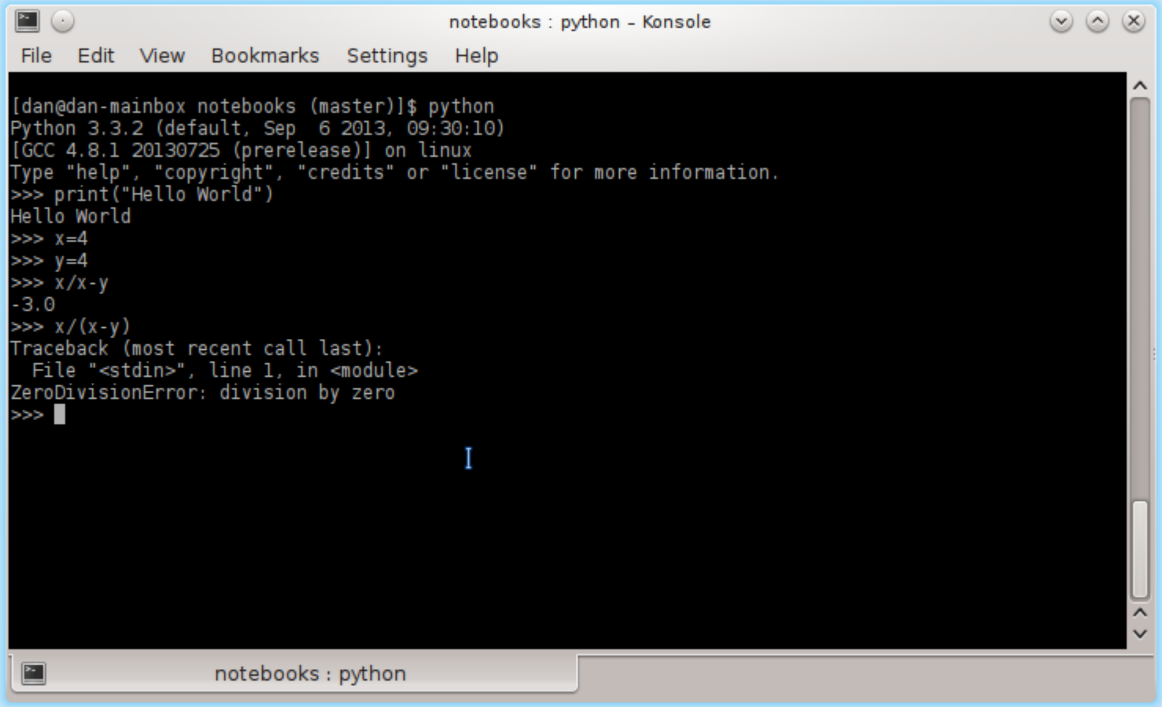
IP[y]: IPython

Interactive Computing

A quick introduction by Dan Liew

Have you ever tried using Python's interactive terminal?

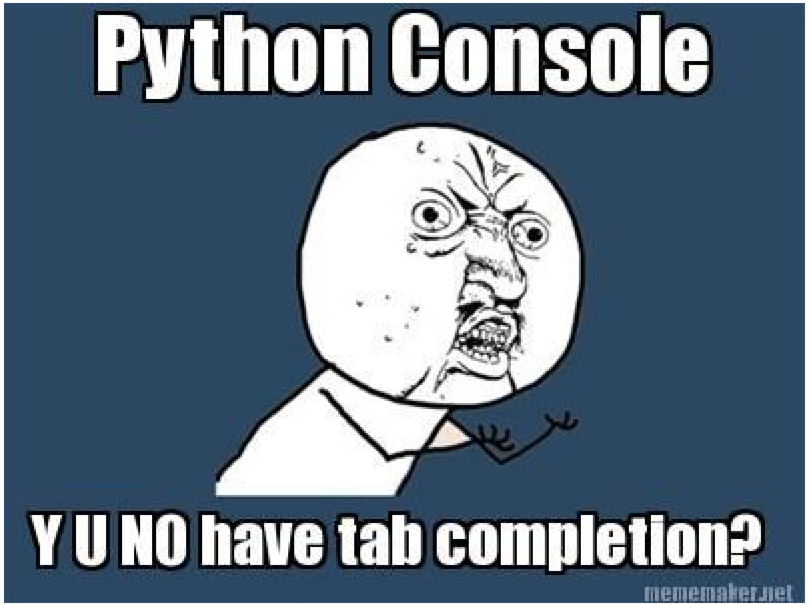


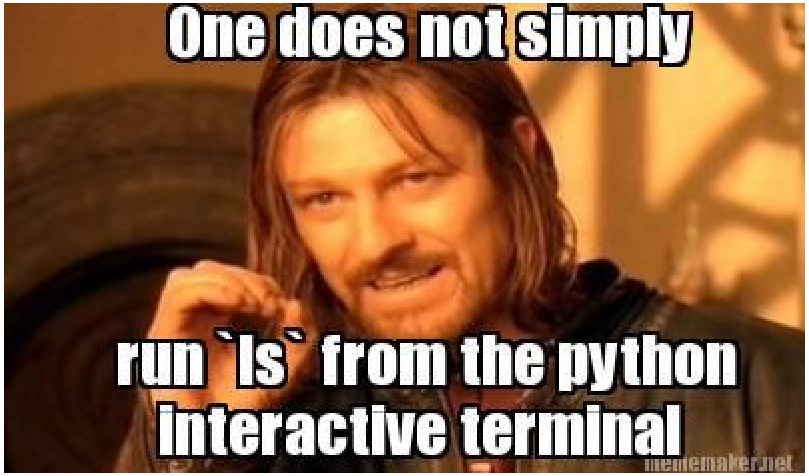
notebooks : python – Konsole
File Edit View Bookmarks Settings Help
notebooks : python

```
[dan@dan-mainbox notebooks (master)]$ python
Python 3.3.2 (default, Sep  6 2013, 09:30:10)
[GCC 4.8.1 20130725 (prerelease)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>> x=4
>>> y=4
>>> x/x-y
-3.0
>>> x/(x-y)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>> 
```

It's lame!

Problems include...





Introduction IPython

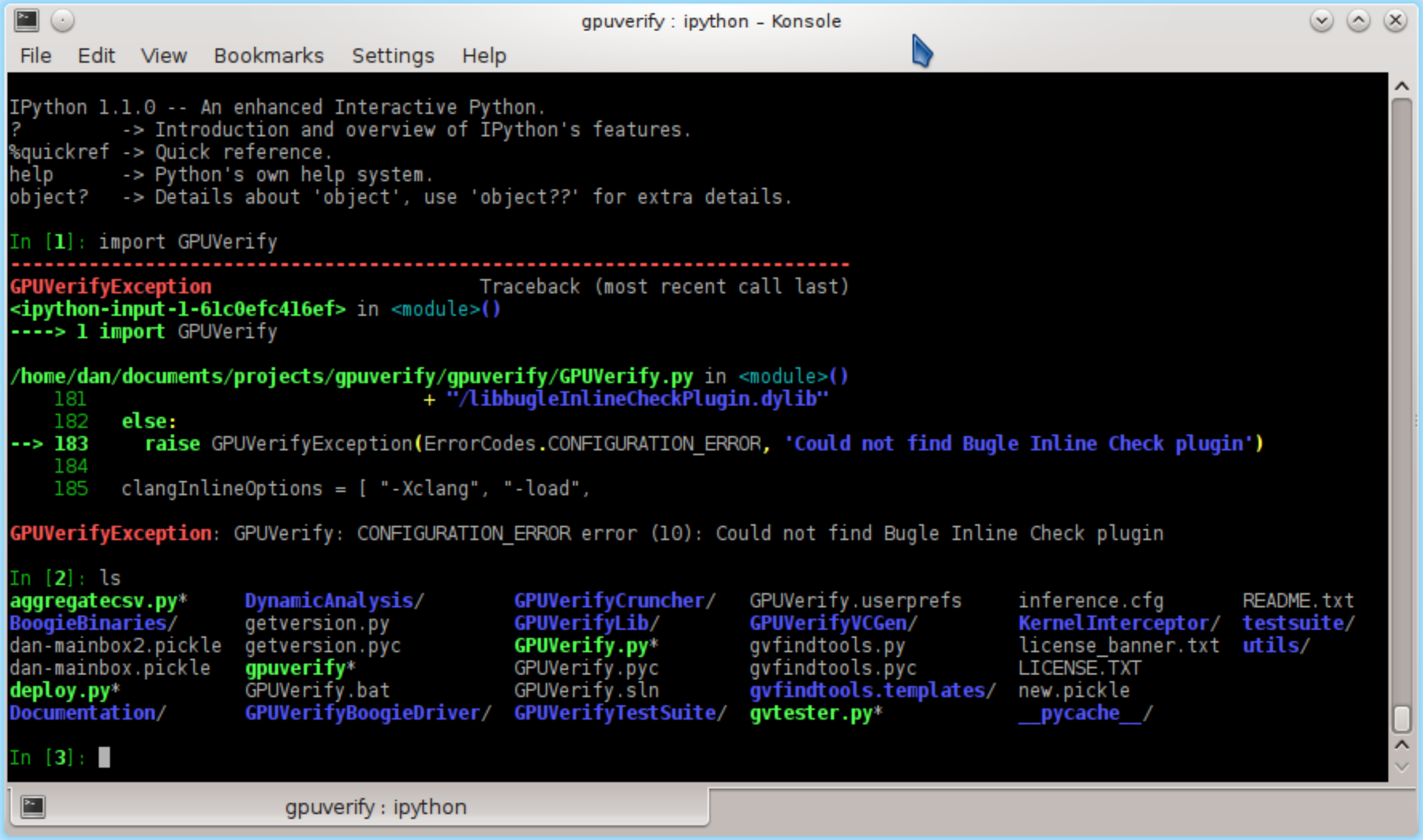
It started off as an "Afternoon hack" by Fernando Pérez

The IPython terminal console offers many features including...

- Tab completion
- Readable backtraces
- Nice shell integration
- Easy access to help

Quick Demo

Demo



```
IPython 1.1.0 -- 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]: import GPUVerify
---------------------------------------------------------------------------
GPUVerifyException                        Traceback (most recent call last)
<ipython-input-1-61c0efc416ef> in <module>()
----> 1 import GPUVerify

/home/dan/documents/projects/gpuverify/gpuverify/GPUVerify.py in <module>()
    181                               + "/libbugleInlineCheckPlugin.dylib"
    182   else:
--> 183     raise GPUVerifyException(ErrorCodes.CONFIGURATION_ERROR, 'Could not find Bugle Inline Check plugin')
    184
    185   clangInlineOptions = [ "-Xclang", "-load",

GPUVerifyException: GPUVerify: CONFIGURATION_ERROR error (10): Could not find Bugle Inline Check plugin

In [2]: ls
aggregatecsv.py*     DynamicAnalysis/        GPUVerifyCruncher/   GPUVerify.userprefs      inference.cfg        README.txt
BoogieBinaries/      getversion.py           GPUVerifyLib/        GPUVerifyVCGen/          KernelInterceptor/   testsuite/
dan-mainbox2.pickle  getversion.pyc          GPUVerify.py*        gvfindtools.py           license_banner.txt   utils/
dan-mainbox.pickle   gpuverify*              GPUVerify.pyc        gvfindtools.pyc          LICENSE.TXT
deploy.py*           GPUVerify.bat           GPUVerify.sln        gvfindtools.templates/   new.pickle
Documentation/       GPUVerifyBoogieDriver/  GPUVerifyTestSuite/  gvtester.py*             __pycache__/

In [3]: 
```

Is that it?

No, otherwise my presentation would be boring!

IPython is much much more.

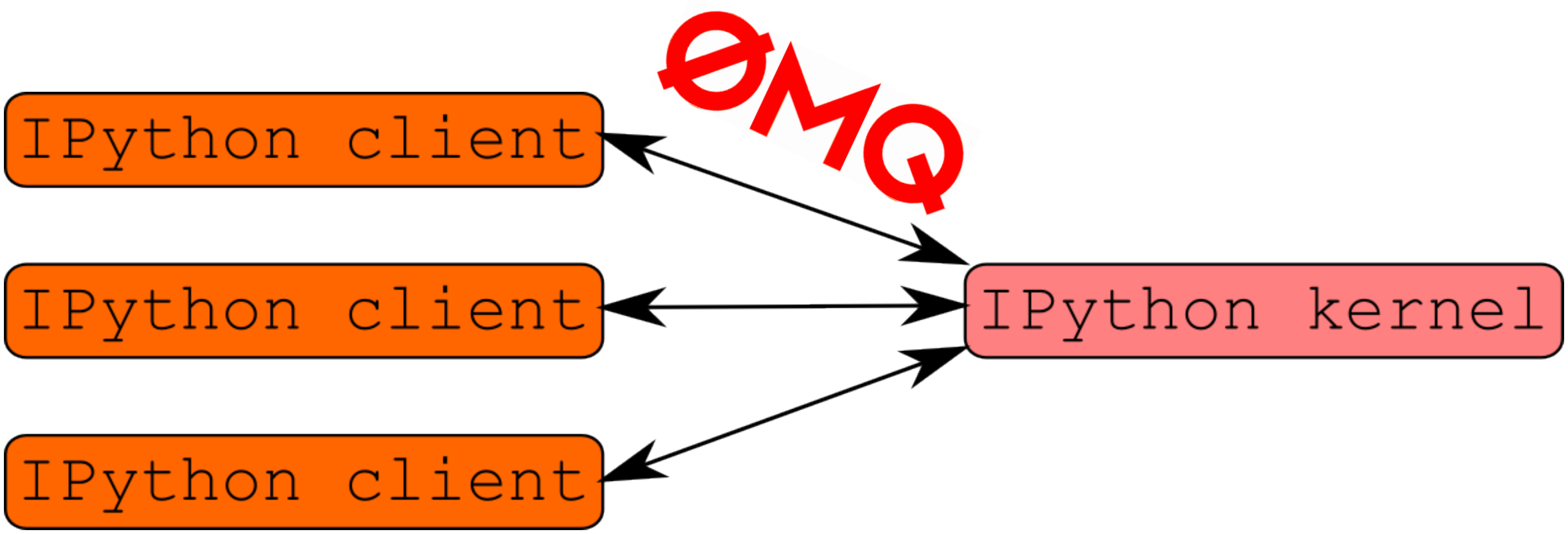
- Powerful interactive shells (terminal and Qt-based).
- A browser-based notebook with support for code, text, mathematical expressions, inline plots and other rich media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Easy to use, high performance tools for parallel computing **(I'm not going to cover this)**

Developed by many developers including the main developers:

Fernando Pérez, Min Ragan-Kelley, Brian E. Granger, Thomas Kluyver

IPython architecture

The interpreter loop abstracted...



- IPython client - Sends commands and receives their output. E.g. qtconsole and web notebook
- IPython kernel - A separate process running a python interpreter that receives commands from clients and executes commands. Multiple clients can connect!
- Communication between the kernel and clients is done via the ZeroMQ library. It is "a socket library that acts as a concurrency framework".

IPython architecture

In Python objects can provide an implementation of `__repr__()` which will return the string representation of an object.

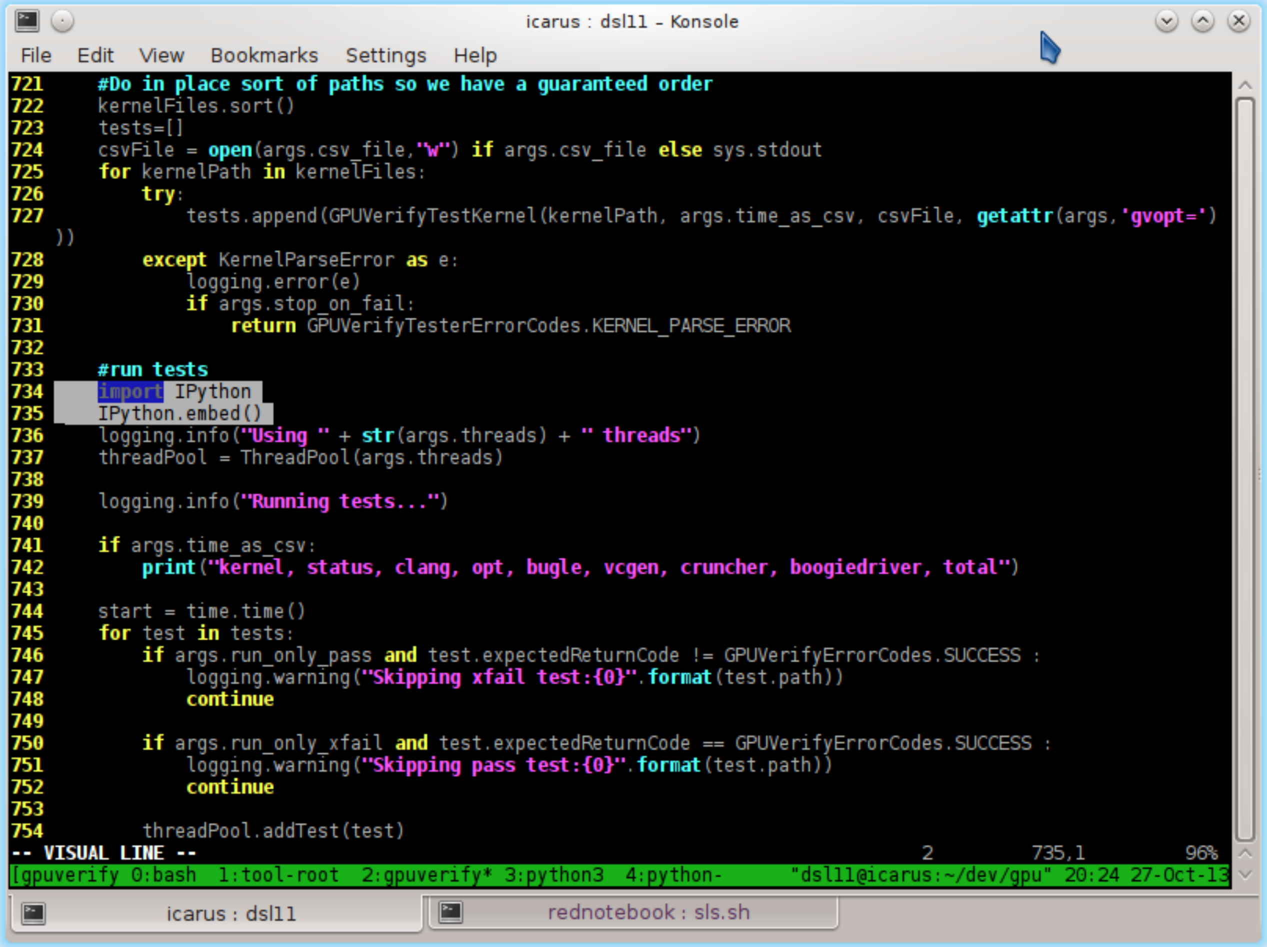
IPython takes this concept further and allows objects to provide other representations which can then be used appropriately. This is used to render things like graphs and LaTeX

`__repr_html() , __repr__svg() , repr__latex() ...`

IPython Notebook demo

Embedding an IPython interpreter in GPUVerify's gvtester.py

Add the following...



```
    #Do in place sort of paths so we have a guaranteed order
    kernelFiles.sort()
    tests=[]
    csvFile = open(args.csv_file,"w") if args.csv_file else sys.stdout
    for kernelPath in kernelFiles:
        try:
            tests.append(GPUVerifyTestKernel(kernelPath, args.time_as_csv, csvFile, getattr(args,'gvopt=')))
        except KernelParseError as e:
            logging.error(e)
            if args.stop_on_fail:
                return GPUVerifyTesterErrorCodes.KERNEL_PARSE_ERROR

    #run tests
    import IPython
    IPython.embed()
    logging.info("Using " + str(args.threads) + " threads")
    threadPool = ThreadPool(args.threads)

    logging.info("Running tests...")

    if args.time_as_csv:
        print("kernel, status, clang, opt, bugle, vcgen, cruncher, boogiedriver, total")

    start = time.time()
    for test in tests:
        if args.run_only_pass and test.expectedReturnCode != GPUVerifyErrorCodes.SUCCESS :
            logging.warning("Skipping xfail test:{0}".format(test.path))
            continue

        if args.run_only_xfail and test.expectedReturnCode == GPUVerifyErrorCodes.SUCCESS :
            logging.warning("Skipping pass test:{0}".format(test.path))
            continue

        threadPool.addTest(test)
-- VISUAL LINE --                                              2          735,1          96%
[gpuverify 0:bash  1:tool-root  2:gpuverify* 3:python3  4:python-     "dsl11@icarus:~/dev/gpu" 20:24 27-Oct-13
```

What do I think IPython is useful for ?

- Hacking on ideas
- Data visualisation
- Collaboration
- Teaching/Learning
- A darn good replacement for native Python console!

Some fantastic example notebooks

- https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks

- http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter1_Introduction/Chapter1_Introduction.ipynb

- http://nbviewer.ipython.org/4042018

For more information

Visit http://ipython.org/