alex". The browser address bar shows the URL `localhost:8889/notebooks/Notebooks/Untitled.ipynb?kernel_name=Python%20[Root]`. The toolbar includes standard browser controls (back, forward, search) and links to various bookmarks and applications. Below the toolbar is the Jupyter logo and the notebook title "Untitled (unsaved changes)". The main workspace contains two code cells:

```
In [1]: import numpy  
x = numpy.arange(100) * 0.01  
y = numpy.exp(-x)
```

```
In [2]: print "Hello World"  
Hello World
```

A green arrow points from the "Edit Mode" label to the input field of the second code cell. A green box highlights the "Edit Mode" label. The bottom left corner of the workspace has a green border and contains the text "In []:".



Jupyter / iPython Notebook

A screenshot of a Jupyter Notebook interface running in a web browser. The browser window title is "Notebooks". The address bar shows "localhost:8889/tree/Notebooks". The user is logged in as "Bruno". The top menu bar includes "File", "Running", "Clusters", and "Conda". Below the menu, there is a message "Select items to perform actions on them." with "Upload", "New", and "Copy" buttons. The main content area shows a file tree under "/ Notebooks". The files listed are: "..", "fig", "Python BootCamp 2017 - A example of a notebook.ipynb", and "Python BootCamp 2017 - Display Data.ipynb".



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The browser window has tabs for 'Notebooks/' (active), 'Untitled', and 'Python BootCamp 2017 - A example of a notebook'. The address bar shows the URL `localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20A%20example%20of%20`. The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar below the menu bar contains icons for file operations like new, open, save, and delete, along with navigation and cell editing tools. The main workspace shows a single code cell with the placeholder text '`<m> in _command mode_: switch to Markdown edition`'. This text is enclosed in a red rectangular box.



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Python BootCamp 2017 - A example of a notebook" and is autosaved. The browser address bar shows the URL "localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20A%20example%20of...". The toolbar includes standard file operations like New, Open, Save, and Delete, along with a CellToolbar button. The main content area displays a sample notebook with the following text:

Sample Notebook

This Jupyter Notebook is intended to be an example with some references. Jupyter uses [\[Markdown\]\(https://en.support.wordpress.com/markdown-quick-reference/\)](https://en.support.wordpress.com/markdown-quick-reference/) Syntax and accept LaTeX and HTML codes.

With this, one can easily write **bold** or *italic* words. One can also type some `code inline` or code block like bellow:

```
```Python
import numpy
x = numpy.arange(10)
print x
```
```



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Python BootCamp 2017 - A example of a notebook" and is autosaved. The browser address bar shows the URL "localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20A%20example%20of...". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar below the menu bar contains icons for file operations like New, Open, Save, and Delete, as well as cell controls like Run, Stop, and CellToolbar. The main content area displays a section titled "Sample Notebook" with the following text:

This Jupyter Notebook is intended to be an example with some references. Jupyter uses [Markdown](#) Syntax and accept LaTeX and HTML codes.

With this, one can easily write **bold** or *italic* words. One can also type some code `inline` or code block like bellow:

```
import numpy
x = numpy.arange(10)
print x
```



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the page is "localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20example%20of..." and the tab says "Python BootCamp 2017 - A example of a notebook (autosaved)". The browser's address bar also shows the URL. The main content area displays a code cell with the following text:

```
One can write formulas using LaTeX syntax like  $\cos(x) = \frac{X}{Y}$  or create numerated lists like:  
1. Item 1  
2. Item 2  
3. Item 3  
  
Or unordered lists:  
• Item 1  
• Item 2  
• Item 3
```

At the bottom left, there is an "In []:" prompt followed by an empty code input field.



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "Python BootCamp 2017 - A example of a notebook". The toolbar includes standard file operations like Open, Save, and New, along with specific Jupyter tools like CellToolbar, Cell, and Kernel. The main content area displays a cell containing the following text:

```
One can write formulas using LaTeX syntax like
$ \cos \left( x \right) = \frac{X}{Y} $
or create numerated lists like:

1. Item 1
2. Item 2
3. Item 3

Or unordered lists:

* Item 1
* Item 2
* Item 3
```

Below the content area, there is a "In []:" prompt, indicating the cell is ready for input.



Jupyter / iPython Notebook

SAMFP - Extract Velocity × Bruno

Not Secure | https://soarbr3:2112/notebooks/SAM-FP/SAMFP%20-%20Extract%20Velocity%20Field.ipynb

Apps Entertainment Work For reading Save to Mendeley ADS Search Lushprojects.com - SOAR - Work time for Other bookmarks

jupyter SAMFP - Extract Velocity Field Last Checkpoint: 01/24/2017 (autosaved)

Logout

In [1]:

Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level actions and is indicated by a grey cell border with a blue left margin.

Command Mode (press `Esc` to enable)

`F`: find and replace
`Ctrl-Shift-P`: open the command palette
`Enter`: enter edit mode
`Shift-Enter`: run cell, select below
`Ctrl-Enter`: run selected cells

`Shift-J`: extend selected cells below
`A`: insert cell above
`B`: insert cell below
`X`: cut selected cells
`C`: copy selected cells

In [9]:

In [10]:

```
del cube
h = pyfits.getheader(filename)
```

`Close`



Jupyter / iPython Notebook

A screenshot of a Jupyter Notebook interface. The title bar shows the URL `localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20A%20example%20of...`. The main window title is "Python BootCamp 2017 - A example of a notebook". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar includes icons for file operations like Open, Save, and New, as well as a "Code" dropdown and a "CellToolbar" button. A code completion dropdown menu is open over a cell containing the code `In []: from numpy.`. The dropdown shows suggestions starting with "2. Ite", "3. Ite", and "Or und". The "Or und" section lists `numpy.complex_`, `numpy.complexfloating`, and three bullet-pointed items: `numpy.compress`, `numpy.concatenate`, and `numpy.conj`. Below these are `numpy.conjugate` and `numpy.convolve`.

```
In [ ]: from numpy.
```



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The browser window has tabs for 'Notebooks/' (active), 'Untitled', and 'Python BootCamp 2017 - A example of a notebook'. The main content area displays a notebook cell containing:

Or unordered lists:

- Item 1
- Item 2
- Item 3

<h1>This is a title in HTML</h1>

In []:

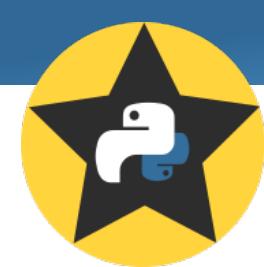
The browser's toolbar includes icons for file operations, navigation, and search. The address bar shows the URL `localhost:8889/notebooks/Notebooks/Python%20BootCamp%202017%20-%20A%20example%20of...`. The status bar at the bottom right shows the name 'Bruno'.



Jupyter / iPython Notebook

The screenshot shows a Jupyter Notebook interface running in a web browser. The browser window has tabs for 'Notebooks/' and 'Untitled'. The main content area displays the following:

- A title bar with the Jupyter logo and the text "Python BootCamp 2017 - A example of a notebook (autosaved)".
- A menu bar with options: File, Edit, View, Insert, Cell, Kernel, Help.
- A toolbar below the menu bar with various icons for file operations like saving, opening, and deleting, as well as cell execution and kernel selection.
- The text "Or unordered lists:" followed by a bulleted list:
 - Item 1
 - Item 2
 - Item 3
- A large bold title "This is a title in HTML".
- A code cell input field labeled "In []:".

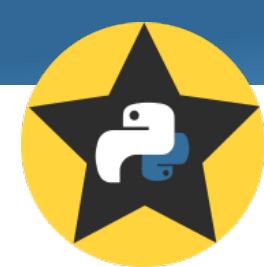


PyCharm – Not Used

The screenshot shows the PyCharm IDE interface. The title bar indicates the file is `main_file.py` in the `PyCharm_Examples` project. The code editor displays the following Python code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
```

The line `x = numpy.arange(10)` is highlighted with a yellow background, indicating it is currently not used in the code. The PyCharm interface includes a Project tool window on the left showing files `main_file.py` and `my_methods.py`, and a Structure tool window below it. The bottom status bar shows PEP 8 compliance information and the current time as 4:21.



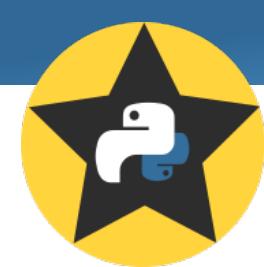
PyCharm – Tab

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard file operations (New, Open, Save, etc.) and search.
- Project View:** Shows the project structure under "PyCharm_Examples" with files main_file.py and my_methods.py.
- Code Editor:** Displays Python code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin
```

A code completion dropdown is open at the line "y = numpy.sin", listing various methods from the numpy module.
- Completion Dropdown:** Shows suggestions for "sin":
 - numpy.core.umath.sin(x, out)
 - numpy.core.umath.sin(x)
 - numpy.dtype.sin
 - numpy.core.umath.sinh(x, out)
 - numpy.core.umath.arcsin(x, out)
 - numpy.core.umath.arcsinh(x, ou...)
 - numpy.dtype.csingle
 - numpy.core.umath.isinf(x, out)
 - numpy.core.umath.isposinf(x, y)
- Bottom Status Bar:** Python Console, Terminal, TODO, and a message about an unregistered VCS root.
- Bottom Footer:** Unregistered VCS root detected: The directory /Users/Bruno/GitHub/PythonBootCamp...



PyCharm – See parameters

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard file operations (New, Open, Save, etc.) and search.
- Project View:** Shows the project structure under "PyCharm_Examples" with files main_file.py and my_methods.py.
- Code Editor:** Displays the following Python code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin()
```

A tooltip appears over the line "y = numpy.sin()", indicating the parameters: "x, out=None".
- Sidebar:** Includes sections for "1: Project", "2: Structure", and "2: Favorites".
- Bottom Navigation:** Python Console, Terminal, TODO, Event Log.
- Status Bar:** Shows "Parameter 'x' unfilled", the current time (5:15), and encoding (UTF-8).



PyCharm

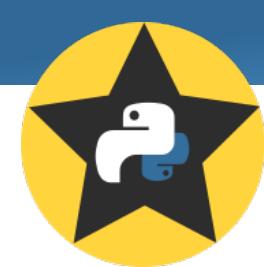
The screenshot shows the PyCharm IDE interface. The code editor displays two Python files: `main_file.py` and `my_methods.py`. The `my_methods.py` file contains the following code:

```
1
2     def say_hello():
3         """A very sad method calling you back"""
4         print "Hello... it's me..."
5
6     def Say_Hello_World():
7         print "Hello World!"
8
```

The `say_hello` function has a TODO item at line 4. The PyCharm TODO tool shows one item found in `main_file.py`: `(7, 3) # ToDo Fix PEP8`.

At the bottom, the status bar indicates an unregistered VCS root and the current time as 4:20.

```
my_methods.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
PyCharm_Examples > my_methods.py >
Project PyCharm_Examples main_file.py my_methods.py External Libraries
1: Project 2: Structure
1
2     def say_hello():
3         """A very sad method calling you back"""
4         print "Hello... it's me..."
5
6     def Say_Hello_World():
7         print "Hello World!"
8
TODO: Project Current File Scope Based
Found 1 TODO item in 1 file
PyCharm_Examples (1 item in 1 file)
  main_file.py
    (7, 3) # ToDo Fix PEP8
Python Console Terminal Run Debug TODO Event Log
Unregistered VCS root detected: The directory /Users/Bruno/GitHub/PythonBootCamp2017 is under Git, but is not register... (35 minutes ago) 4:20 LF UTF-8
```



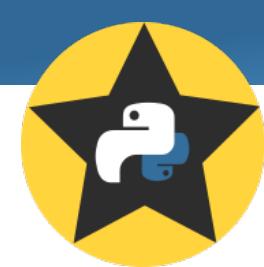
PyCharm – Auto-complete

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard file operations (New, Open, Save, etc.) and search.
- Project View:** Shows the project structure under "PyCharm_Examples" with files main_file.py and my_methods.py.
- Code Editor:** Displays the following Python code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 my_methods.
```

The word "my_methods" is highlighted in blue, indicating it is being typed or is a suggestion.
- Status Bar:** Shows "Statement seems to have no effect" and "7:12 LF UTF-8".
- Bottom Status Bar:** Shows "Event Log" with one entry.



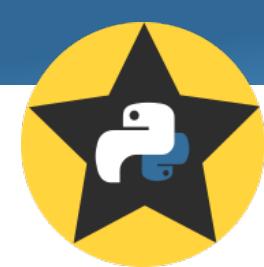
PyCharm – Auto-complete

The screenshot shows the PyCharm IDE interface with the following details:

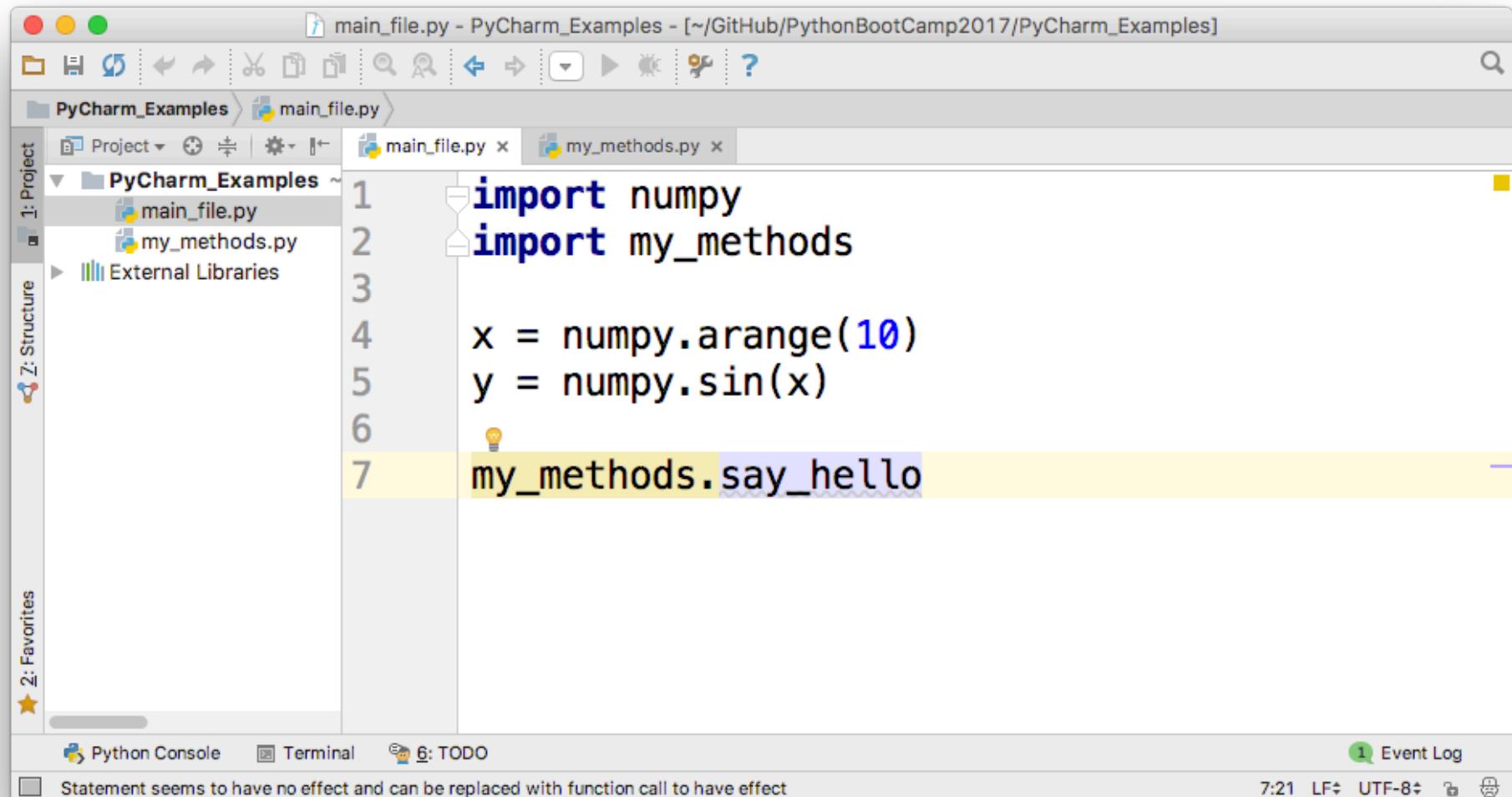
- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard file operations (New, Open, Save, etc.) and search.
- Project View:** Shows the project structure under "PyCharm_Examples" with files main_file.py and my_methods.py.
- Code Editor:** Displays the following code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 my_methods.
```

A dropdown auto-complete menu is open at the end of the line "7 my_methods.". It lists several suggestions:
 - say_hello() my_methods
 - Say_Hello_World() my_methods
 - if if expr
 - ifn if expr is None
 - ifnn if expr is not None
 - main if __name__ == '__main__'...
 - not not expr
 - par (expr)
 - print print(expr)
 - return return expr
 - while while expr
- Bottom Status Bar:** "Statement seems to have no effect".
- Bottom Right:** Event Log, UTF-8 encoding, and other status indicators.



PyCharm – Quick Help

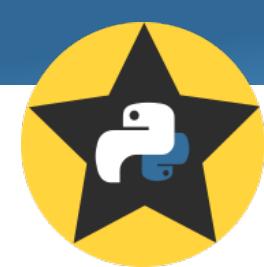


The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard file operations (New, Open, Save, etc.) and search.
- Project View:** Shows the project structure under "PyCharm_Examples" with files main_file.py and my_methods.py.
- Code Editor:** Displays Python code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 my_methods.say_hello
```

The line "7 my_methods.say_hello" is highlighted with a yellow background, and a lightbulb icon is positioned above the cursor, indicating a code completion suggestion.
- Bottom Status Bar:** Python Console, Terminal, TODO (6), Event Log (1), and file encoding (UTF-8).
- Message Bar:** Statement seems to have no effect and can be replaced with function call to have effect.



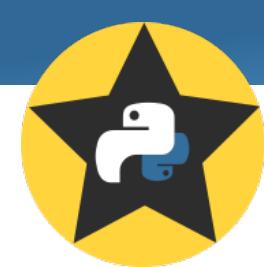
PyCharm – Quick Help

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** PyCharm_Examples
- Files:** main_file.py, my_methods.py
- Code:** main_file.py contains:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 my_methods.say_hello
```
- Completion Pop-up:** A tooltip for the method call "my_methods.say_hello" is displayed, titled "Documentation for say_hello()". It shows the definition:

```
def say_hello()
Inferred type: () -> None
```
- Status Bar:** A message states: "Statement seems to have no effect and can be replaced with func".
- Bottom Bar:** Includes links to Python Console, Terminal, and TODO, along with a log message: "A very sad method calling you back".



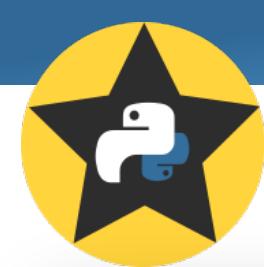
PyCharm - Run

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** PyCharm_Examples
- Files:** main_file.py, my_methods.py
- Code in main_file.py:**

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 my_methods.Say_Hello_World()
```
- Run Tab Output:**

```
Run main_file
Process started...
Process finished with exit code 0
Hello World!
```
- Status Bar:** Unregistered VCS root detected: The directory /User... (19 minutes ago) Connecting to console...
- Bottom Right:** Event Log icon, 1 event.



PyCharm - Debug

The screenshot shows the PyCharm IDE interface during a debugging session. The main window displays the code in `main_file.py`:

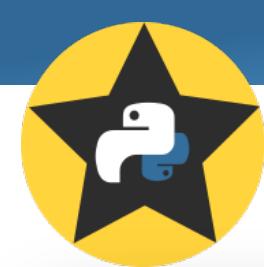
```
5      ^ - numpy.arange[0, 10, 1]
6          y = numpy.sin(x)    y: [ 0.
7  ● my_methods.Say_Hello_World()
```

The line `my_methods.Say_Hello_World()` is highlighted in blue and has a red circular breakpoint icon at the start. The variable `y` is shown as an array of values: `[0. 0.84147098 0.90929743 0.14112001 -0.7568025 -0.9589242 ...]`.

The bottom panel shows the **Debugger** tool window, which includes the **Frames** and **Variables** tabs. The **Variables** tab lists:

- Special Variables
- `x = {ndarray} [0 1 2 3 4 5 6 7 8 9] ...View as Array`
- `y = {ndarray} [0. 0.84147098 0.90929743 0.14112001 -0.7568025 -0.9589242 ...View as Array`

At the bottom, the status bar indicates: "Unregistered VCS root detected: The directory /Users/Bruno/GitHub/PythonBootCamp2017 is under Git, but is not registered ... (19 minutes ago)".



PyCharm – PEP-8

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** main_file.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard PyCharm icons for file operations.
- Project View (1: Project):** Shows the PyCharm_Examples project with files main_file.py and my_methods.py.
- Structure View (2: Structure):** Shows the code structure with imports and function definitions.
- Code Editor:** The main_file.py file contains the following code:

```
1 import numpy
2 import my_methods
3
4 x = numpy.arange(10)
5 y = numpy.sin(x)
6
7 # ToDo Fix PEP8
8 my_methods.Say_Hello_World()
```

A yellow TODO marker is placed before the comment "# ToDo Fix PEP8".
- TODO Tool Window:** Shows "Found 1 TODO item in 1 file" under the main_file.py file.
- Bottom Status Bar:** Unregistered VCS root detected: The directory /Users/Bruno/GitHub/PythonBootCamp2017 is under Git, but is not registered... (20 minutes ago)
- Bottom Navigation:** Python Console, Terminal, Run, Debug, TODO (selected), Event Log.



PyCharm – PEP-8

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** my_methods.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard PyCharm icons for file operations.
- Project View (1: Project):** Shows the project structure with PyCharm_Examples containing main_file.py and my_methods.py.
- Code Editor (2: Structure):** Displays the Python code:

```
1
2     def say_hello():
3         """A very sad method calling you back"""
4         print "Hello... it's me..."
5
6     def Say_Hello_World():
7         print "Hello World!"
```

The line `print "Hello... it's me..."` is highlighted with a yellow background and a lightbulb icon, indicating a PEP-8 violation.
- TODO View (3: TODO):** Shows a single TODO item: `(7, 3) # ToDo Fix PEP8`.
- Bottom Status Bar:** Unregistered VCS root detected: The directory /Users/Bruno/GitHub/PythonBootCamp2017 is under Git, but is not registered... (35 minutes ago)



PyCharm - Structure

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** Shows the file "bouncing_ball.py" and the project path "[~/GitHub/PBC2017_IAG_Material/PyCharm_Examples]".
- Toolbar:** Standard PyCharm icons for file operations, navigation, and tools.
- Project View (Left):** Shows the project structure under "PyCharm_Examples". The "bouncing_ball.py" file is selected. Other files listed are "main_file.py" and "my_methods.py".
- Code Editor (Center):** Displays the Python code for a bouncing ball simulation. The "main()" function is highlighted.

```
8
9 import matplotlib.pyplot as plt
10 import matplotlib.animation as pani
11 import numpy as np
12
13
14 def main():
15
16     my_particle_1 = Particle(0.1, 0.001)
17     my_particle_2 = Particle(0.2, 0.0025)
18
19     my_box = Box()
20     my_box.add_particle(my_particle_1)
21     my_box.add_particle(my_particle_2)
```

- Toolbars and Panels (Bottom):** Includes "TODO", "Version Control", "Python Console", "Terminal", "Event Log", and status information like "14:12", "LF", "UTF-8", "Git: bquint-updates", and battery level.



PyCharm - Structure

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** Shows the file name `bouncing_ball.py` and the project path `~/GitHub/PBC2017_IAG_Material/PyCharm_Examples`.
- Toolbar:** Standard PyCharm icons for file operations, navigation, and tools.
- Project View (Left):** Shows the project structure with `bouncing_ball.py` as the main file. It lists methods like `main()`, `update(frame_number)`, and class definitions for `Box` and `Particle`. The `main()` method is currently selected.
- Code Editor (Right):** Displays the code for `bouncing_ball.py`. The `main()` function is highlighted in yellow. The code imports `matplotlib.pyplot`, `matplotlib.animation`, and `numpy`, and defines two particles and a box.
- Bottom Navigation:** Icons for TODO, Version Control, Python Console, Terminal, and Event Log.
- Status Bar:** Shows the current time (14:12), file encoding (UTF-8), Git status (Git: bquint-updates), and other system information.

```
8     .....
9
10    import matplotlib.pyplot as plt
11    import matplotlib.animation as pani
12    import numpy as np
13
14    def main():
15
16        my_particle_1 = Particle(0.1, 0.001)
17        my_particle_2 = Particle(0.2, 0.002)
18
19        my_box = Box()
20        my_box.add_particle(my_particle_1)
21        my_box.add_particle(my_particle_2)
22
```



PyCharm – Version Control (Git)

The screenshot shows the PyCharm IDE interface. The top bar displays the file 'my_methods.py' under the project 'PyCharm_Examples'. The left sidebar shows the project structure with files 'main_file.py' and 'my_methods.py' in the 'PyCharm_Examples' folder. The main editor window contains two Python functions:

```
1
2 def say_hello():
3     """A very sad method calling you back"""
4     print "Hello... it's me..."
5
6 def Say_Hello_World():
7     print "Hello World!"
```

The code editor highlights the second function definition with a red squiggly underline, indicating a PEP8 violation. The bottom-left panel, titled 'Favorites', shows a 'TODO' section with one item found in 'main_file.py': '(7, 3) # ToDo Fix PEP8'. The bottom navigation bar includes tabs for 'Version Control', 'Python Console', 'Terminal', 'Run', 'Debug', 'TODO', and 'Event Log'. A status bar at the bottom indicates 'Added Git root: /Users/Bruno/GitHub/PythonBootCamp2017 (3 minutes ago)'.



PyCharm – Version Control (Git)

The screenshot shows the PyCharm IDE interface. The top bar displays the file 'my_methods.py' under the project 'PyCharm_Examples'. The left sidebar shows the project structure with files 'main_file.py' and 'my_methods.py' in the 'PyCharm_Examples' folder. The main editor window contains the following Python code:

```
1
2     def say_hello():
3         """A very sad method calling you back"""
4         print "Hello... it's me..."
5
6     def Say_Hello_World():
7         print "Hello World!"
```

The code editor highlights the string in line 3 with a light yellow background. The bottom navigation bar includes tabs for Version Control, Python Console, Terminal, Run, Debug, and TODO. The TODO tab is selected, showing a list of items: 'Found 1 TODO item in 1 file' under 'PyCharm_Examples'. A specific entry is expanded to show '(7, 3) # ToDo Fix PEP8'. The status bar at the bottom indicates 'Added Git root: /Users/Bruno/GitHub/PythonBootCamp2017 (3 minutes ago)' and other system details.



PyCharm – Version Control (Git)

The screenshot shows the PyCharm IDE interface. The code editor displays two Python files: `main_file.py` and `my_methods.py`. The `my_methods.py` file contains the following code:

```
1
2 def say_hello():
3     """A very sad method calling you back"""
4     print "Hello... it's me..."
5
6 def Say_Hello_World():
7     print "Hello World!"
8
```

The line `def Say_Hello_World():` is highlighted with a yellow background and has a lightbulb icon above it, indicating a potential fix or suggestion. The PyCharm navigation bar shows the current file is `my_methods.py`.

In the bottom left corner, a green message box indicates: **2 files committed: Added PyCharm_Examples files.**

The bottom status bar shows the following information: **PEP 8: expected 2 blank lines, found 1. Function name should be lowercase.**, **1:30 LF UTF-8 Git: master**, and icons for Version Control, Python Console, Terminal, Run, Debug, and TODO.



PyCharm – Version Control (Git)

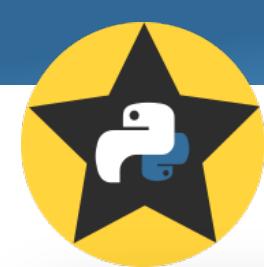
The screenshot shows the PyCharm IDE interface. The code editor displays two Python files: `main_file.py` and `my_methods.py`. The `my_methods.py` file contains the following code:

```
1 def say_hello():
2     """A very sad method calling you back"""
3     # Just a new line
4     print "Hello... it's me..."
5
6 def say_hello_world():
7     print "Hello World!"
```

The run terminal at the bottom shows the output of running `main_file.py`:

```
Run main_file
Process started...
>Hello World!
Process finished with exit code 0
```

The status bar at the bottom indicates "2 files committed: Added PyCharm_Examples files. (a minute ago)".



PyCharm – Version Control (Git)

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** my_methods.py - PyCharm_Examples - [~/GitHub/PythonBootCamp2017/PyCharm_Examples]
- Toolbar:** Standard PyCharm icons for file operations, search, and navigation.
- Project View:** Shows the project structure with PyCharm_Examples and my_methods.py selected.
- Code Editor:** Displays two versions of the file my_methods.py for comparison. The left column shows the history:
 - A minute ago: my_methods.py (Commit Changes: Added PyCharm_Examples files.)
 - 57 minutes ago: my_methods.py
 - Today 14:37: my_methods.py
 - Today 14:29: Create Python script my_methods
- Diff View:** The main area shows the differences between the current version and the commit from 57 minutes ago. The changes are:
 - Added a new function: `def say_hello():`
 - Changed the docstring of `say_hello()` from `"""A very sad metho` to `"""A very sad method`
 - Changed the print statement in `say_hello()` from `print "Hello... it'` to `print "Hello... it"`
 - Added a new function: `def Say_Hello_World():`
 - Changed the print statement in `Say_Hello_World()` from `print "Hello World` to `print "Hello World!"`
- Bottom Navigation:** Includes tabs for Version Control, Python Console, Terminal, Run, Debug, TODO, and Event Log. The Version Control tab is active.
- Status Bar:** Shows "2 files committed: Added PyCharm_Examples files. (a minute ago)" and system information like time and encoding.



PyCharm – Version Control (Git)

The screenshot shows the PyCharm IDE interface. The code editor displays a Python script named `my_methods.py` with the following content:

```
def say_hello():
    """A very sad method calling you back"""
    # Just a new line
    print "Hello... it's me..."
```

The project browser on the left shows a project named `PyCharm_Examples` containing files `main_file.py` and `my_methods.py`. The bottom panel shows the Git log for the `master` branch:

| Commit | Branch | Author | Date |
|----------------------------------|---------------|-------------|---------------|
| Added PyCharm_Examples files. | master | Bruno Quint | 2/8/17, 15:49 |
| Added .gitignore file | master | Bruno Quint | 2/8/17, 15:47 |
| Added new notebook with examples | origin/master | Bruno Quint | 2/7/17, 17:33 |
| Cleaned undesired files | master | Bruno Quint | 2/6/17, 09:24 |
| Finished presentation. | master | Bruno Quint | 2/6/17, 00:27 |
| Updated Data Display on MacBook | master | Bruno Quint | 2/3/17, 14:39 |
| Added a folder | master | Bruno Quint | 2/2/17, 11:58 |

On the right, a "Commit details" pane is visible. The bottom status bar indicates "2 files committed: Added PyCharm_Examples files. (2 minutes ago)".



Questions?

