**10 удивительно полезных базовых функций Python**

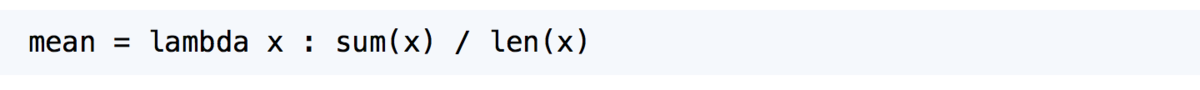
Вчера

**Введение**

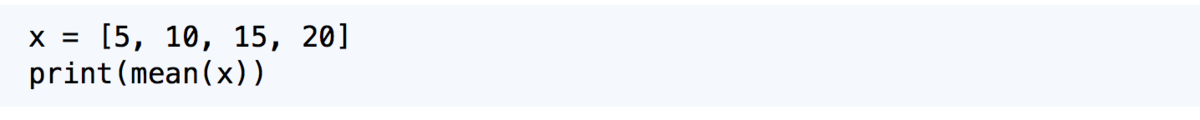
У языка Python есть несколько действительно интересных приемов, которые делают его привлекательным для аудитория научного программирования. Используя эти фишки, вы можете сохранить себе много времени и в целом упростить программирование на Python !

**№1: лямбда**

Некоторое время назад я написал статью о том, почему, на мой взгляд, лямбда-функции в Python делают его гораздо более жизнеспособным языком программирования для статистических вычислений. Кроме того, поскольку Python не подчиняется некоторым строгим объектно-ориентированные правилам, он может быть функциональным и декларативным. Лямбда-выражения, можно использовать для математических операции с любым типом данных, используя одну строку выражения, а не целые функции. Это можно использовать для более простых в программировании вещей, когда не нужно указывать целую функцию со своим именем. Лямбда -выражения можно определить в глобальной области видимости, поскольку это Python, и вывести Python на другой научный уровень, используя функционально-подобный синтаксис и методологию языка, который все еще имеет структуру классов. Хотя для некоторых эта концепция может показаться не очень удачной, ее использование позволяет сэкономить время, память и и может сделать ваш код более лаконичным. Излишне говорить, что для Data Scientist эти операции очень удобны, и возможность выполнять их в одной строке очень эффективно делает ,что делает наш стиль программирования на Python намного проще! В синтаксис это выглядит следующим образом; мы начинаем с установки значения, равного возвращаемому объекта лямбда- выражения , за которым следует переменная, которую мы хотели бы предоставить в качестве позиционного аргумента. Затем выполняем наша операция с использованием этого аргумента в качестве переменной, например поиск среднего:

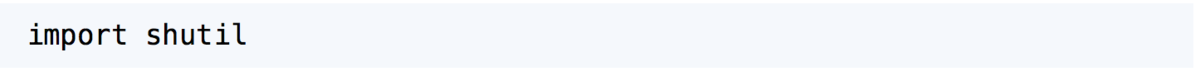


Теперь мы можем называть лама-выражение, как любой метод языка программирования Python:



**№2: Shutil**

Одним из наиболее недооцененных инструментов в арсенале Python является модуль Shutil. Как и другие модули в этом списке, они включены в стандартную библиотеку и могут быть импортированы как вы обычно используете любой модуль на python:

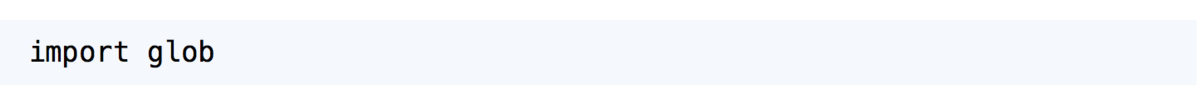


Так что же делает shutil? Модуль shutil - это высокоуровневый интерфейс для Python. Язык программирования для файловой системы в вашей операционной системе. Чаще всего работа с операционной системой , осуществляется с использованием модуля os, но модуль shutil, который менее известен, я думаю, следует использовать чаще. Как часто вам нужно было быстро переместить файл из из каталога в каталог , приходилось ли проделать массу утомительной работы, чтобы сделать это? Shutil решает эти классические проблемы с файлами и таблицами размещения, создавая высокоуровневый решение общей проблемы.. Для всех ваших реализаций dev-ops это, безусловно, ключ к экономии времени, Shutil позволяет сделать файловые операции намного быстрее, чем с использованием os. Ниже приведены некоторые примеры высокоуровневых вызовов, которые модуль shutil предоставляет для базовых файловых операциях:

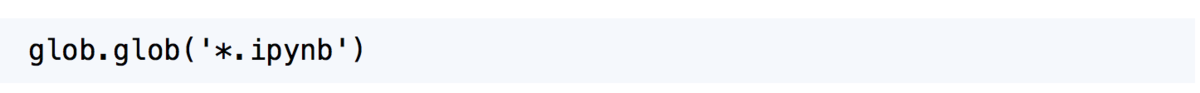


**№3: glob**

Хотя модуль glob может быть не таким крутым, как shutil, но почти таким же полезным для программирования, как Lambda -выражения , он довольно полезен для определенных ситуаций. В Модуль glob используется для поиска в каталогах файлов нужных типов. Это означает, что его можно использовать для агрегирования данных о файлах и их расширениях, хранящихся на вашем компьютере.Конечно, это модуль импортируется так, как вы могли бы ожидать)



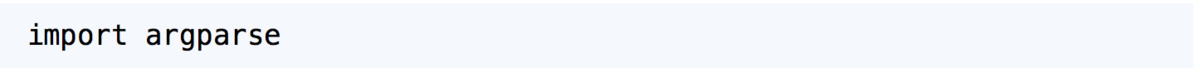
Эта функция будет выполнять любой типичный поиск файлов. Это включает файл- выполняет поиск с использованием синтаксиса Unix, Windows т.е. \*, / и т. д. Попробуйте сами :



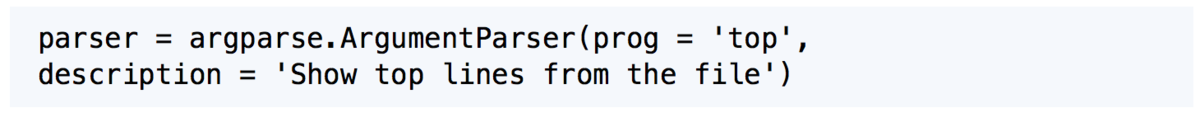
Эта функция список имен файлов, которые соответствуют файлам с , Это также может использоваться для агрегирования данных или для работы с файлами. которые хранят данные!

**№4: argparse**

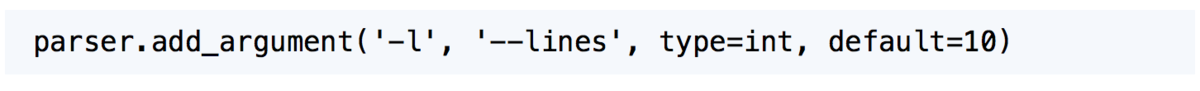
Модуль argparse предоставляет более надежный и глубокий метод анализа команд: строковые аргументы. Многие инструменты разработки используют эту концепцию, чтобы с ними можно было взаимодействовать. с помощью из командной строки Unix. Это полезно для многих инструментов разработчика, которые обычно используется разработчиками Python и специалистами по данным. Отличным примером этого будет быть модулем Python Gunicorn, который может обрабатывать переданные аргументы командной строки длярабочие и различные другие параметры. Как и раньше, этот модуль необходимо импортировать в использоваться:



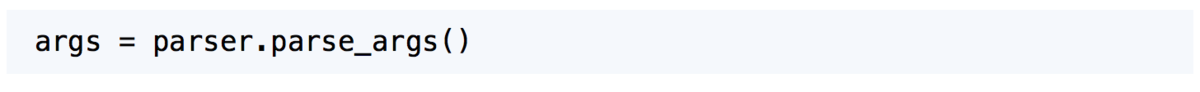
Чтобы использовать этот модуль, мы создадим новый тип, который будет анализатором аргументов:



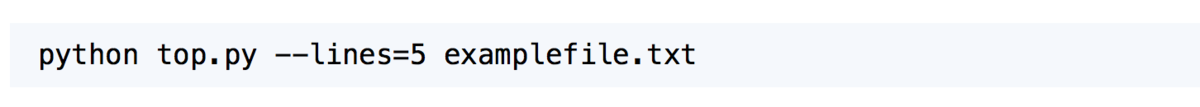
Теперь мы добавляем аргументы в наш синтаксический анализатор, используя добавление аргумента в этот новый читатель. За в этом случае мы создадим аргумент, который можно передать, чтобы определить количество строки, которые мы хотим напечатать из каждого файла:



Я добавил несколько аргументов ключевых слов, один из которых предоставит тип данных, который мы ожидает передачи для этого аргумента, а другой является значением по умолчанию, когда файл вызывается без этого аргумента. Теперь мы можем получить аргументы, вызвав parse\_args () для нашего нового типа парсера аргументов:



Теперь мы можем вызвать этот файл Python для компиляции, а также предоставить необходимые параметры легко из Bash.



Излишне говорить, что это определенно может пригодиться. Некоторые примечательные ситуации, когда у меняиспользуется этот модуль при работе с Crontab, который может запускать скрипты в определенных Unix отметки времени. Кроме того, этот сценарий также можно использовать для супервизоров, которые запускают Bash.команды без участия пользователя в качестве рабочего. Если вы хотите узнать больше о crontab, у меня есть целая статья о том, почему мне это нравится:

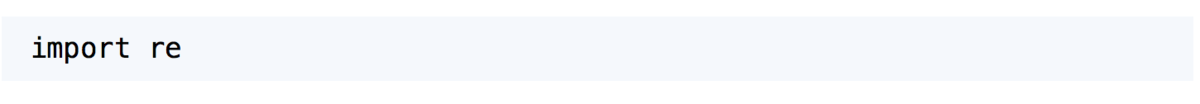
https://towardsdatascience.com/why-crontab-is-the-ultimate-data-science-back-end-tool-e3f212f2b13d

Кроме того, если вы хотите узнать больше о некоторых других приложениях, которые я использую дляразвертывать конечные точки и веб-сайты Pythonic, у меня есть целая статья с подробностями от А до Яразвертывание с самообучающейся моделью машинного обучения, которую вы можете проверить здесь:

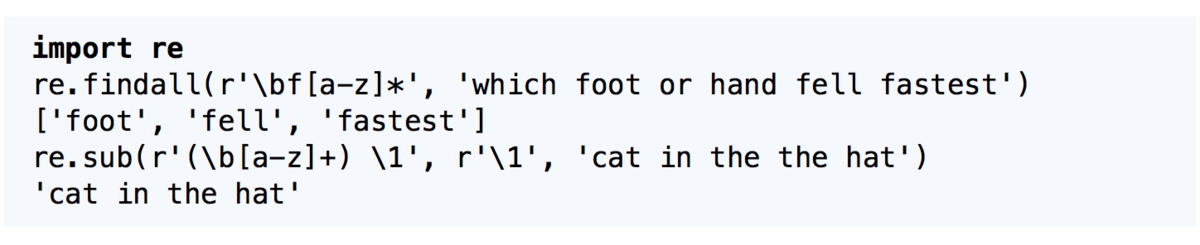
<https://medium.com/chifi-media/build-a-beautiful-model-that-trains-itself-the-easy-way-a0d31b611cee>

**№5: import re**

Из всех инструментов, которые я рассмотрел сегодня, модуль re является самым критичным.недооценен. Буквы «R» и «E» в слове re - это сокращение от регулярных выражений. Модуль re используется для синтаксического анализа строк с помощью регулярных выражений и предоставления дополнительных возможностей для работы с строки в Python. Сколько раз вам приходилось принимать алгоритмические решения на основе на функции, которые есть в строковом классе, например str.split ()? Больше этого не нужно, потому что регулярно выражения намного проще и их намного проще использовать!



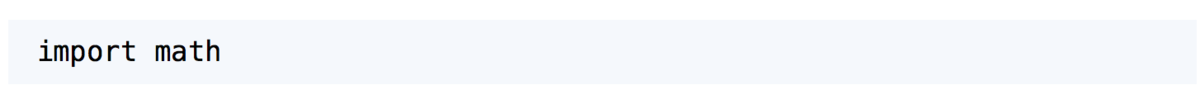
Модуль re, в отличие от некоторых других в этом списке, предоставляет множество очень полезных функции;особенно для работы с огромными объемами данных, как это обычно делают ученые. При этом два примера, которые помогут вам начать, - это sub () и findall (). функции.



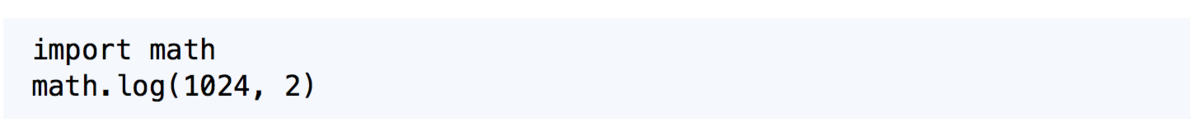
Излишне говорить, что это может быть очень полезно для работы с данными, которые часто встречаются. в этом мире.

**№6: Math**

Хорошо - выслушай меня. Математика может быть не лучшим модулем в истории стандартабиблиотека, но она часто бывает полезной. Это, конечно, еще более верно для научные вычисления.Математический модуль дает доступ ко всему от греха и cos вплоть до логарифмов. Все это может пригодиться при работе с алгоритмы, само собой разумеется, и многие пакеты действительно зависят от этого базового модуля, чтобы получить много чего сделано.

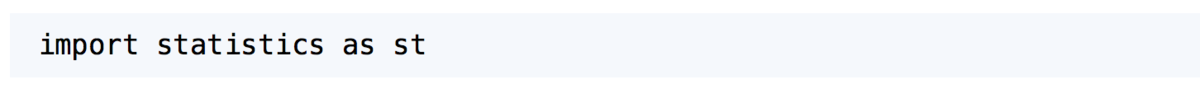


Модуль, безусловно, может сэкономить время, легко выполняя математические операции. доступно без зависимостей. В этом примере я продемонстрирую log () функции, но вы наверняка могли бы глубже изучить этот модуль и найти целый мир математических способностей!

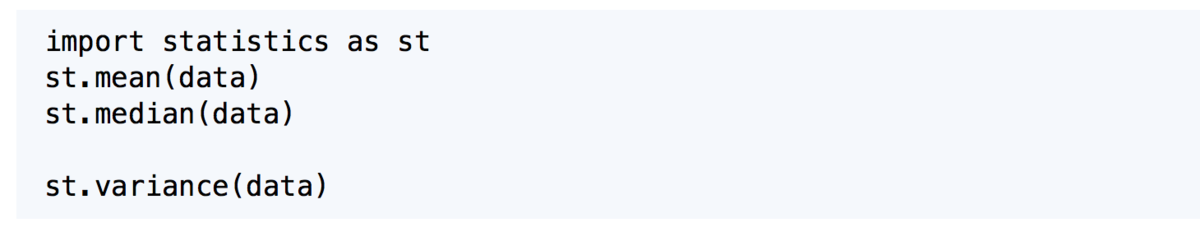


**№7: Statistics**

Еще один пакет, который почти наверняка не представляет сложности для научных вычислений, - это модуль статистики. Этот модуль предоставляет некоторую элементарную статистику, которая может не соответствовать как что-то вроде SciPy, но достаточно часто для анализа данных. Хотя выводная и байесовская статистика не может быть основным направлением этого пакета, математические значения, такие как среднее или стандартное отклонение, могут многое рассказать о данные, с которыми вы, возможно, работаете. Я часто называю этот модуль st, но у меня также есть видел, что он имеет псевдоним stc или sts, не путать с общим псевдонимом scs для Scipy.stats.

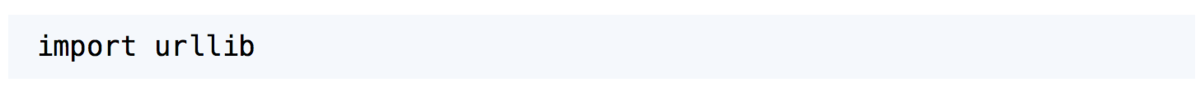


Этот модуль предоставляет множество полезных функций, которые, безусловно, стоят проверка! Самое замечательное в этом пакете то, что он всегда будет там, будь то нужен он вам или нет, и никаких зависимостей не потребуется. Давайте посмотрим на некоторые основные универсальные статистические операции, которые легко предоставляются с помощью этого пакета:

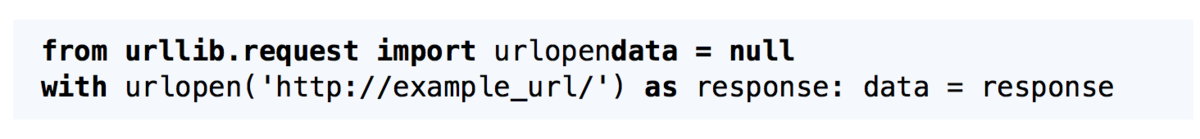


**№8: urllib**

Хотя многие из этих пакетов могут быть не так хорошо известны, модуль urllib наиболее конечно исключение из этого правила. Этот модуль можно использовать для обработки многих распространенных веб-задачи, такие как запросы. Импортируем!



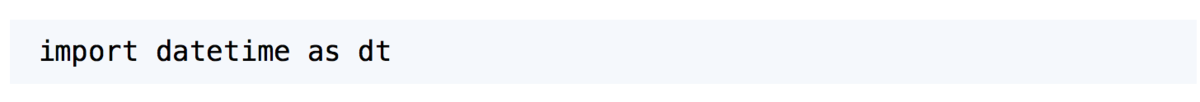
Часто запросы - это то, что специалисты по обработке данных делают с зависимостью. Обычный тоже Под это описание подходит модуль запросов Flask. Этот модуль - еще один распространенный использованное решение для запросов и имеет больше функций, чем urllib, например возможность запрашивать аргументы запроса из http-запроса. При этом, это может быть удобно использовать стандартную библиотеку и меньше беспокоиться о зависимостях, если проект не требует этих дополнительных функций.



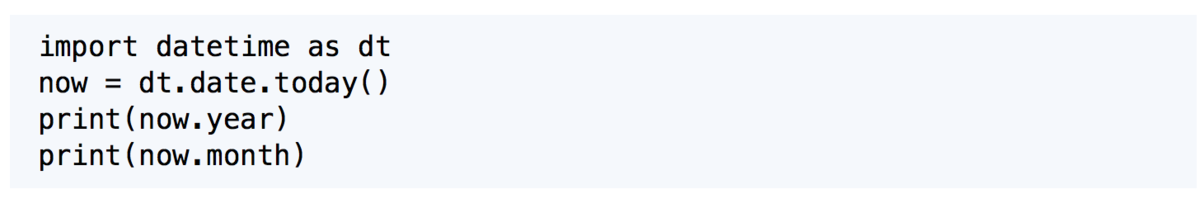
Хотя функция urlopen, безусловно, является отличным примером общего использования urllib, в этом относительно небольшом пакете есть много функций, и это определенно инструмент, который, на мой взгляд, стоит изучить.

**№9: datetime**

Еще один отличный пример инструмента, который довольно часто встречается в научных вычислениях, - это тип даты и времени. Очень часто данные имеют метки времени, иногда временные метки - это даже прогностическая функция, которая будет обучать нашу модель. Тот как говорится, работа с датой и временем может быть относительно важной концепцией для Специалист по данным. Мы можем импортировать модуль даты и времени именно так, как вы ожидаете; это очень часто можно увидеть этот модуль с псевдонимом dt, поэтому для этого примера я собираюсь сделать именно это:

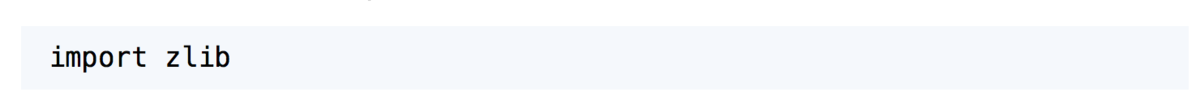


Теперь мы можем создавать типы даты и времени и работать с типичным синтаксисом даты и времени с такие свойства, как год, месяц и день. Это невероятно полезно для переформатирования, разбора, и работа с отдельными разделами дат в ваших данных. Давайте посмотрим на некоторые базовый функционал из этого пакета:

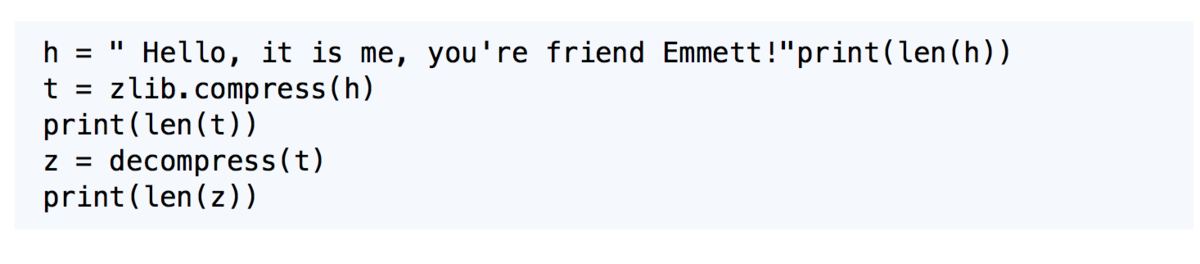


**№10: zlib**

Последний претендент в этом списке - модуль zlib. Модуль zlib - это универсальный решение для сжатия данных с использованием языка программирования Python. Это может войти удобно для распространения пакетов, особенно если распространение пакетов полностью автоматизировано. В Python замечательно то, что его можно использовать как фантастический инструмент автоматизации. Тот как говорится, возможность автоматически сжимать tarball, а затем даже отправлять его в другом месте - довольно ценный актив.



В модуле zlib наиболее важными функциями, вероятно, являются compress () и распаковать (). Используя эти функции, мы можем увидеть сжатие типов данных. на наших глазах с помощью этого кода:



**Заключение**

Хотя часто в программировании на Pythonic может показаться, что зависимостей, в стандартной библиотеке Python также есть несколько действительно полезных инструментов! Много эти инструменты можно быстро использовать, чтобы сэкономить значительное количество времени, интерфейс уровня для сложных задач, который упакован с самим языком программирования. Это можно использовать для минимизации зависимых пакетов, виртуальных сред и общих беспорядок вокруг сервера. Кроме того, нет необходимости устанавливать какие-либо пакеты, поэтому много время экономится благодаря этой инициативе. Это только некоторые из моих любимых модулей, которые доступны в стандартной библиотеке, однако есть еще много всего! Хотя эти модули, безусловно, пригодятся, я конечно уверен, что есть много других доступных пакетов, которые сделают многопрограммировать операции намного проще, чем в противном случае.