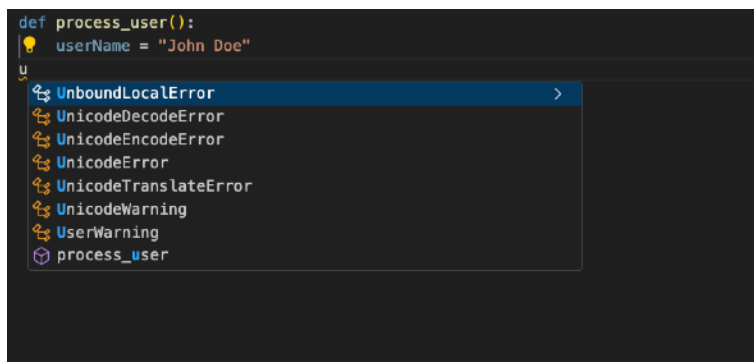
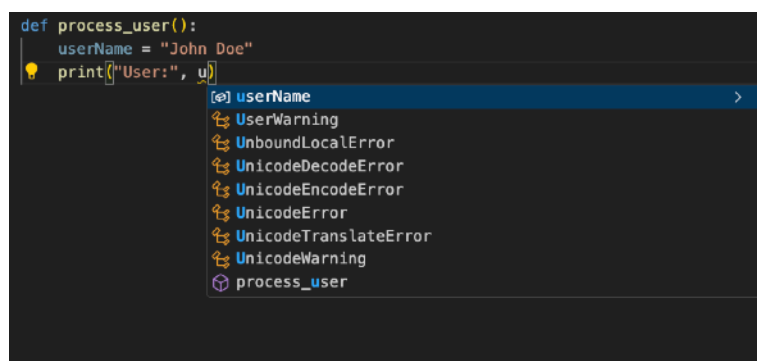


# № 1.1 (VS Code)

VS Code – не просто текстовый редактор с подсветкой синтаксиса языков программирования, а мощный и удобный инструмент для повседневного написания и отладки кода. Одна из особенностей, которая серьёзно выделяет VS Code на фоне остальных сервисов – это IntelliSense. IntelliSense представляет собой умную систему подсказок и автодополнения кода, которая работает по ходу его написания и не просто пытается завершить слова, а понимает контекст. Например, если объявить переменную `userName` и начать вводить первые буквы, то среди прочего VS Code предложит и `userName`.

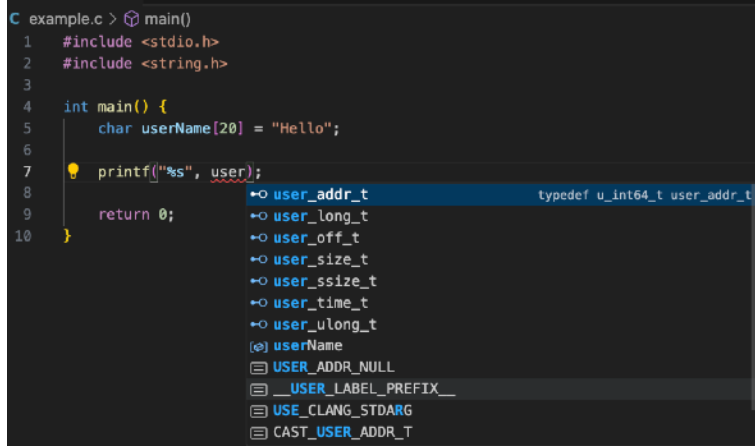


Однако в случае из первого скриншота, вариант ввести `userName` не предложен – это наглядная демонстрация того, что IntelliSense понимает контекст – в данном случае сразу после объявления такой переменной разработчик не будет начинать с неё ввод новой строки. Но как только он переходит к месту, где эта переменная становится нужна, IntelliSense начинает корректно предлагать `userName` в списке.



Правда при работе с языками C/C++ (даже при установленных расширениях) IntelliSense даёт стандартные подсказки, зачастую не вникая в контекст даже правильного кода.

```
C example.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char userName[20] = "Hello";
6
7      printf("%s", user);
8
9      return 0;
10 }
```

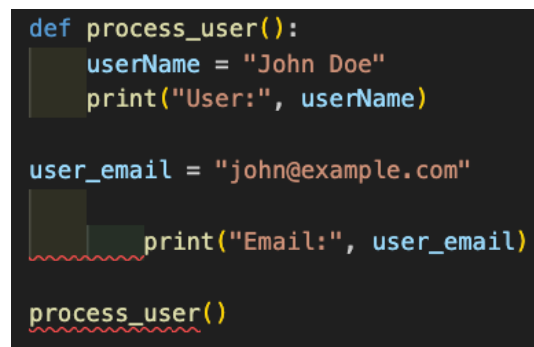


Также VS Code позволяет находить ошибки в коде, например такие как неправильные отступы в Python. Для этого необходимо установить одно из специальных дополнений: Autopep8, Black или Python Extension Pack.

```
def process_user():
    userName = "John Doe"
    print("User:", userName)

    user_email = "john@example.com"
    print("Email:", user_email)

process_user()
```

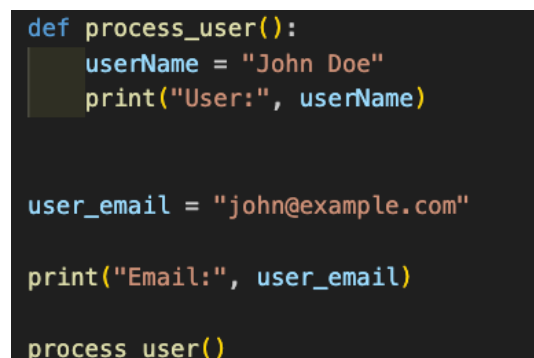


А по сочетанию клавиш Cmd+Shift+P можно автоматически исправить подобные недочёты:

```
def process_user():
    userName = "John Doe"
    print("User:", userName)

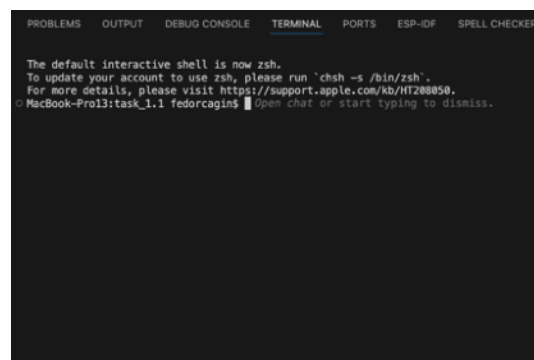
    user_email = "john@example.com"
    print("Email:", user_email)

process_user()
```

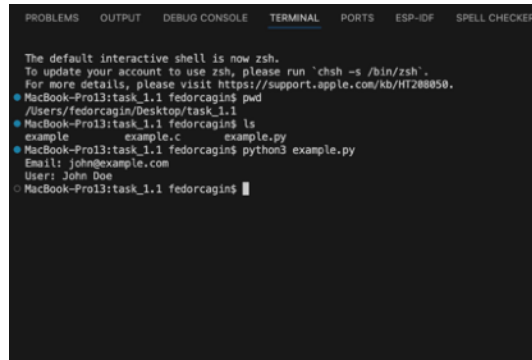


После форматирования, как видно на скриншоте, код приобрел правильные отступы и структуру.

Также в VS Code есть интегрированный терминал, доступный по сочетанию клавиш Ctrl + `



Он позволяет выполнять команды ОС непосредственно в среде разработки. На примере видно, как без переключения окон можно проверить содержимое папки (ls) и запустить программу (python3 example.py), сразу увидев результат её работы.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF SPELL CHECKER

The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT20050.
MacBook-Pro13:task_1.1 fedorcapin$ pwd
/Users/fedorcapin/Desktop/task_1.1
MacBook-Pro13:task_1.1 fedorcapin$ ls
example      example.c    example.py
MacBook-Pro13:task_1.1 fedorcapin$ python3 example.py
Email: john@example.com
User: John Doe
MacBook-Pro13:task_1.1 fedorcapin$
```

К тому же, такое приложение как VS Code может похвастаться огромной библиотекой доступных дополнений: магазин содержит огромную библиотеку плагинов для различных языков программирования, фреймворков и инструментов. Это позволяет адаптировать редактор под конкретные нужды проекта, добавляя поддержку синтаксиса, линтеры, отладчики и средства тестирования, что превращает его из простого редактора в мощную среду разработки.

Именно эти и многие другие особенности делают из VS Code не просто текстовый редактор с подсветкой синтаксиса кода, а мощным инструментом для программирования.