Эта статья не сможет охватить все модули python потому что их очень много. Поэтому мы поговорим про самые интересные и полезные. Дальше вы сможете понять на что способен этот язык. Список отсортирован по алфавиту.

* **audioop** - это библиотека для работы со звуком. Она содержит несколько полезных функций для обработки записей 8, 16, 24 или 32 бит. Модуль поддерживает различные кодировки. Есть поддержка конвертации форматов, настройки битов, и множество других возможностей.
* **base64** - шифрование base64 стало очень популярным в последнее время. Эта библиотека позволяет шифровать бинарные данные в читаемые ASCII символы, а затем обратно раскодировать в бинарные данные. Можно кодировать не только последовательности байт, но и обычные строки.
* **calendar** - как следует из названия, этот модуль позволяет работать с календарем. Вы можете выводить календарь, так же как это делает утилита cal в Linux. По умолчанию, началом недели считается понедельник, но можно поменять настройки на воскресение. Вы можете просматривать информацию о днях недели, месяца, года, выводить списки и многое другое.
* **cgi** - несмотря на то, что лидером в плане серверного языка программирования является PHP, Python тоже поддерживает такую возможность. Модуль cgi позволяет интерпретатору обрабатывать скрипты по запросу веб-сервера и возвращать ему же результат обработки. С помощью модуля в скрипте мы можем получить переменные, переданные браузером с помощью GET или POST, а также влиять на отправляемые данные.
* **configparser**- это простая библиотека, которая позволяет разбирать содержимое простейших конфигурационных файлов формата ini. Такие файлы очень часто используются в Windows. Вы можете не только читать содержимое файлов, но и изменять его.
* **csv** - модуль, позволяющий работать с форматом файлов csv (Command Separated Values). Этот формат очень популярен при импорте и экспорте из различных таблиц или баз данных. Модуль имеет два класса, reader и writer, которые позволяют читать и записывать данные в формат csv.
* **curses** - наверное, уже многие слышали про движок псевдографического интерфейса curses. Он позволяет довольно просто реализовать простой графический интерфейс в терминале. С помощью этого модуля такая функция появляется в Python.
* **datetime**- как и большинство языков высокого уровня, python позволяет работать с датой и временем. Эта библиотека реализует набор методов для получения информации, преобразования, изменения даты и времени. Можно преобразовать дату в строку или прочитать ее из строк различных форматов. Также можно выполнять арифметические операции с датами и временем.
* **decimal** - этот модуль содержит функции для быстрого преобразования чисел с плавающей точкой. Также содержит несколько дополнительных возможностей для встроенного типа float.
* **difflib** - эта библиотека содержит набор функций для сравнения различных последовательностей. Например, можно сравнивать файлы, строки, различную информацию в HTML и многое другое.
* **email**- python поддерживает обработку email сообщений на уровне языка. Модуль не реализует никаких методов для отправки сообщений через SMTP или NNTP, этим занимаются другие протоколы. Здесь есть функции для разбора структуры email сообщений, проверки списка почты, преобразования и много другого.
* **gettext** - этот модуль реализует функции локализации и интернационализации L10N для ваших программ на Python. Поддерживается стандартное API GNU gettext так и свое собственное API на основе классов. Все модули пишутся на вашем нативном языке, а затем к программе прикрепляется каталог для перевода на другие языки.
* **gzip, zlib** - библиотеки python для работы со сжатыми данными. Вы можете не только распаковывать и упаковывать файлы, но и работать со строками, а также использовать пароли.
* **hashlib** - этот модуль python предоставляет интерфейс для получения различных хэшей для данных. Поддерживаются такие алгоритмы:  SHA1, SHA224, SHA256, SHA384, и SHA512, а также MD5.
* **html, http** - эти модули работают в одной и той же области. Модуль http позволяет работать с интернет ресурсами по протоколу HTTP, отправлять запросы GET/POST, принимать запросы, обрабатывать Cookie и фактически реализовать свой клиент или сервер на Python. Библиотека html, в свою очередь, позволяет выполнять разбор html страниц.
* **io** - это базовая библиотека, которая содержит основные функции для работы с потоками ввода/вывода. Поддерживаются различные виды потоков, текстовые, бинарные и RAW потоки. Каждый поток может иметь несколько атрибутов, это разрешения на только чтение, только запись и чтение запись.
* **itertools**- недавно операторы итераций начали появляться в PHP. Язык программирования Python тоже имеет средства для организации итераций, похожих на Haskell, APL и SML. Модуль использует эффективные методы работы с памятью, а также имеет некоторые дополнительные функции.
* **json** - библиотека python для работы с очень популярным сейчас форматом передачи данных - json. Есть функции как для разбора формата, так и для создания объектов для отправки.
* **logging** - модуль для логирования в программах Python. Библиотека реализует удобную систему логирования, которая используется в стандартных модулях. Ее преимущество в том, что вы можете отключить ведение лога в любой момент одной строчкой или изменить его подробность.
* **match** - библиотека содержит стандартные функции для работы с математикой. Вычисление корня, синусов, косинусов и другие подобные функции.
* **os** - одна из самых важных библиотек python. Она предназначена для взаимодействия с операционной системой. Через нее вы можете работать с файлами, получить информацию об интерфейсах операционной системы и другое.
* **pathlib** - позволяет работать с путями в файловой системе. Можно преобразовывать пути из одного типа в другой, выполнять с ними различные операции.
* **random** - модуль реализует генератор псевдо-случайных чисел. Он работает на основе генератора случайности вашей операционной системы.
* **re** - очень часто используемый модуль python, который содержит базовый набор функций для работы с регулярными выражениями синтаксиса perl. Есть методы для решения различных задач, таких как поиск, замена, редактирование, удаление и многое другое.
* **socket** - python поддерживает работу с сокетами напрямую. Вы можете без модуля http или url подключатся к любому системному или сетевому сокету и использовать его.
* **sqlite** - большинство программ в среде веб используют базу данных. Расширение sqlite позволяет вашей программе использовать высокопроизводительную базу данных, которая полностью хранится в одном файле, в папке с программой.
* **ssl** - библиотека, которая позволяет работать с сертификатами ssl, используется для получения html страниц по протоколу https.
* **string** - еще один часто используемый модуль, который содержит множество функций для работы со строками. Поддерживаются большинство функций, которые есть в других языках, например, слияние строк, удаление лишних символов, замена, поиск и так далее.
* **threading** - библиотека реализует поддержку многопоточности для python. Здесь содержатся методы для управления потоками и получения информации о них;
* **time** - по возможностях и назначению эта библиотека похожа на datetime. Только ее методы рассчитаны на работу с датой и временем. Здесь реализовано множество функций стандартной библиотеки Си.
* **tkinter** - это стандартные библиотеки python 3 для реализации графического интерфейса программ с помощью инструментария Tk GUI. Этот интерфейс будет работать как в Windows, так и в Linux системах.
* **urllib** и **urllib2** - эти библиотеки python позволяют реализовать простой парсер или браузер на python. Вы можете в несколько строк получать и разбирать содержимое веб-страниц, работу с кукси, заголовками и другими вещами библиотека берет на себя.
* **xml** - библиотека, которая помогает анализировать структуры XML, а также добавлять в структуру новые теги, менять значения существующих.
* **Numpy** – тоже будет

Uuid

Наверняка вам приходилось генерировать для пользователей ID, или рассылать покупателям промокоды, или делать ещё что-то, где нужно создать уникальные последовательности. [UUID](https://docs.python.org/2/library/uuid.html) вам в этом поможет:

import uuid

print uuid.uuid4()

# e7bafa3d-274e-4b0a-b9cc-d898957b4b61

И если вы переживаете, что ID кончатся, то не стоит: их количество [сравнимо](https://wisdomofganesh.blogspot.ru/2009/04/guidness-of-uuid.html) с количеством атомов во вселенной.

**1. Requests**

Имея больше 26 тысяч звезд на GitHub и тысячи скачиваний ежедневно, [Requests](http://www.python-requests.org/en/master/) занимает первую строчку рейтинга самых популярных и полезных библиотек для разработки на Python. Эта библиотека используется в проектах Spotify, Microsoft, NPR, Heroku, Amazon, BuzzFeed, Reddit, Twitter, Mozzila (список можно продолжать еще долго).

Requests уменьшает тяжелую ручную работу и автоматизирует такие задачи как добавление строки запроса к URL, шифрование форм отправки данных, постоянное HTTP-соединение, объединение HTTP-соединений и т. д. путем использования urllib3.

Для облегчения разработки также обеспечивается автоматическое дешифрование контента. Библиотека предлагает загрузку других функций, таких как журнал аутентификации, элегантные key/value cookies, Unicode-представление для аттрибута body объекта response, потоковое скачивание, тайм-аут соединения, поддержка .netrc.

В общем, Requests – не лишняя библиотека для любого веб-проекта.

**2. Simplejson**

Это еще одна очень популярная библиотека и одна из самых скачиваемых. [Simplejson](https://pypi.python.org/pypi/simplejson/3.8.1) это быстрый, точный и расширяемый json кодировщик и декодировщик, который поддерживает Python 3.3+ с обратной совместимостью с Python 2.5.

Simplejson написан на Python и не имеет внешних зависимостей. Он также включает расширения, написанные на С для представлений с особо высоким быстродействием. Помимо дефолтной UTF-8, декодер Simplejson может оперировать строками JSON, поступающими в любой кодировке.

**3. SQL Alchemy**

Базы данных — неотъемлемая часть разработки приложений, а в мире Python [SQLAlchemy](http://www.sqlalchemy.org/) — необходимая библиотека для работы с базами данных. Предлагая полный комплект устойчивых моделей предметных областей для высокоэффективного доступа к базам данных, она действует как набор инструментов SQL и объектно-реляционное отображение.

Вызывает интерес модель отображения данных, в которой различные классы отображаются в базу данных различными способами, что позволяет объектной модели и схеме базы данных быть изначально развязанными.

Разработчики получают полный контроль и обзор конструкции SQL, ничто не прячется за стенами упаковщиков.

Подход этой библиотеки гораздо более современный и эффективный, если сравнивать с доступными инструментами SQL / ORM, а потому в списке самых полезных библиотек Python она занимает высокое место.

После выхода последнего релиза SQLAlchemy выбивается в лидеры; ее используют такие организации как Freshbooks, Survey Monkey, Mozilla, reddit, Yelp и многие другие.

**4. TensorFlow**

[TensorFlow](https://www.tensorflow.org/install/) идет за основными библиотеками и погружает вас мир машинного интеллекта. Это опенсорсная библиотека Python, успешно применяемая для произведения расчетов с использованием графов потоков данных.

Вычисление изначально представляется в форме графов, где каждый узел графа предназначен для выполнения математических операций. Фактическое вычисление, тем не менее, производится по запросу, что позволяет повысить продуктивность сложных расчетов.

Библиотека учитывает нужды комплексных вычислений: она обслуживает вычисления, распределенные на CPU/GPU и несколько систем, заботясь о дублировании.

TensorFlow это бесплатная библиотека с открытым исходным кодом. Она была разработана инженерами Google, работающими над проектом Google Brain.

**5. Scrapy**

[Scrapy](https://scrapy.org/download/) это фреймворк, использующийся для извлечения структурированных данных с вебсайтов. Это быстрая, простая, расширяемая, легкая в использовании библиотека. Ее использование включает добычу данных, обработку информации, поисковый робот, извлечение данных с помощью API и многое другое.

Scrapy «из коробки» обеспечивает поддержку извлечения данных из HTML/XML с использованием выражений Xpath и селекторов CSS, интерактивную оболочку консоли, экспорт данных в json, csv, xml, бэк-энд хранилище ftp, s3, локальное хранилище, управление cookies и сессиями, сжатие, кэширование, аутентификацию, user-agent спуфинг.

Scrapy пзволяет вам прописывать правила для извлечения данных. Библиотека написана на Python, ее конструкция предполагает возможность расширения за счет нового функционала, и вы можете перенести ее для запуска на Linux, Windows, Mac и BSD.

**6. Matplotlib**

Библиотека двумерных числовых построений . Она отлично подходит для анализа данных и создания высококачественных фигур различного формата, включая интерактивные графики и схемы.

[Matplotlib](https://scrapy.org/download/) можно использовать а скриптах, написанных на Python, в оболочке IPython и на множестве разнообразных серверов веб-приложений. Это значительно упрощает сложные задачи на построение. Написав небольшой кусочек кода, вы сможете генерировать графики, гистограммы, диаграммы разброса данных.

Модули pyplot и Ipython combo предлагают MATLAB в качестве интерфейса для построения простых графиков. Вы можете полностью контролировать вид линий, свойства системы координат и тому подобное. Это, несомненно, одна из лучших библиотек для числовых построений, доступных для Python.

Такие инструменты как basemap, cartopy, mplot3d, axes\_grid увеличивают возможности Mataplotlib.

Эта библиотека – детище Джона Хантера. Она применяется в тысячах исследовательских, научных и издательских проектов.

**7. Scikit-Learn**

Высокоуровневая библиотека, содержащая алгоритмы, подобные random forest, готовые к использованию в проектах, связанных с машинным обучением.

Языком [Scikit-learn](http://scikit-learn.org/stable/install.html) по большей части является Python, но также задействуется Cython для улучшения производительности в некоторых внутренних алгоритмах. Оболочка Cython вокруг LIBSVM используется для поддержки векторной реализации, а LIBLINEAR – для вычислительной регрессии и линейной опоры векторных машин. Scikit-learn также использует CBLAS, который является C-интерфейсом для пользования библиотекой Basic Linear Algebra Subprograms (CBLAS).

Эта библиотека построена на SciPy и распространяется на условиях лицензии 3-Clause BSD с открытым исходным кодом для исследований, а также для коммерческого использования.

**8. Pygame**

Любите разработку игр? [Pygame](http://www.pygame.org/news) предназначенная для создания мультимедиа-приложений с играми включительно. Она построена на мощной Simple Directmedia Library (SDL). Эта библиотека среди прочего используется для обучения детей разработке игр. Имеет открытый исходный код.

Сила Pygame в том, что она не требует OpenGL, позволяет использование множественных CPU для работы в многоядерной системе для супервысокой производительности, а также использование оптимизированных кодов на C и Assembly для внутренних функций.

Pygame весьма портативна и может запускаться практически на любых платформах и операционных системах. Эту библиотеку скачали миллионы людей и очень вероятно, что это одна из лучших игровых библиотек.

**9. Arrow**

[Arrow](https://github.com/crsmithdev/arrow) предназначена для преодоления недостатков встроенного в Python функционала даты и времени, который не совсем понятен и легок. Может быть использована в качестве прекрасной замены для модулей datetime и time в Python.

**10. wxPython**

[wxPython](https://wxpython.org/pages/downloads/) входит в тройку пользующихся наибольшей популярностью GUI-библиотек для Python (остальные две – PyQT и Tkinter). wxPython интуитивна для разработчиков python и проста в использовании, это прекрасная смесь из C++ wxWidget-ов и программирования на Python.

wxPython применяется в качестве расширяющего модуля Python и является кросс-платформенным набором инструментов, запускаемым на многих платформах без нужды в модификациях. Поддерживается на многих платформах, включая Unix, Macintosh OS X и Microsoft Windows (32 bit).

wxPython предлагает тысячи элементов, позволяющих с легкостью создавать трудоемкие и функциональные GUI приложения на Python.

NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

Основным объектом NumPy является однородный многомерный массив (в numpy называется numpy.ndarray). Это многомерный массив элементов (обычно чисел), одного типа.

Наиболее важные атрибуты объектов ndarray:

**ndarray.ndim** - число измерений (чаще их называют "оси") массива.

**ndarray.shape** - размеры массива, его форма. Это кортеж натуральных чисел, показывающий длину массива по каждой оси. Для матрицы из n строк и m столбов, shape будет (n,m). Число элементов кортежа shape равно ndim.

**ndarray.size** - количество элементов массива. Очевидно, равно произведению всех элементов атрибута shape.

**ndarray.dtype** - объект, описывающий тип элементов массива. Можно определить dtype, используя стандартные типы данных Python. NumPy здесь предоставляет целый букет возможностей, как встроенных, например: bool\_, character, int8, int16, int32, int64, float8, float16, float32, float64, complex64, object\_, так и возможность определить собственные типы данных, в том числе и составные.

**ndarray.itemsize** - размер каждого элемента массива в байтах.

**ndarray.data** - буфер, содержащий фактические элементы массива. Обычно не нужно использовать этот атрибут, так как обращаться к элементам массива проще всего с помощью индексов.

**Создание массивов**

В NumPy существует много способов создать массив. Один из наиболее простых - создать массив из обычных списков или кортежей Python, используя функцию numpy.array() (запомните: array - функция, создающая объект типа ndarray):

>>>

**>>> import** **numpy** **as** **np**

**>>>** a = np.array([1, 2, 3])

**>>>** a

array([1, 2, 3])

**>>>** type(a)

<class 'numpy.ndarray'>

Функция array() трансформирует вложенные последовательности в многомерные массивы. Тип элементов массива зависит от типа элементов исходной последовательности (но можно и переопределить его в момент создания).

>>>

**>>>** b = np.array([[1.5, 2, 3], [4, 5, 6]])

**>>>** b

array([[ 1.5, 2. , 3. ],

[ 4. , 5. , 6. ]])

Можно также переопределить тип в момент создания:

>>>

**>>>** b = np.array([[1.5, 2, 3], [4, 5, 6]], dtype=np.complex)

**>>>** b

array([[ 1.5+0.j, 2.0+0.j, 3.0+0.j],

[ 4.0+0.j, 5.0+0.j, 6.0+0.j]])

Функция array() не единственная функция для создания массивов. Обычно элементы массива вначале неизвестны, а массив, в котором они будут храниться, уже нужен. Поэтому имеется несколько функций для того, чтобы создавать массивы с каким-то исходным содержимым (по умолчанию тип создаваемого массива — float64).

Функция zeros() создает массив из нулей, а функция ones() — массив из единиц. Обе функции принимают кортеж с размерами, и аргумент dtype:

>>>

**>>>** np.zeros((3, 5))

array([[ 0., 0., 0., 0., 0.],

[ 0., 0., 0., 0., 0.],

[ 0., 0., 0., 0., 0.]])

**>>>** np.ones((2, 2, 2))

array([[[ 1., 1.],

[ 1., 1.]],

[[ 1., 1.],

[ 1., 1.]]])

Функция eye() создаёт единичную матрицу (двумерный массив)

>>>

**>>>** np.eye(5)

array([[ 1., 0., 0., 0., 0.],

[ 0., 1., 0., 0., 0.],

[ 0., 0., 1., 0., 0.],

[ 0., 0., 0., 1., 0.],

[ 0., 0., 0., 0., 1.]])

Функция empty() создает массив без его заполнения. Исходное содержимое случайно и зависит от состояния памяти на момент создания массива (то есть от того мусора, что в ней хранится):

>>>

**>>>** np.empty((3, 3))

array([[ 6.93920488e-310, 6.93920488e-310, 6.93920149e-310],

[ 6.93920058e-310, 6.93920058e-310, 6.93920058e-310],

[ 6.93920359e-310, 0.00000000e+000, 6.93920501e-310]])

**>>>** np.empty((3, 3))

array([[ 6.93920488e-310, 6.93920488e-310, 6.93920147e-310],

[ 6.93920149e-310, 6.93920146e-310, 6.93920359e-310],

[ 6.93920359e-310, 0.00000000e+000, 3.95252517e-322]])

Для создания последовательностей чисел, в NumPy имеется функция arange(), аналогичная встроенной в Python range(), только вместо списков она возвращает массивы, и принимает не только целые значения:

>>>

**>>>** np.arange(10, 30, 5)

array([10, 15, 20, 25])

**>>>** np.arange(0, 1, 0.1)

array([ 0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9])

Вообще, при использовании arange() с аргументами типа float, сложно быть уверенным в том, сколько элементов будет получено (из-за ограничения точности чисел с плавающей запятой). Поэтому, в таких случаях обычно лучше использовать функцию linspace(), которая вместо шага в качестве одного из аргументов принимает число, равное количеству нужных элементов:

>>>

**>>>** np.linspace(0, 2, 9) *# 9 чисел от 0 до 2 включительно*

array([ 0. , 0.25, 0.5 , 0.75, 1. , 1.25, 1.5 , 1.75, 2. ])

fromfunction(): применяет функцию ко всем комбинациям индексов

>>>

**>>> def** f1(i, j):

**...**  **return** 3 \* i + j

**...**

**>>>** np.fromfunction(f1, (3, 4))

array([[ 0., 1., 2., 3.],

[ 3., 4., 5., 6.],

[ 6., 7., 8., 9.]])

**>>>** np.fromfunction(f1, (3, 3))

array([[ 0., 1., 2.],

[ 3., 4., 5.],

[ 6., 7., 8.]])

**Печать массивов**

Если массив слишком большой, чтобы его печатать, NumPy автоматически скрывает центральную часть массива и выводит только его уголки.

>>>

**>>>** print(np.arange(0, 3000, 1))

[ 0 1 2 ..., 2997 2998 2999]

Если вам действительно нужно увидеть весь массив, используйте функцию numpy.set\_printoptions:

np.set\_printoptions(threshold=np.nan)

И вообще, с помощью этой функции можно настроить печать массивов "под себя". Функция numpy.set\_printoptions принимает несколько аргументов:

*precision* : количество отображаемых цифр после запятой (по умолчанию 8).

*threshold* : количество элементов в массиве, вызывающее обрезание элементов (по умолчанию 1000).

*edgeitems* : количество элементов в начале и в конце каждой размерности массива (по умолчанию 3).

*linewidth* : количество символов в строке, после которых осуществляется перенос (по умолчанию 75).

*suppress* : если True, не печатает маленькие значения в scientific notation (по умолчанию False).

*nanstr* : строковое представление NaN (по умолчанию 'nan').

*infstr* : строковое представление inf (по умолчанию 'inf').

*formatter* : позволяет более тонко управлять печатью массивов.