Работа с виртуальными окружениями в ОС

Виртуальное окружение – это окружение, которое позволяет изолировать зависимости для определенного проекта. Например, предположим вы разрабатываете на машине два проекта, причем в этих проектах используется одна и та же библиотека, но разных версий. В одном проекте чуть более старая версия этой библиотеки, в другом чуть более новая. Виртуальное окружение позволяет вам изолировать эти две зависимости друг от друга, так что вы можете работать над двумя проектами параллельно на одном компьютере. Это очень удобно. Тем самым, обновляя библиотеку в одном проекте, вы не сломаете код другого проекта.

Чтобы создать виртуальное окружение мы воспользуемся модулем, который идет в поставке Python3 из коробки он называется **venv**. Чтобы создать виртуальное окружение мы пишем

Python3 -m-venv <virt_Dir_Name>

и дальше название директории, в которой будет создано наше виртуальное окружение. Однако перед этим создайте директорию, в которой вы будете работать и перейдите в нее. Команду

Python3 - m-venv

нужно запустить именно в этой новой директории. В этот момент создается виртуальное окружение и, если посмотреть на структуру директорий, можно увидеть, что создалось папочка venv.

Внутри директории venv создалось несколько директорий, одна из них **Bin**, другая **Include**, а также **Lib**. Директория **Bin** содержит исполняемый файл *Python3,* который на самом деле ссылается на наш глобально установленный ***Python3***. Также внутри Bin есть утилита **pip**, и другие важные утилиты, например, скрипт activate. Этот скрипт позволит активировать виртуальное окружение. Также обратите внимание на директорию **Lib**, именно в эту директорию ставятся сторонние зависимости во время работы в виртуальном окружении.

Работа в виртуальном окружении начинается с его активации. Обратите внимание, что все примеры, которые здесь показаны, одинаково работают, как на Mac OS, так и для Linux. Однако, для Windows виртуальное окружение, структура его директории будут немного отличаться. Процесс работы с виртуальным окружением специфичный для Windows, описан в следующем пункте.

Активация виртуального окружения осуществляется с помощью команды **source** на unix-подобных системах, и мы должны вызвать

source <путь к скрипту../activate>

Чтобы деактивировать виртуальное окружение, мы набираем **deactivite**.

Работать в виртуальном окружении можно только в активированном режиме. Поэтому, для установки сторонних модулей снова наберем команду активации и установим библиотеку requests. Это позволит теперь использовать для запуска интерпретатора Python, а не Python3, потому что, в данном случае мы находимся в виртуальном окружении, и Python ссылается на Python3. Также после установки библиотеки request можно использовать ее импорт.

Установим программу, работающую как веб-интерфейс для интерпретатора Python – это Jupyter-ноутбук (ноутбук содержится в пакете jupyter):

pip install jupyter

Все необходимые зависимости, а также зависимости зависимостей, пакета jupyter pip-утилита загружает с ресурса pip.org.

Хотелось бы обратить внимание на один полезный пакет, который установился вместе с зависимостями Jupiter это пакет ipython.

Ipython это расширенная версия, интерактивный интерпретатор Python. Если вы любите экспериментировать в интерактивном интерпретаторе Python, то рассмотрите iPython в качестве альтернативы, помимо подсветки синтаксиса, которая присутствует в iPython, там есть ещё много полезных особенностей, которые позволят вам упростить жизнь при написании и отладке кода. Например, это авто-дополнение, хранение истории и всевозможные полезные макросы.

Теперь мы можем попробовать всё-таки запустить наш Jupyter ноутбук, он установлен в виртуальное окружение, теперь его исполняемые файлы находятся в папочке bin виртуального окружения и командой Jupyter ноутбук мы можем его запустить. это клиент-серверное веб-приложение, которое запускается в веб-браузере, веб-браузер должен открыться автоматически, и мы внутри этого веб-приложения можем создавать наши ноутбуки.

Для создания нового документа нужно выбрать в верхнем меню (справа) New, в выпадающем списке выбираем Python3, после чего открывается ваш ноутбук. Он состоит из ячеек, в которых можно набирать питоновский код, делать любые импорты пакетов, которые уже установлены, и даже инсталлировать новые, используя специальную команду

!install <модуль>

Все изменения в ноутбуке можно сохранить, можно вернуться к ним в дальнейшем и просмотреть заново, перезапустить ячейки. После этого данный файл можно использовать в качестве инструмента для создания презентационных слайдов, сохранив его в формате pdf или html.

Виртуальное окружение на Windows

Работа с виртуальным окружением на Windows немного отличается от того, что мы видели в видео для Unix-подобных систем. Чтобы создать виртуальное окружение на Windows запустите в терминале (PowerShell):

python3 -m venv c:<\path\to\myenv>

Где вместо c:<\path\to\myenv> укажите путь до папки с виртуальным окружением, которую вы хотите создать.

После того как скрипт отработает, вы можете активировать виртуальное окружение с помощью:

c:<\path\to\myenv>\Scripts\activate.bat

Чтобы деактивировать виртуальное окружение:

c:<\path\to\myenv>\Scripts\deactivate.bat

После активации виртуального окружения вы можете устанавливать в него дополнительные сторонние библиотеки точно так же, как было это описано в предыдущем пункте, например:

pip install requests

Обратите внимание, что в то время как на Unix системах интерпретатор python находится в директории bin/ внутри виртуального окружения – на Windows интерпретатор будет находиться внутри директории Scripts/ созданного виртуального окружения.

На этом занятии мы с вами познакомились с виртуальным окружением в Python, научились ставить пакеты в нашу операционную систему, также мы установили Jupiter ноутбук, который является полезным инструментом для всех программистов на Python.