VS Code portable, делаем настоящую переносную сборку для Windows.

Я не так давно начал изучать Python, и решил, что мне необходима портативная сборка. Причин для этого несколько, но статья не об этом. Если вам такое не нужно, дальше можно не читать. Во время поисков решения этой задачи часто сталкивался с вопросами людей по этой теме, но однозначного решения так и не нашел, но ответы некоторых пользователей натолкнули на верное решение. Почему VS Code? Ну, просто у них есть версия “portable”, так она гордо называется на сайте, но скачав ее, возникает вопрос, а как к тебе приделать Python?

Сразу оговорюсь, целью было сделать полноценно переносную версию из связки Python + среда разработки + Git (для изучения). В статье все расписано, так же на github закину файлы со всеми изменениями, и вам останется только создать структуру каталогов как у меня. Если хотите все разложить по своему- читайте-исправляйте, по аналогии думаю не сложно будет сделать под себя.

Что нам потребуется:

1. Winpython- <https://winpython.github.io/>
2. Cmder full – замена командной строке <https://cmder.net/>
3. VS code portable -https://code.visualstudio.com/download качаем по ссылке .zip нужное разрядности, но для универсальности лучше 32бит

Если вам не нужен Git или Cmder их можно не добавлять, так же можно добавить portable Git отдельно, но тогда изучайте что нужно дописать и куда.

Извлекаем winpython в любое удобное место. Я ставил версию Winpython32-3.8.10.0dot, она последняя поддерживает 32 битные системы. После извлечения папку назвал ” WPy32-38”, можете обозвать как нравится, но дальше в статье я буду ее называть так.

И так заходим в нашу папку “WPy32-38”, дальше нас интересует папка “t” вот в нее мы и извлекаем Cmder и VS code. Это уже обязательное условие.

После извлечения VS Code настраиваем по инструкции с оф.сайта, так же это сработает и для миграции уже установленного VS Code. Для портативной версии достаточно внутри папки VS Code создать папку ”data” и в ней папку “user-data”. Должно получится как-то так:

|- VSCode-win32-x64-1.25.0-insider

| |- Code.exe (or code executable)

| |- data

| | |- user-data

| | | |- ...

Cmder просто извлекаем рядом с VS code. Папку с проектами я разместил в папке WPy32-38.Так проще всего прописать пути к ней, иначе виртуальное окружение не подхватывал VS code.

|- WPy32-38

| |- projekt

| |- t

| | |- VSCode

| | |- Cmder

Все с размещение файлов и папок разобрались, теперь все это надо прописать.

Первым делом открываем блокнотом файл “..\..\WPy32-38\scripts\env.bat

Там много чего расписано, можете посмотреть и разобраться самостоятельно, но нам надо добавить вот такие строки. Добавляем там же где находятся подобные блоки. Первый блок отвечает за Cmder, второй за Git. Если вы решили их не добавлять, пропускайте этот шаг. Помните я писал что расположение важно? Так вот если у вас пути к папкам свои исправляем на данном этапе. Этот код располагаем после 55 строки примерно, между другими подобными блоками

Это для установки папки с Cmder

rem \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rem handle Cmder if included

rem \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if not exist "%WINPYDIRBASE%\t\cmder\vendor\" goto cmder\_bad

set CMDER\_HOME=%WINPYDIRBASE%\t\cmder\vendor\

set CMDER\_EXE=init.bat

set CMDER=%CMDER\_HOME%%CMDER\_EXE%

set CMDER\_PKGDIR=%WINPYDIRBASE%\cmder\

:cmder\_bad

Это для установки папки с Git

rem \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rem handle GIT if included

rem \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if not exist "%WINPYDIRBASE%\t\cmder\vendor\git-for-windows\cmd\" goto git\_bad

set GIT\_HOME=%WINPYDIRBASE%\t\cmder\vendor\git-for-windows\cmd\

set GIT\_EXE=git.exe

set GIT=%GIT\_HOME%%GIT\_EXE%

set GIT\_PKGDIR=%WINPYDIRBASE%\t\cmder\vendor\git-for-windows\

:git\_bad

Так же после строки ” if %ERRORLEVEL% NEQ 0” (у меня это 26 строка) в поле “set” дописываем пути к Cmder и Git, можете просто заменить этим:

set "PATH=%WINPYDIR%\Lib\site-packages\PyQt5;%WINPYDIR%\Lib\site-packages\PySide2;%WINPYDIR%\;%WINPYDIR%\DLLs;%WINPYDIR%\Scripts;%WINPYDIR%\..\t;%WINPYDIR%\..\t\mingw32\bin;%WINPYDIR%\..\t\R\bin\i386;%WINPYDIR%\..\t\cmder\vendor;%WINPYDIR%\..\t\cmder\vendor\git-for-windows;%WINPYDIR%\..\t\cmder\vendor\git-for-windows\cmd;%WINPYDIR%\..\t\cmder\vendor\git-for-windows\bin;%WINPYDIR%\..\t\cmder\vendor\git-for-windows\mingw32\bin;%WINPYDIR%\..\t\Julia\bin;%WINPYDIR%\..\n;%PATH%;"

На данном этапе все настройки для WinPython мы сделали.

Переходим к настройке VS Code. Для запуска используем ..\WPy32-38\VS Code.exe

Возможно при первом запуске будут ошибки, пока их просто игнорируем и открываем настройки. ..\WPy32-38\t\VSCode\data\user-data\User\ settings.json

// Указываем путь для питона

"python.interpreterPath": "${env:WINPYDIR}\\\\python.exe",

// Указываем путь для для проектов и соответсвенно для “venv”

"python.venvPath": "${env:WINPYDIR}\\${workspacefolder}/.venv",

// Указываем путь до Cmder

"terminal.integrated.profiles.windows": {

"Cmder": {

"path": "${env:windir}\\System32\\cmd.exe",

"args": ["/k", "${env:WINPYDIRBASE}\\t\\cmder\\vendor\\init.bat"]

},

// Тут путь для GIT, и соотвественно появится выбор командной строки GIT,можно не добавлять.

"GIT": {

"path": "${env:WINPYDIRBASE}\\t\\cmder\\vendor\\git-for-windows\\git-cmd.exe",

}

},

//Устанавливаем Cmder по умолчанию

"terminal.integrated.defaultProfile.windows": "Cmder",

//Активируем GIT

"git.enabled": true

//Прописываем путь до GIT

"git.path": "${env:WINPYDIRBASE}\\t\\cmder\\vendor\\git-for-windows\\cmd\\git.exe",

Сохраняем, перезапускаем VS Code.

Для проверки открываем командную строку “Ctrl+~” , если видите там лямбду, Cmder подцепился и работает. Пишем в командной строке git --version, получили версию, значит и git VS Code видит.

Пробуем запускать проект и проверяем venv, если venv все таки не заработал, то тут пока одно решение. Если вы знаете что виртуальная среда создавалсь в этой же версии питона, то просто вводим в терминале VS Code : python –m venv venv и перезаходим, зависимости ставить повторно не надо. Да, такой вот костыль, но другого варианта я не придумал. Так же не могу сказать как активировать виртуальную среду другой версии. Просто, по какой-то причине стандартная команда для активации venv у меня не срабатывает. Тут уже тестируйте-смотрите.