

Язык программирования Python

Для выполнения анализа экспериментальных данных и различного рода моделирований, я использую Python — универсальный интерпретируемый язык программирования, который прост в изучении, легко расширяется и имеет огромное сообщество разработчиков.

Существует несколько удобных для использования дистрибутивов Python, среди них:

1. Чистый Python с сайта <https://www.python.org>
2. Anaconda/Miniconda с сайта <https://docs.anaconda.com/anaconda/install/> or <https://docs.conda.io/en/latest/miniconda.html>
3. Mambaforge с сайта <https://github.com/mamba-org/mamba>
4. Enthought Canopy <https://assets.enthought.com/downloads/>
5. WinPython <https://winpython.github.io> и полезная для этого дистрибутива библиотека пакетов <https://www.lfd.uci.edu/~gohlke/pythonlibs/>

Как правило, чистый python (1) лучше не устанавливать, так как придется самому вести настройку среды выполнения, компиляторов если таковые потребуются и прочее. Лучше воспользоваться одним из вариантов (2-4). Они представляют пользователю законченную среду окружения с библиотекой пакетов и программ. В случае anaconda/miniconda вы получаете еще и удобный пакетный менеджер с возможностью скачать версии компиляторов c/c++/fortran для сборки пакетов для python. По сравнению с pip он скачивает не только установочный Python-пакеты, но и необходимые зависимости — динамические библиотеки.

В своей работе я использую пакет Miniconda с python версии 3.8, во-первых из-за того, что при установке загружаются и устанавливаются только необходимые пакеты, во-вторых, существуют сборки этого пакета под основные типы операционных систем, и в-третьих — он из коробки поддерживает средства создания и управления виртуальным окружением, которое полностью изолируется от других виртуальных окружений python, установленных на компьютере.

На сегодняшний момент существует 2 ветки python: python 2 и python 3, первая из них еще поддерживается, но активно не развивается, вторая — активно развивается и является стандартом языка python во всех unix-подобных операционных системах

Для своей работы я устанавливаю следующий набор пакетов:

- [Numpy](#) — библиотека для работы с многомерными массивами
- [SciPy](#) — библиотека — надмножество NumPy, представляет набор функций для анализа (интерполяция, обработка сигналов, интегрирование и прочее..)
- [Matplotlib](#) — библиотека дву- и трехмерной графики
- [Pandas](#) — библиотеками для работы с табличными данными и рядами
- [IPython](#) — оболочка -интерпретатор

если вы планируете работать в области data-science или deep learning, возможно нужно будет установить пакет <https://scikit-learn.org/stable/> или аналогичный по функциональности.

В случае, если для работы нужно использовать методы минимизации, в этом случае можно дополнительно установить LmFit. Это оболочка над функциями минимизации из пакета SciPy, которая облегчает работу с функциями. Добавляет ограничения на диапазон принимаемых значений искомых параметров.

Интересным является и пакет NLOpt. Здесь также реализовано множество алгоритмов глобальной и локальной минимизации/максимизации как с аналитически вычисляемым градиентом, так и с его оценкой методом конечных разностей. Важно, если вы не используете Miniconda, и работаете в windows, то NLOpt лучше ставить из исходников.

Пакеты для работы с научными форматами данных

Если вдруг вам в работе придется столкнуться с файлами netCDF и/или HDF5/HDF4 вы можете работать с ними с помощью следующих пакетов:

1. [netCDF4](#)
2. [h5py](#)
3. [pytables](#)

Первый из них поддерживает работу с файлами netCDF версии 3 и 4, последние два ориентированы на работу с форматом hdf5.

From:

<http://lidarbackup.dvo.ru/dokuwiki/> - **Записки репетитора**

Permanent link:

http://lidarbackup.dvo.ru/dokuwiki/doku.php/posts:python_the_proglang



Last update: **2022/05/03 16:36**