

Updates (2020)

As you will notice, some of the tools related to the best practices have improved. Note that most best practices remain the same, though some may not be relevant any more.

I have compiled here a few updates that will get you better prepared in using the tools.

Versioning:

Git has become the standard for versioning, though many other tools still exist.

GitHub repositories are commonplace, and those in educational institutes can create private as well as public repos. Besides versioning they are a great tool for collaborative work.

<https://github.com/>

Programming:

Python is widely used as it is a first class language, the syntax is very readable, and there are a large number of libraries/modules available as a large number of users and companies are using it. Especially with deep learning there are many libraries that you can use out of the box (but don't - until you understand at least the basics)

<https://www.python.org/>

R is also very widely used, especially for statistical purposes, but also in the bio-world.

<https://www.r-project.org/>

Both these languages can be used from inside *Jupyter notebooks*, and in various forms these have become a norm too for quick prototyping and sharing.

<https://jupyter.org/>

For deep learning it is helpful to have GPUs. Google have create a colab, or *colab* for short, for you to play with the deep learning libraries using a Jupyter notebook interface where you can use GPUs (and TPUs) for free.

<https://colab.research.google.com/>

Benchmarking (Python):

<https://pyperformance.readthedocs.io/>

<https://dzone.com/articles/python-101-an-intro-to-benchmarking-your-code>

Testing (Python):

<https://docs.python.org/3/library/unittest.html>

<https://wiki.python.org/moin/PythonTestingToolsTaxonomy>

Debugger (Python):

<https://docs.python.org/3/library/pdb.html>