

SciLifeLab



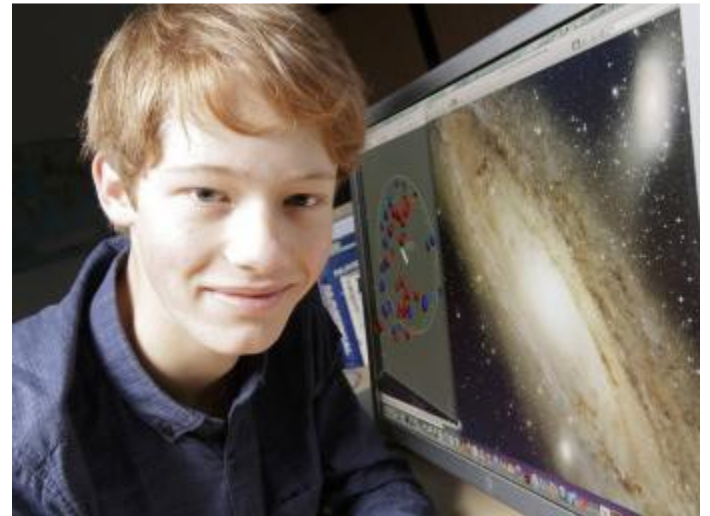
python™

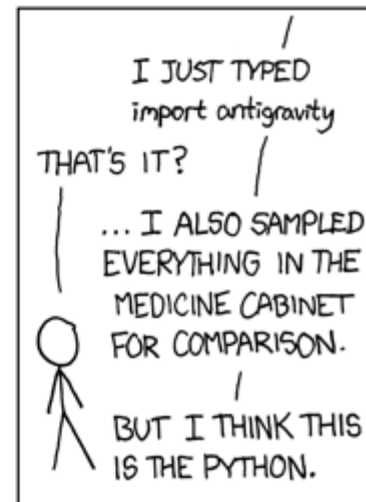
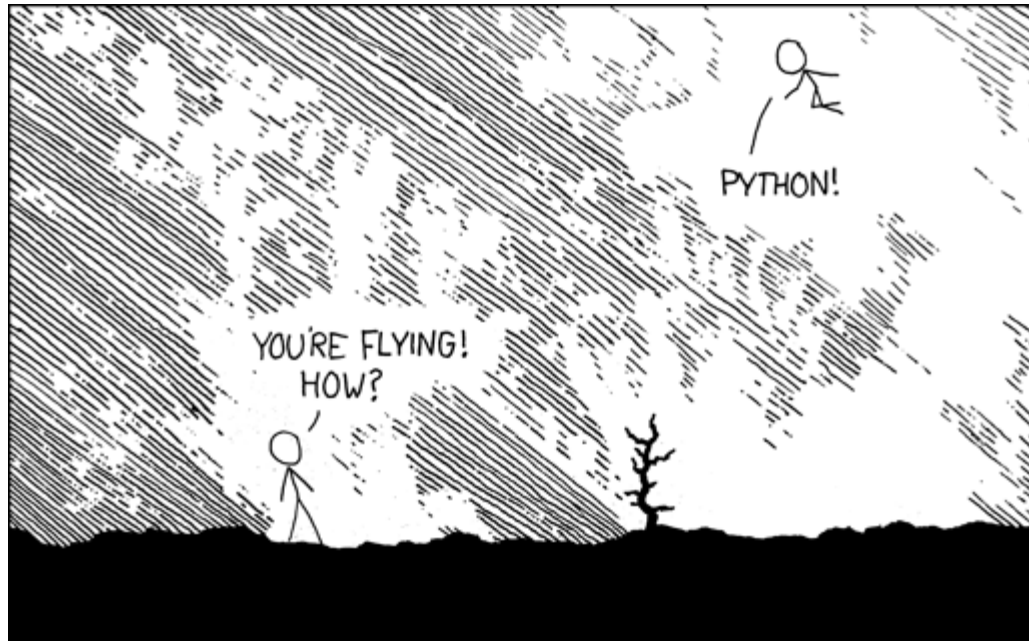
Welcome!

- Feedback/support as much as possible in the course google group.
- scientific-programming-in-python@scilifelab.se
 - Helps on learning new things together with all assistants.
 - Helps students support help each other.
 - Don't be shy! ;)
- Use your editor of choice, we have recommendations, ask TA's

"French teenager's research published in Nature"

"(...) Neil Iбата said he completed work experience with his father's team to learn about the computer programming language **Python**."





<http://xkcd.com/353/>

Organization

- Setting your environment
 - Important to get it right today, base for sessions
 - VirtualEnv
 - Test PyPi repository
 - Git
- for slides in this_course:
 - In-session exercises
 - At home exercises
- Some limited time for each in-session exercise
 - If not finished: TA's or @home

This does not happen here...

Anti-pattern

- I create my "sciency" scripts.
- I'm in a hurry, structure is secondary.
- I'll share and release later:
 - When my scripts are "good enough" for release.
 - When I have the time to learn version control.
 - I fear my NDA: Non Disclosure Agreement.
- I can document later on.
- When reviewers ask me for code, I send a .zip over gmail.
 - I mail the wrong version. Have to send it again. I get retracted for non-reproducible research.

Hopefully, after this course...

Good practice

- I learn about structure and packaging.
- I learn version control.
- I release my software in beta.
 - I do my science, release, science, release...
 - ... I get unexpected collaborations, improving my software as I go.
 - ... if someone asks if I fear being scooped, I point out that most bioinformatics software was released [way before](#) the manuscript.
- I have a simple URL to my code in my paper.
 - They can fetch many different versions.

Outline

1. Setup a basic python environment and project structure.
2. Push its code contents and keep track of them.
3. Push it for public distribution.

Virtual python environments

```
$ curl -s http://x90.es/pythonkurs | $SHELL
```

Original:

(<https://raw.githubusercontent.com/brainsik/virtualenv-burrito/master/virtualenv-burrito.sh> | \$SHELL)

Create your own env

```
$ workon
```

```
$ mkvirtualenv py2.7
```

```
$ workon py2.7
```

```
(py2.7)$ pip search bio
```

```
(py2.7)$ pip install biopython
```

Homework @home: Do it @UPPMAX

<http://x90.es/venvupppmax>

Git & Github

<http://try.github.com>

Those who have not gone through it,
please, do!



...unless you know what you are doing ;)

https://github.com/scilifelab



**Science For Life
Lab**

SciLifeLab

🕒 Joined on Apr 06, 2011

15 **0** **35**
public repos private repos members

 **Repositories**

 **Members**

 **Edit SciLifeLab's Profile**

Find a Repository...

All [Public](#) [Private](#) [Sources](#) [Forks](#) [Mirrors](#)



scilifelab

forked from [percyfal/scilifelab](#)

Misc code snippets for Science For Life Laboratory

Last updated 6 hours ago

Python ★ 4 🍴 10



status

forked from [vals/status](#)

Last updated 22 days ago

JavaScript ★ 1 🍴 1



bcbb

forked from [chapmanb/bcbb](#)

Useful bioinformatics code, primarily in Python and R

Last updated a month ago

Python ★ 13 🍴 50



logbook

forked from [mitsuhiko/logbook](#)

A cool logging replacement for Python. Under construction

Last updated a month ago

Python ★ 0 🍴 34



bcbio-nextgen-deploy

forked from [guillermo-carrasco/bcbio-nextgen-deploy](#)

Suite of scripts to automatically deploy nextgen pipeline

Last updated a month ago

Python ★ 0 🍴 1



portal

PHP ★ 1 🍴 0

SSH intermission

SSH keys

We check if we have our keypair already

```
$ ls -alh ~/.ssh
```

```
romanmac:pkg roman$ ls -alh ~/.ssh
```

```
-rw-r--r--  1 roman  staff  1,0K 25 Maj  2012 config  
-rw-----  1 roman  staff  1,6K  2 Sep  2011 id_rsa  
-rw-r--r--  1 roman  staff  417B  2 Sep  2011 id_rsa.pub
```

If not, we create it with:

```
$ ssh-keygen
```

... And then paste the **id_rsa.pub** into GitHub

Does everyone get this response?

```
$ ssh git@github.com
```

Hi brainstorm! You've successfully authenticated, but GitHub does not provide shell access.

Connection to github.com closed.

Create a new repository

Type in your surname

Your program skeleton

biostuff/

LICENSE.txt

README.txt

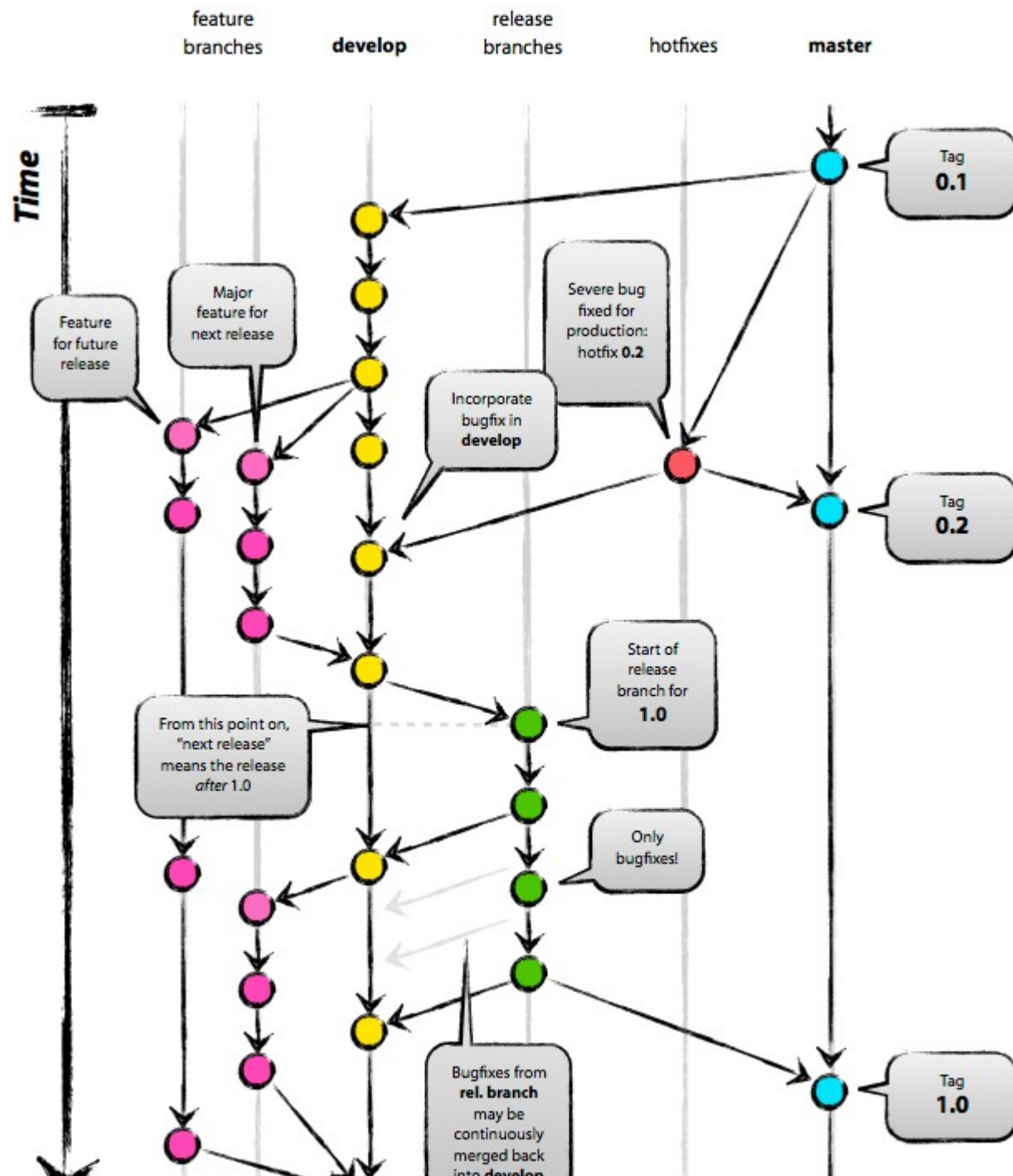
setup.py

biostuff/

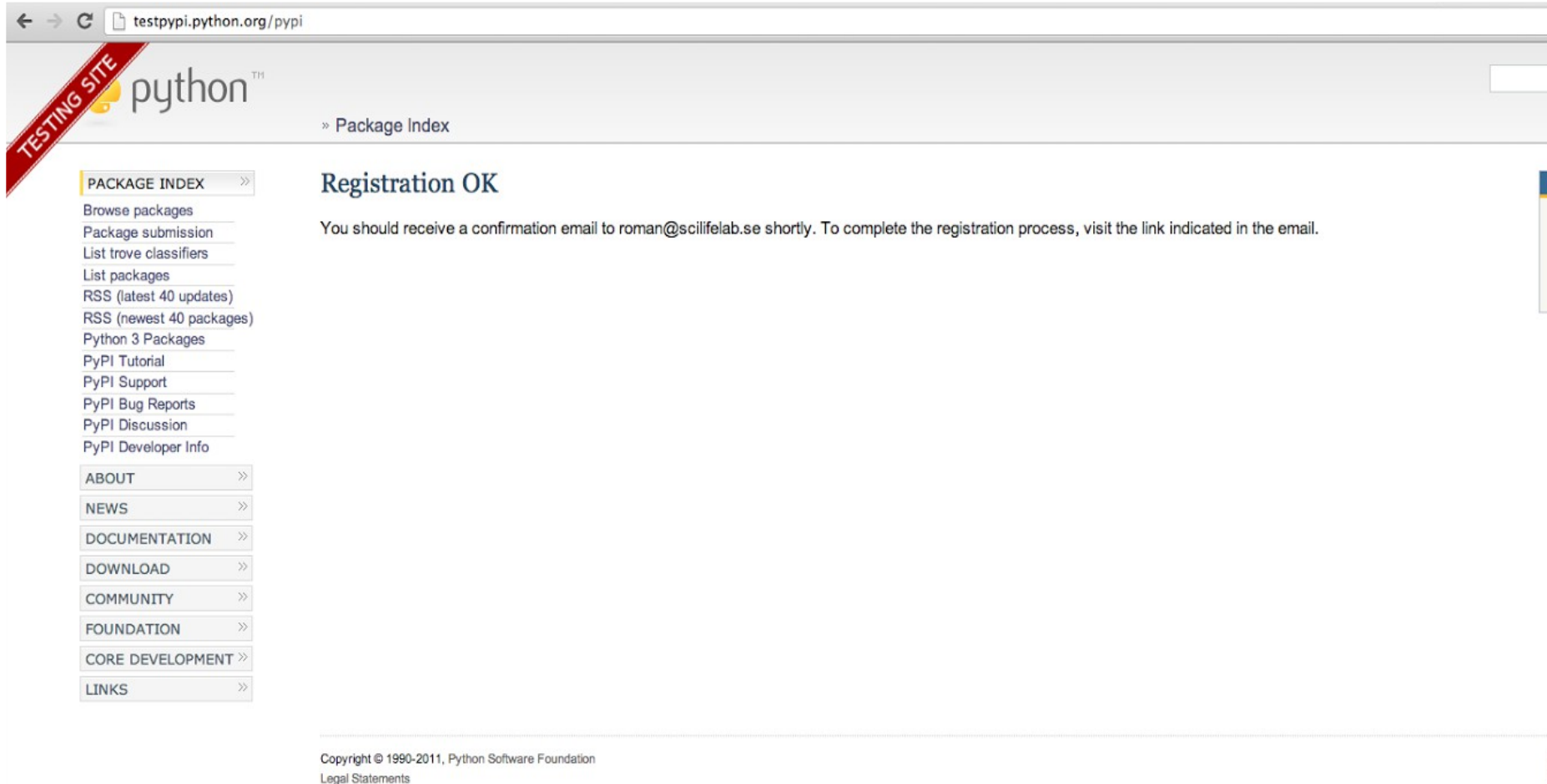
__init__.py

```
$ mkdir -p biostuff/biostuff  
$ cd biostuff  
$ touch LICENSE.txt README.txt  
$ touch biostuff/__init__.py  
$ touch setup.py
```

<http://guide.python-distribute.org/>



http://testpypi.python.org/pypi



The screenshot shows a web browser window with the address bar displaying `testpypi.python.org/pypi`. The page features a header with the Python logo and a red diagonal banner on the left that reads "TESTING SITE". Below the header, the main content area is titled "» Package Index" and "Registration OK". A message states: "You should receive a confirmation email to roman@scilifelab.se shortly. To complete the registration process, visit the link indicated in the email." On the left side, there is a sidebar menu with categories like "PACKAGE INDEX", "ABOUT", "NEWS", "DOCUMENTATION", "DOWNLOAD", "COMMUNITY", "FOUNDATION", "CORE DEVELOPMENT", and "LINKS", each with a list of sub-links. The footer contains copyright information: "Copyright © 1990-2011, Python Software Foundation" and "Legal Statements".

testpypi.python.org/pypi

TESTING SITE

python™

» Package Index

PACKAGE INDEX »

- Browse packages
- Package submission
- List trove classifiers
- List packages
- RSS (latest 40 updates)
- RSS (newest 40 packages)
- Python 3 Packages
- PyPI Tutorial
- PyPI Support
- PyPI Bug Reports
- PyPI Discussion
- PyPI Developer Info

ABOUT »

NEWS »

DOCUMENTATION »

DOWNLOAD »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

LINKS »

Registration OK

You should receive a confirmation email to roman@scilifelab.se shortly. To complete the registration process, visit the link indicated in the email.

Copyright © 1990-2011, Python Software Foundation
Legal Statements

python setup.py install

(devel) biostuff \$ python setup.py install

running install

running build

running build_py

creating build

creating build/lib

creating build/lib/biostuff

copying biostuff/__init__.py -> build/lib/biostuff

running install_lib

creating /Users/roman/.virtualenvs/devel/lib/python2.7/site-packages/biostuff

copying build/lib/biostuff/__init__.py -> /Users/roman/.virtualenvs/devel/lib/python2.7/site-packages/biostuff

byte-compiling /Users/roman/.virtualenvs/devel/lib/python2.7/site-packages/biostuff/__init__.py to __init__.pyc

running install_egg_info

Writing /Users/roman/.virtualenvs/devel/lib/python2.7/site-packages/biostuff-0.1dev-py2.7.egg-info

setup.py

```
from distutils.core import setup
```

```
setup(  
    name='biostuff',  
    version='0.1dev',  
    author='John Doe',  
    author_email='johndoe@example.com',  
    url='example.com',  
    packages=['biostuff'],  
    license='GPLv3',  
    long_description=open('README.txt').read(),  
)
```

pastertool

```
$ pip install pastescript
```

```
$ paster create myapp
```

```
$ cd myapp/
```

Your ~/.pypirc

```
(devel)romanmac:biostuff roman$ cat ~/.pypirc
```

```
[distutils]
```

```
index-servers =
```

```
    testpypi
```

```
[testpypi]
```

```
repository: http://testpypi.python.org/pypi
```

```
username:scilifelab
```

```
password:<sharedpassword>
```


Register with testpypi

```
$ python setup.py sdist register -r testpypi
```

running sdist

running check

warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)

writing manifest file 'MANIFEST'

creating biostuff-0.1dev

creating biostuff-0.1dev/biostuff

making hard links in biostuff-0.1dev...

hard linking README.txt -> biostuff-0.1dev

hard linking setup.py -> biostuff-0.1dev

hard linking biostuff/__init__.py -> biostuff-0.1dev/biostuff

creating dist

Creating tar archive

removing 'biostuff-0.1dev' (and everything under it)

running register

Registering biostuff to <http://testpypi.python.org/pypi>

Server response (200): OK

Upload to testpypi

```
$ python setup.py sdist upload -r testpypi
```

running sdist

running check

warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)

writing manifest file 'MANIFEST'

creating biostuff-0.1dev

creating biostuff-0.1dev/biostuff

making hard links in biostuff-0.1dev...

hard linking README.txt -> biostuff-0.1dev

hard linking setup.py -> biostuff-0.1dev

hard linking biostuff/__init__.py -> biostuff-0.1dev/biostuff

Creating tar archive

removing 'biostuff-0.1dev' (and everything under it)

running upload

Submitting dist/biostuff-0.1dev.tar.gz to <http://testpypi.python.org/pypi>

Server response (200): OK

Try to install your own package

```
$ pip install -i http://testpypi.python.org/simple biostuff
```

```
$ pip install -i http://testpypi.python.org/simple biostuff
```

```
Downloading/unpacking biostuff
```

```
Downloading biostuff-0.1dev.tar.gz
```

```
Running setup.py egg_info for package biostuff
```

```
Installing collected packages: biostuff
```

```
Running setup.py install for biostuff
```

```
Successfully installed biostuff
```

```
Cleaning up...
```

Why document code

1. You protect yourself against workers leaving with legacy application knowledge.
2. You shorten the ramp-up time of the new employees that take over the application.
3. Automation over documentation.
 - a. Please, no command lists. Script it as much as you can.
 - b. Documentation must be kept clear, simple, straightforward.

Documenting

● User level

```
def get_barcode_metrics(workdir):
```

```
    """Parse the [lane]_*_bc.metrics files in the *_barcode directories into
    a dictionary.
```

```
    If the samples have been demultiplexed by CASAVA, parses the
    Unaligned/Basecall_Stats_*/Demultiplex_Stats.htm file for barcode metrics.
```

```
    """
```

● Programmer level (useful in API's)

```
def _set_sample_table_values(sample_name, project_sample, barcode_seq, ordered_million_reads, param):
```

```
    """Set the values for a sample that is to appear in the final table.
```

```
    :param sample_name: string identifier of sample
```

```
    :param project_sample: project sample dictionary from project summary database
```

```
    :param barcode_seq: barcode sequence
```

```
    :param ordered_million_reads: the number of ordered reads
```

```
    :param param: project parameters
```

```
    :returns: vals, a dictionary of table values
```

```
    """
```



Output formats: HTML (including Windows HTML Help), LaTeX (for printable PDF versions), manual pages, plain text

Extensive cross-references: semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information

Hierarchical structure: easy definition of a document tree, with automatic links to siblings, parents and children

Automatic indices: general index as well as a module index

Code handling: automatic highlighting using the [Pygments](#) highlighter

Extensions: automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and more

IP[y]: IPython

Interactive Computing

Powerful Python shells (terminal and Qt-based).

A web-based notebook with the same core features but 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.

Organization++

- **MANDATORY:** We are going to look at your packages in **pypi's test repository** and **github**.
 - We will evaluate you on this.
 - Ask TA's to get into the mailing list.
 - Make sure you are part of the python-course Github organization, this is mandatory:
 - <http://www.github.com/pythonkurs>
 - [Send a mail with your github username to the mailing list.](#)

Wrap up

- We set up a python virtual environment.
- Understood what version control is.
- Configured a hosted DVCS account.
- Packaged our own software in a standard way to ease publishing and redistribution.