

Как работать в Visual Studio Code с Python: полное руководство по установке и использованию

Введение

Visual Studio Code (VS Code) — лёгкий, кроссплатформенный и расширяемый редактор, который при правильной настройке превращается в полноценную среду разработки для Python: с автодополнением, отладкой, тестированием, форматированием кода, работой с Git

Что такое VS Code?

VS Code — редактор от Microsoft с богатой экосистемой расширений. Для Python-стека вам понадобятся:

- Python (Microsoft) — ядро поддержки Python.
- Pylance — быстрое IntelliSense и типизация.
- (Опционально) Black Formatter, isort, Ruff или Pylint, GitLens

Установка VS Code и Python

1. Скачайте и установите VS Code с официального сайта.
2. Установите Python 3.8+ (если ещё не установлен).
3. Запустите VS Code → вкладка Extensions → установите: Python, Pylance, Jupyter (и рендереры).
4. Откройте Command Palette (Ctrl+Shift+P) → Python: Select Interpreter → выберите нужный интерпретатор/виртуальное окружение.

Первый проект и виртуальное окружение

5. Создайте папку проекта и откройте её в VS Code (File → Open Folder...).
6. Откройте терминал (Ctrl+`) и создайте venv:

```
bash
```

```
python -m venv .venv
```

7. Активируйте venv:

Windows:

```
.venv\Scripts\activate
```

macOS/Linux:

```
source .venv/bin/activate
```

8. В VS Code снова Python: Select Interpreter → выберите интерпретатор из .venv.
9. Создайте main.py:

```
python
```

```
def main():
```

```
    print("Hello, VS Code + Python!")
```

```
if __name__ == "__main__":
```

```
    main()
```

10. Запустите файл: кнопка Run в правом верхнем углу редактора.

Интерфейс VS Code (аналоги из PyCharm-раздела)

- Activity Bar слева: Explorer, Search, Source Control (Git), Run and Debug, Extensions.
- Editor: вкладки файлов, миникарта, индикаторы проблем.
- Panel (внизу): Terminal, Problems, Output, Debug Console.
- Status Bar: текущий Python-интерпретатор, ветка Git, форматтер, кодировка.

Работа с кодом

Быстрые действия и исправления

- Ctrl+. — контекстные «Quick Fix/Refactor...».
- Подсветка ошибок в «Problems».

Форматирование и стиль

- Включите форматирование при сохранении: Settings → «format on save».
- Рекомендуемые инструменты: Black (форматирование), isort (импорты), Ruff/Pylint (линтинг).

Полезные хоткеи

- Ctrl+P — быстрый переход к файлам.
- Ctrl+Shift+P — все команды.
- Ctrl+/ — комментирование.
- Alt+↑/↓ — перемещение строки.
- Shift+Alt+↓ — дублирование строки.
- Ctrl+D / Ctrl+Shift+L — множественное выделение.

Отладка в VS Code

Запуск отладки

11. Откройте вкладку Run and Debug.
12. Нажмите create a launch.json и выберите Python.
13. Поставьте брейкпоинты (клик по полю слева от номера строки).
14. Кнопки управления: F5 — старт/продолжить, F10 — Step Over, F11 — Step Into, Shift+F11 — Step Out.

Пример launch.json для обычного скрипта

```
json
```

```
{
```

```
"version": "0.2.0",  
"configurations": [  
  {  
    "name": "Запуск main.py",  
    "type": "python",  
    "request": "launch",  
    "program": "${workspaceFolder}/main.py",  
    "console": "integratedTerminal"  
  }  
]  
}
```

Условные брейкпоинты и Watches

- Правый клик по брейкпоинту → Condition...
- Панель WATCH для выражений.

Управление зависимостями

Через pip и requirements.txt

bash

pip install -r requirements.txt

Git и командная разработка

- Вкладка Source Control: staged/unstaged, коммиты, истории.
- Переключение веток, создание pull requests (через GitHub расширение).
- GitLens для продвинутой аналитики.
- Мерж-конфликты VS Code показывает в три-панельном режиме с подсветкой и быстрыми действиями.