pycon13_ipython

Unknown Author

September 04, 2013

1 PyConZa 2013

1.1 Cape Town

placeholder fro pycon talk if approved

1.2 The IPython notebook and the Python science stack

1.3 What is this

This repo contains the full

The Complete Talk GitHub Website can be accessed here

1.4 Description

IPython had become a popular choice for doing interactive scientific work. In addition to this IPython offers a web based Notebook that makes interactive work much easier. Notebooks have been used to write repeatable science papers and more recently a book has been published using this platform, the online Notebook Viewer and GitHub.

Combining the most common science packages with IPython makes it a formidable tool and serious competition to R.

As a matter of fact you can run R in the notebook session, embed YouTube Videos, Images and lots more.

The science stack consists of (but not limited to):

1.5 Talk contents

The talk will aim to introduce these tools and give some practical examples. Once completed it will be shown how easy it is to publish your work to :

num	item	description
1	ipython	quick intro to ipython and the notebook
1	setup	set up your environment / get the talk files
1	notebook basics	navigate the notebook
1	notebook magics	special notebook commands that can be very usefull
1	getting input	as from IPython 1.00 getting input from sdtin is possible
1	local files	how to link to local files in the notebook directory
1	plotting	how to create beautifull inline plots
1	symbolic math	quick demo of sympy model
1	pandas	quick intro to pandas dataframe
1	typsetting	include markdown, Latex via MathJax
1	loading code	how to load a remote .py code file
1	gist	paste some of your work to gist for sharing
1	js	some javascript examples
1	customising	loading a customer css and custom matplotlib config file
1	git cell	add code to a special cell that would commit to git
1	output formats	how to publish your work to html, pdf or jeveal.js presentation

In [2]:

```
#Some standard stuff. Also see last cell for custom css
%pylab inline
import json
s = json.load( open("static/matplotlibrc.json") )
matplotlib.rcParams.update(s)
figsize(16, 4)
```

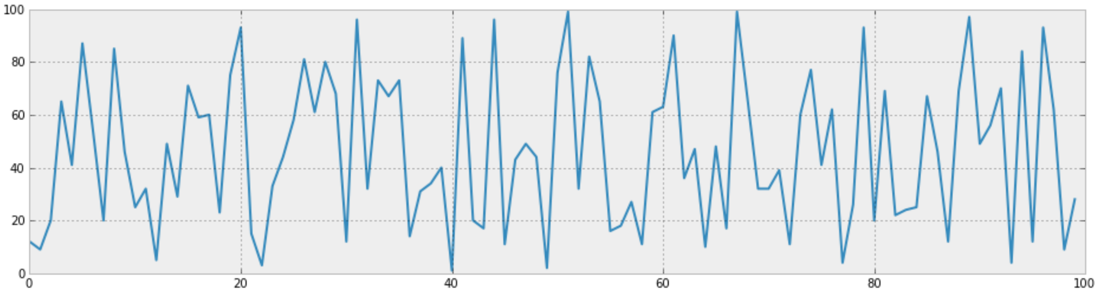
```
Populating the interactive namespace from numpy and matplotlib
```

In [3]:

```
x = randint(1, 100, 100)
plot(x)
```

```
[<matplotlib.lines.Line2D at 0x485c510>]
```

Out [3]:



In [4]:

```
from IPython.core.display import HTML
def css_styling():
    styles = open("static/custom.css", "r").read()
    return HTML(styles)
css_styling()
```

```
<IPython.core.display.HTML at 0x47d9c10>
```

```
Out [4]:
```

2 The End. . . .

Thank you.

```
In []:
```

```
In []:
```