

Настройка и использование VS Code для Python-кода

В 2016-м году компания Microsoft представила миру свой новый редактор программного кода. В отличие от старшей сестры — полноценной IDE Visual Studio — VS Code получился куда более компактным и легковесным решением. Он разработан как кроссплатформенное ПО и может быть успешно установлен в системах Windows, Linux и macOS.

Visual Studio Code

Бесплатность Visual Studio Code абсолютно не мешает ему обладать весьма богатым современным функционалом. VS Code имеет встроенный отладчик, позволяет работать с системами контроля версий, обеспечивает интеллектуальную подсветку синтаксиса, а также поддерживает целый ряд популярных языков программирования.

И хоть, за годы своего существования, VSCode зарекомендовал себя, в основном, как продукт для веб-разработки, в 2018 году появилось расширение "**Python**", которое дало программистам многочисленные возможности для редактирования, отладки и тестирования кода на нашем любимом языке.

`__main__.py`

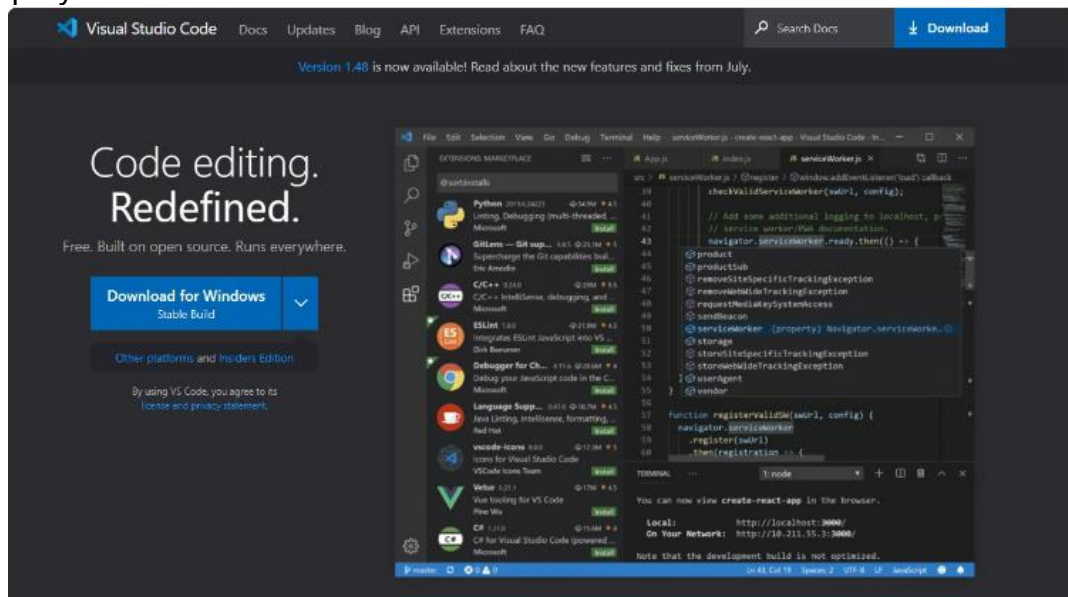
Установка VSCode

Поистине смешные системные требования Visual Studio Code обязательно порадуют владельцев старых машин.

Для полноценной работы редактору требуется всего лишь 1 ГБ оперативной памяти и процессор с частотой от 1.6 ГГц.

Windows

Сначала нужно скачать с [официального сайта](https://code.visualstudio.com/) установочный файл небольшого размера, а затем установить сам ВиЭс код, следуя подсказкам от мастера установки.



VS Code можно скачать с официального сайта code.visualstudio.com

Linux

На сайте программы можно изучить способы инсталляции редактора на разные Linux-дистрибутивы, но здесь рассмотрим процесс установки для самого популярного из них – Ubuntu.

Установить VSCode можно несколькими способами:

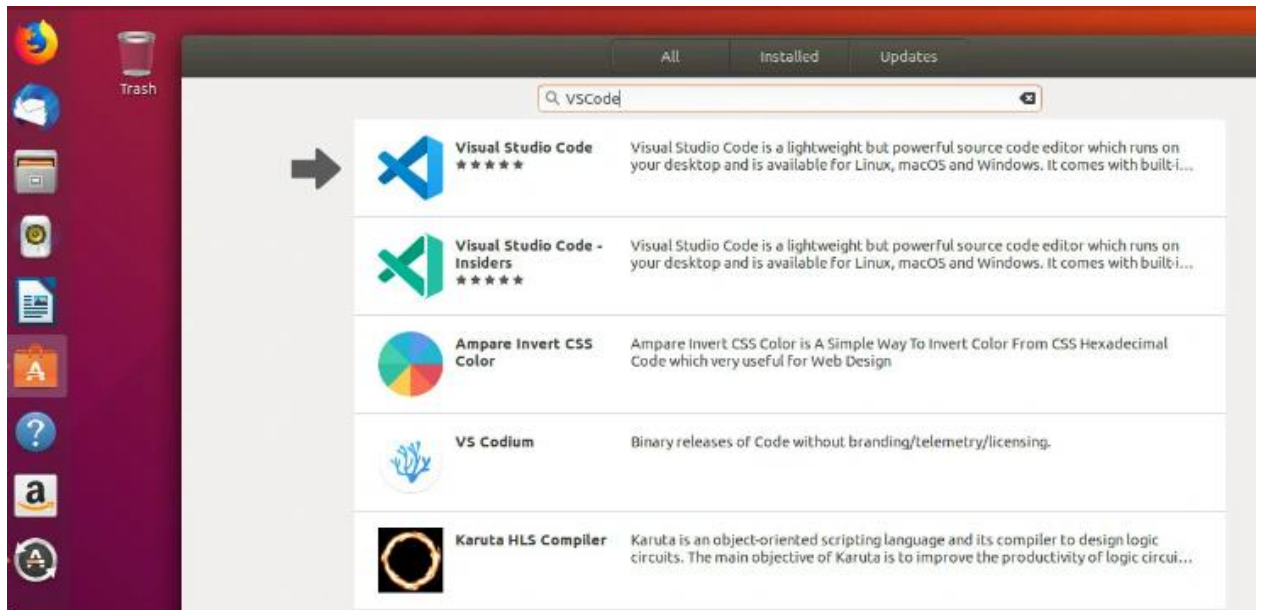
Способ №1: Самый простой способ – воспользоваться менеджером установки "Ubuntu Software".

Откройте "Ubuntu Software" -> введите в поиск "VSCode" -> выберите первую строку и нажмите "Install"

Способ №2: Скачать дистрибутив с [официального сайта](https://code.visualstudio.com/) и установить его командой:

```
sudo apt install ./<file>.deb
```

О других способах установки читайте на странице официальной документации в разделе "**Setup**";



macOS

Алгоритм установки редактора внутри яблочной операционной системы также не представляет собой ничего сложного:

1. Сначала нужно [скачать](#) Visual Studio Code с официального сайта.
2. Затем открыть список загрузок браузера и найти там VSCode-Darwin-Stable.zip.
3. Нажмите на иконку увеличительного стекла, чтобы открыть архив.
4. Перетащите Visual Studio Code.app в папку приложений, сделав ее доступной на панели запуска.
5. Щёлкните правой кнопкой мыши по значку и выберите команду **"Оставить в Dock"**.

Настройка под Python

Установка расширения "Python"

Для начала работы с Python, нужно перейти на вкладку `Extensions`, что находится на панели слева, либо нажать `Ctrl + Shift + X`. Сделав это, набираем в строке поиска "Python".



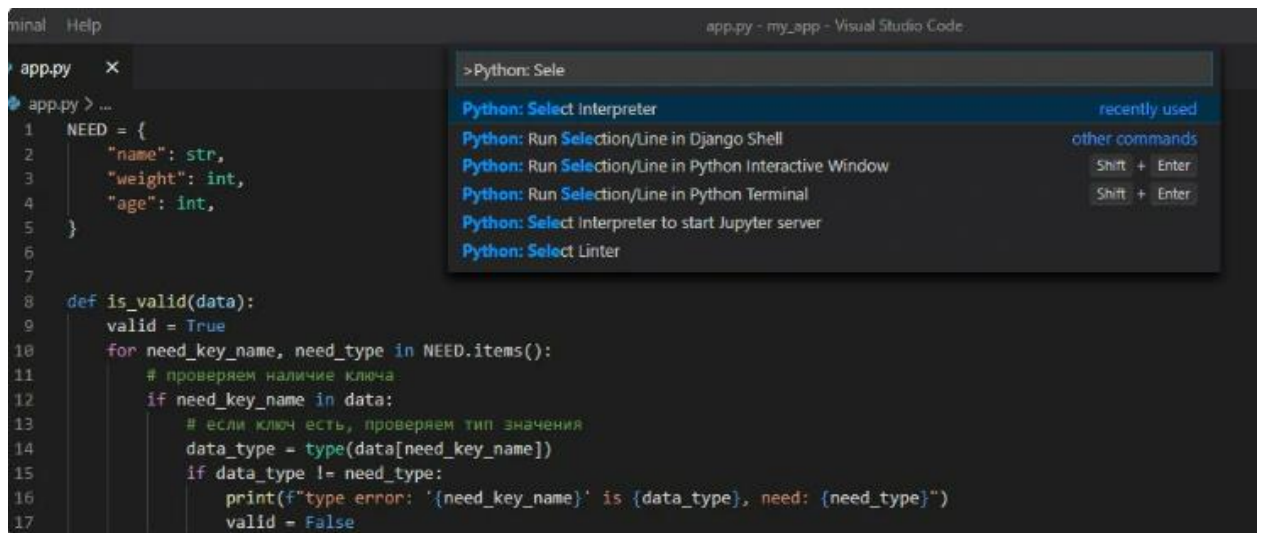
Для начала работы с Python, установите расширение от Microsoft — "Python".

VS Code поддерживает, как вторую, так и третью версию языка, однако python интерпретатор на свою машину вам придётся поставить самостоятельно.

Выбор версии интерпретатора Python

После от вас потребуется совершить выбор версии интерпретатора внутри самого редактора (обычно VS code знает, где он расположен). Для этого:

1. Откройте командную строку VSCode (Command Palette) комбинацией `Ctrl + Shift + P`.
2. Начинайте печатать "Python: Select Interpreter";
3. После, выберите нужную версию интерпретатора.



Выбор интерпретатора в VSCode

Также выбрать версию можно в панели активности в левом нижнем углу:



Выбор версии Python-интерпретатора в панели активности VS Code

Если вы хотите использовать `pipenv` в своем проекте:

1. Установите `pipenv` командой `pip install pipenv` (или `pip3 install pipenv`);
2. Выполните команду `pipenv install`;
3. Откройте "Command Palette", напечатайте "Python: Select Interpreter" и из списка выберите нужную версию интерпретатор.

Работа в VS Code

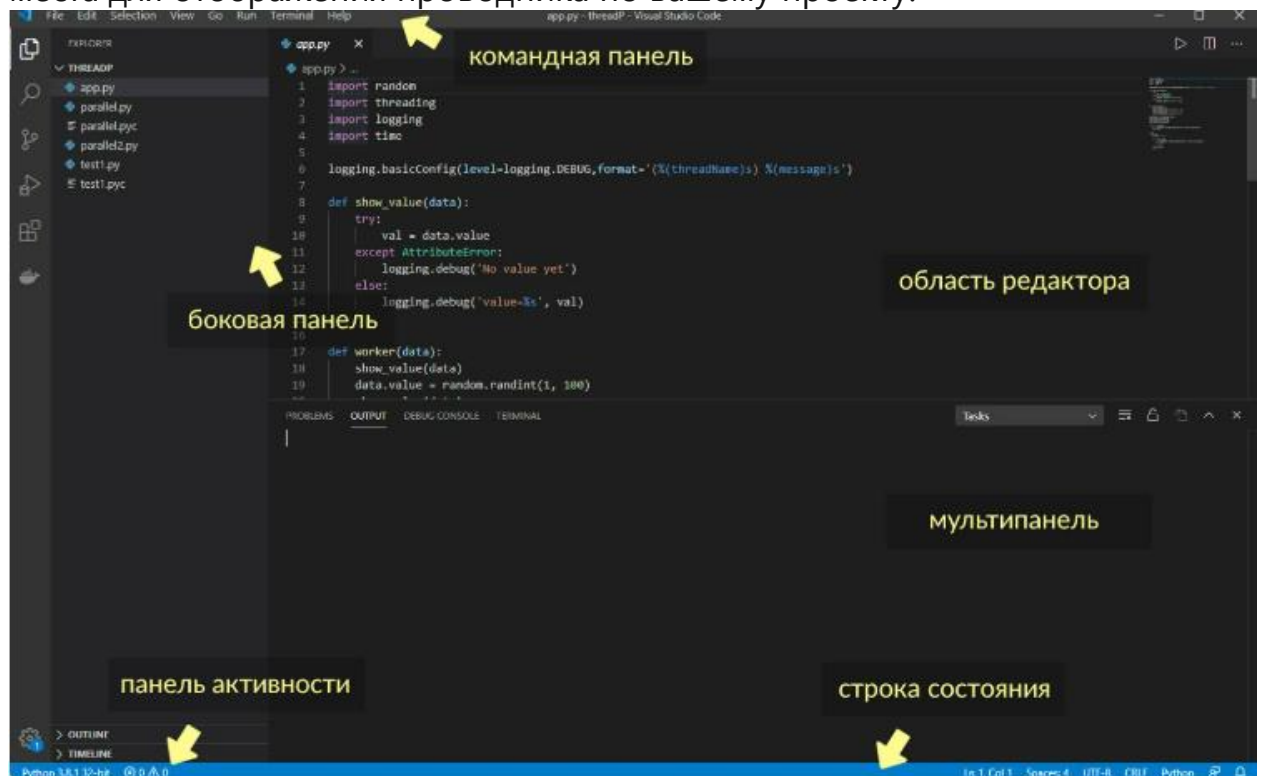
Запуск редактора

Как и другие современные редакторы и среды разработки, VS Code фиксирует состояние на момент закрытия программы. При следующем запуске, он открывается в том же самом виде, в котором существовал до завершения работы.

Так как VSCode, в первую очередь — редактор, а не полновесная среда разработки, здесь нет особой привязки к проекту. Вы можете сходу создавать, открывать и редактировать нужные вам файлы. Достаточно, после запуска, нажать **Open File** или **New File** и можно начинать работу.

Интерфейс

Интерфейс программы разрабатывался в стремлении сделать его как можно более простым и интуитивно понятным. Дизайнеры постарались, как максимизировать пространство для редактора, так и оставить достаточно места для отображения проводника по вашему проекту.



Так выглядит интерфейс VS Code

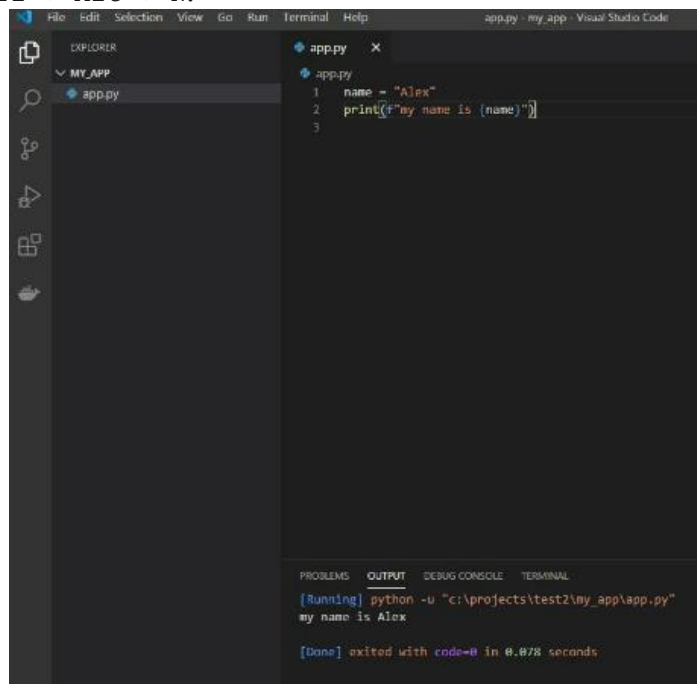
Весь UI VSCode разделился, таким образом, на шесть областей:

1. **Область редактора** — основная область для написания и редактирования вашего кода.

2. **Боковая панель** — здесь содержатся различные представления (например проводник).
3. **Строка состояния** — визуализирует рабочую информацию об открытом в данный момент файле.
4. **Командная панель** — классическая главная панель с вкладками file, edit, go, run и так далее.
5. **Панель активности** — область в крайнем левом углу, где находятся важные вспомогательные вкладки, вроде контроля версий, дебаггера и магазина расширений.
6. **Мультипанель** — панель на которой располагается вывод отладку, информация об ошибках и предупреждениях, а также встроенный в VS Code терминал.

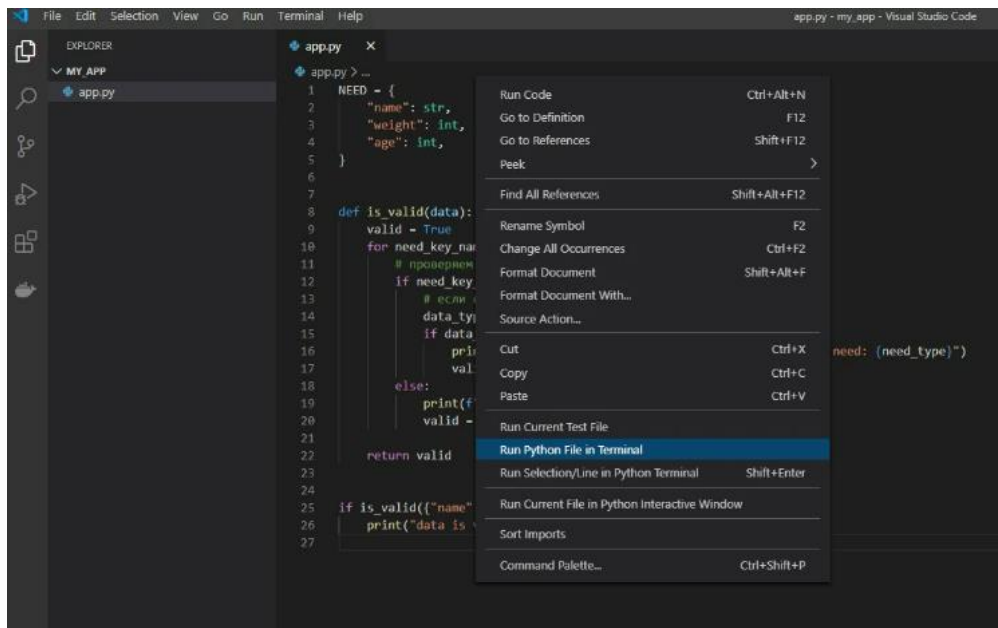
Запуск Python-кода (run)

Выполнить код можно несколькими способами. Самый простой — комбинацией `Ctrl + Alt + N`.



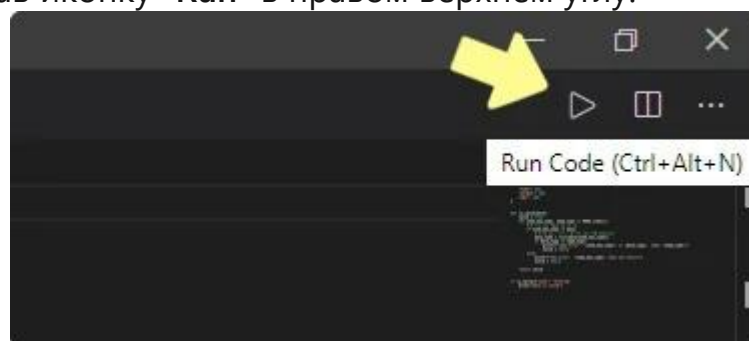
Для запуска python-кода выполните комбинацию "Ctrl + Alt + N"

Также можно вызвать скрипт контекстным меню, выбрав строку "Run Python File in Terminal".



Запуск Python-кода через контекстное меню в VSCode

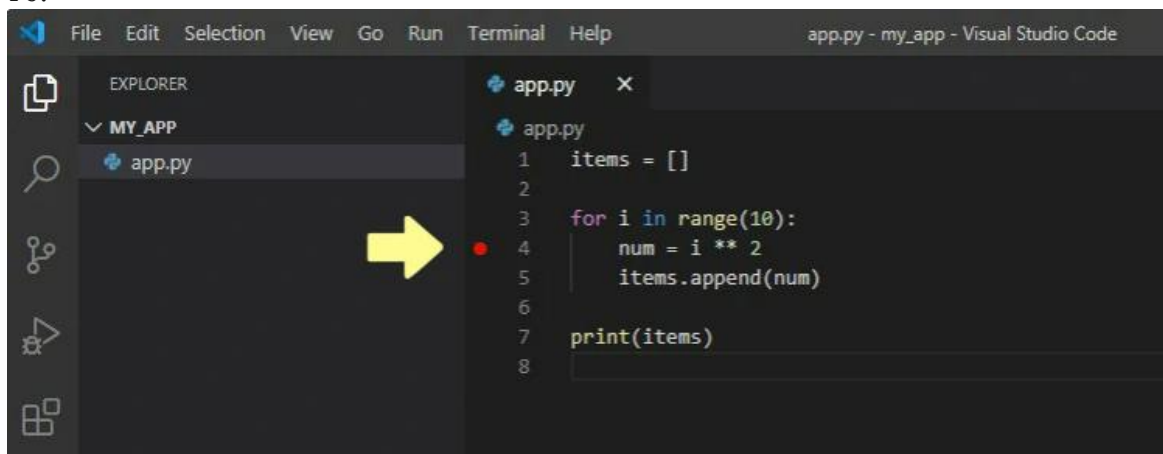
Или нажав иконку "Run" в правом верхнем углу.



Иконка "Run" в правом верхнем углу запустит Python-код на выполнение

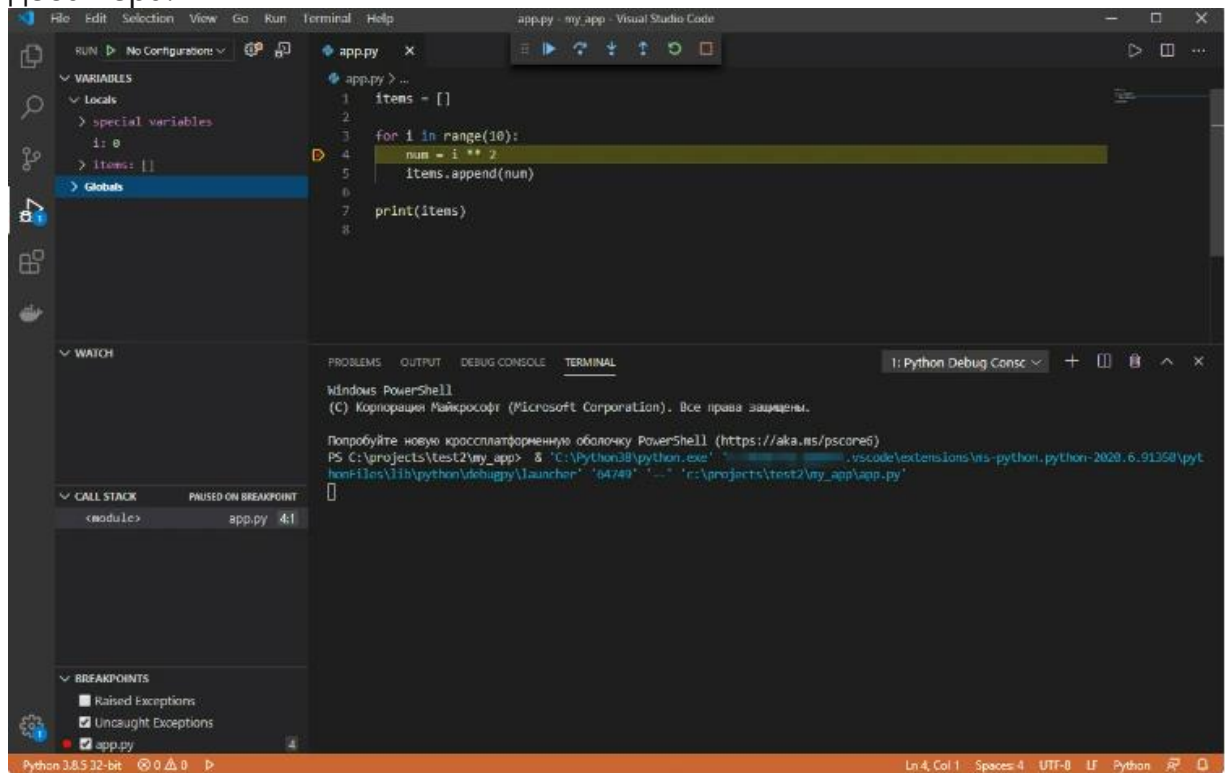
Отладка (debugger)

Возможность полноценной отладки — сильная сторона редактора. Чтобы перейти в режим отладки, нужно установить точку останова и нажать F5.



Для перехода в режим отладки, установите breakpoint и нажмите "F5"

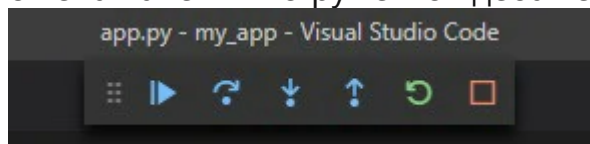
Вся информация о текущем состоянии будет выводиться на панель дебаггера.



Так выглядит debugger в VSCode

Слева откроется панель дебаггера с информацией о состоянии переменных (Variables), отслеживаемых переменных (Watch) и стеке вызова (Call stack).

Сверху расположена панель инструментов дебаггера.



Панель инструментов дебаггера в VSCode

Рассмотрим команды (слева направо):

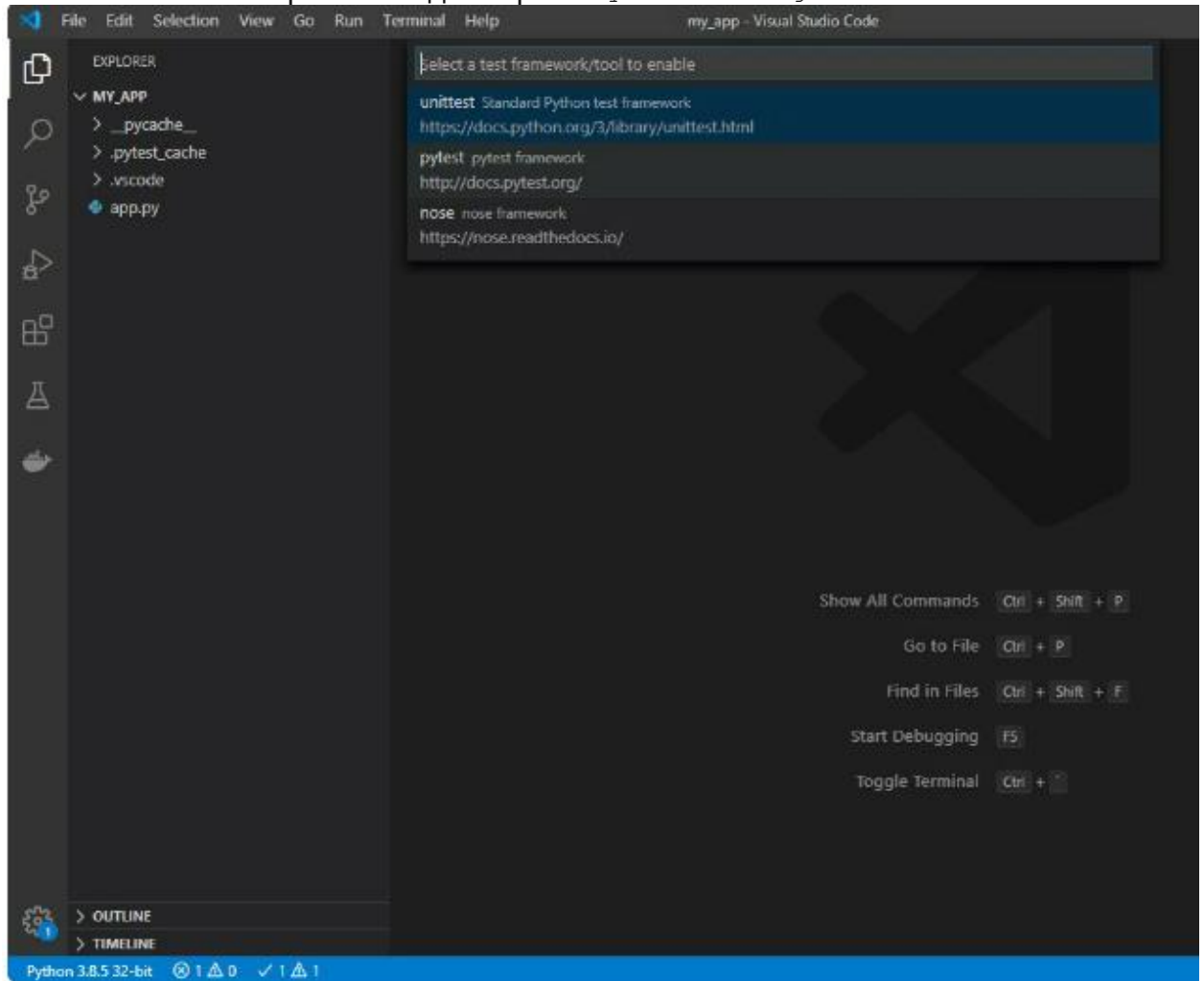
1. **continue** (F5) — перемещает между breakpoint-ами;
2. **step over** (F10) — построчное (пошаговое) перемещение;
3. **step into** (F11) — построчное (пошаговое) перемещение с заходом в каждую вызываемую функцию;
4. **step out** (Shift + F11) — работает противоположно step into — выходит из вызванной функции, если в данный момент вы находитесь внутри неё. Далее работает как continue.
5. **restart** (Ctrl + Shift + F5) — начинаем отладку с начала.

6. **stop** (Shift + F5) — остановка и выход из режима отладки.

Тестирование (testing)

С поддержкой тестов у VS Code тоже всё в порядке, однако, по умолчанию тестирование отключено. Для его активации нужна небольшая настройка.

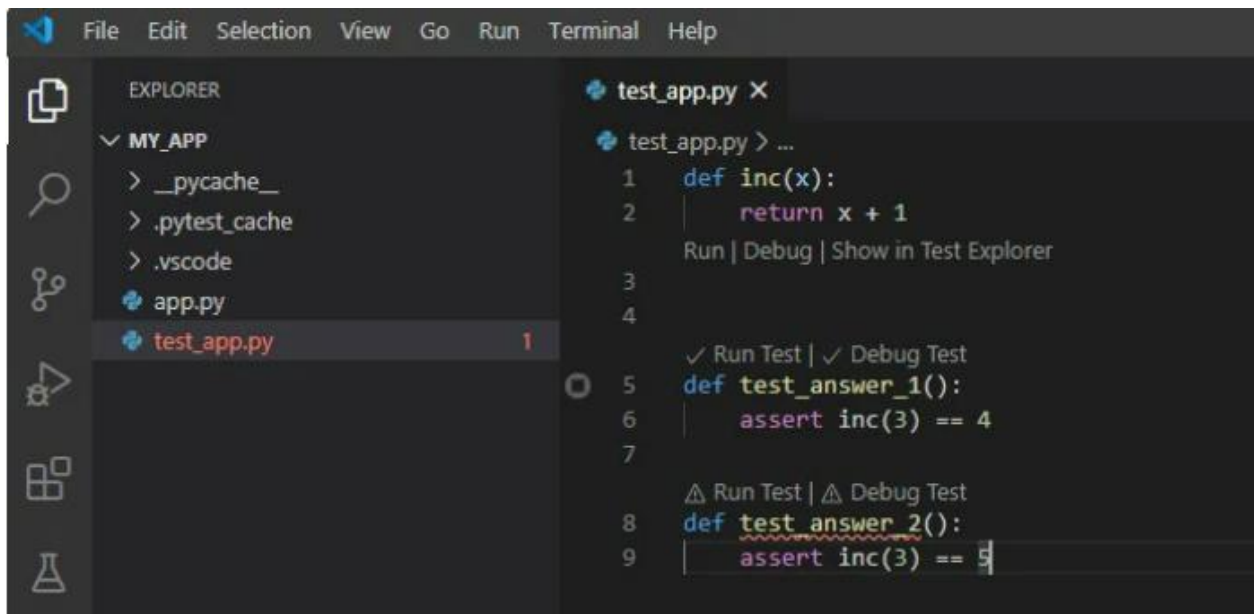
Сначала следует нажать комбинацию клавиш Ctrl + Shift + P и в так называемой палитре команд выбрать Python: Configure Tests.



Для выбора фреймворка для тестов, выполните комбинацию "Ctrl + Shift + P" и наберите "Python: Configure Tests"

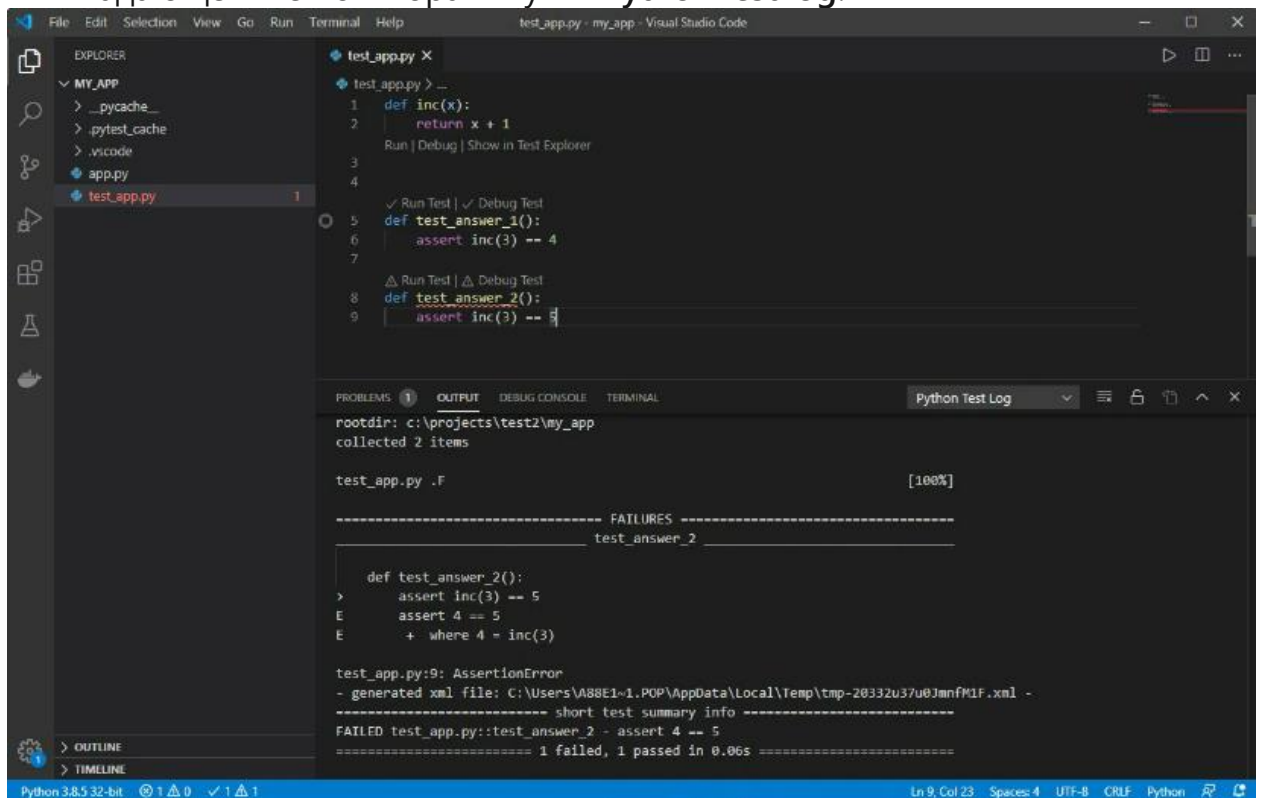
Редактор предложит вам определить фреймворк (мы выбрали "pytest") и папку, содержащую тесты (мы выбрали ". Root directory"). Создадим новый файл с тестами (**test_app.py**) и запустим его, кликнув правой кнопкой мыши на этом файле и выбрав пункт **"Run Current Test File"**.

Также тесты можно запускать по нажатию на кнопку **Run Tests** в нижней строке состояния



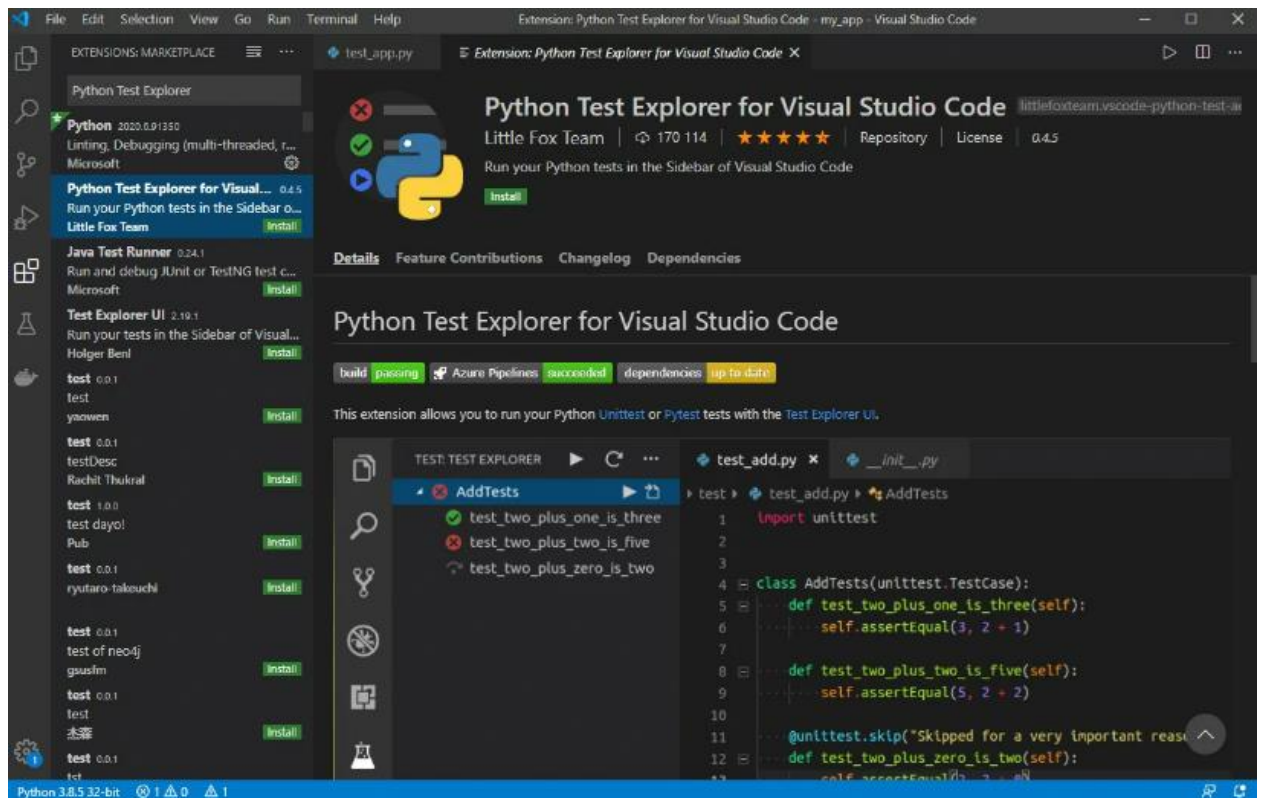
Запуск тестов в VSCode с использованием фреймворка pytest.

Чтобы увидеть результаты, необходимо открыть вкладку **Output** на панели, и в выпадающем меню выбрать пункт **PythonTestLog**.



Результат Python-тестов в Output

Как мы видим, первый тест прошел успешно, а второй провалился. Для удобства работы с тестами, установим расширение "Python Test Explorer for Visual Studio Code".



Расширение для VSCode "Python Test Explorer for Visual Studio Code"

Теперь информацию по тестам можно посмотрев, кликнув на левой панели "иконку с колбой", предварительно запустив тесты.



Вкладка "test" (иконка с колбой в панели слева) откроет удобный проводник для запуска тестов и просмотра их состояния.

Git Graph — VS Code расширение для удобного просмотра графа состояния (git log)

ТОП плагинов для VS Code

- **Python** extension for Visual Studio Code — официальное расширение Python от Microsoft.
- **TabNine** — магией машинного обучения пропитан сей плагин сотворённый для высших целей интеллектуального автозаполнения кода.
- **Python Preview** — расширение, способствующее более простой и лёгкой отладке в VSCode.
- **Indent-Rainbow** — плагин выделяет цветом все отступы, чередуя четыре разных цвета на каждом блоке.
- **Bracket Pair Colorizer** — плагин для лучшей читаемости кода, который разукрашивает соответствующие друг другу скобки в один цвет.
- **Better Comments** — позволяет создавать более красивые и удобные комментарии к вашему программному коду.
- **Python Test Explorer** for Visual Studio Code — запуск тестов с помощью TestExplorer UI.
- **Debugger for Chrome** — добавление отладчика браузера Google Chrome в ваш редактор.
- **Path Intellisense** — плагин, который автозаполняет имена файлов.
- **Python Docstring Generator** — плагин для быстрого генерирования docstrings.
- **Bookmarks** — расширение позволяет создавать закладки в коде и перемещаться по ним с помощью горячих клавиш.
- **Error Lens** — удобная подсветка ошибок.
- **TODO Highlight** — подсветка TODO и FIXME в комментариях.