

Шпаргалка по дебаггингу

Какие бывают ошибки

Принято выделять три вида ошибок:

- синтаксические,
- логические,
- исключения.

Синтаксические ошибки программисту искать не нужно. Такие ошибки интерпретатор находит и обрабатывает сам. Если в программе есть подобные ошибки, программа не запустится, и Python выведет в терминал соответствующее сообщение.

Логические ошибки интерпретатор не умеет отслеживать. Программисту придётся отыскивать их самостоятельно. Логические ошибки появляются, когда разработчик неправильно описывает алгоритм работы программы.

Исключения (от англ. Exceptions) — это вид ошибок, какой появляется в тех случаях, когда в ходе работы программы возникает ситуация, которую разработчик не описал в коде. То есть программа запускается, но не выполняется целиком.

The Python Debugger в терминале

В Python «из коробки» встроена интерактивная среда отладки программ — The Python Debugger (pdb).

У pdb есть [свой раздел документации](#).

Запуск отладчика в терминале:

```
# На Windows:
python -m pdb main.py

# На Linux/macOS:
python3 -m pdb main.py
```

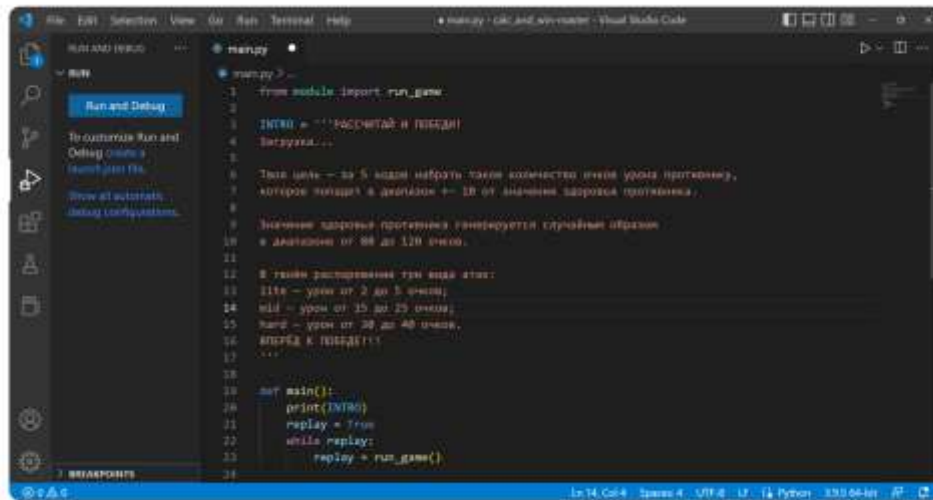
Основные команды для управления ходом выполнения программы:

- **h (help)** — вывести все доступные команды pdb.
- **l (list)** — вывести в терминал по пять строк до и после строки, на которую интерпретатор указывает в данный момент.
- **s (step)** — выполнить строку, на которую указывает интерпретатор в данный момент, и остановить выполнение программы на следующей строке, если это возможно.
- **n (next)** — эта команда схожа на команду **step**, но она будет выполнять программу не по шагам. Команда **next** не станет заходить внутрь функций, а выполнит их целиком, в один шаг.
- **b (break) имя_файла:номер_строки** — указать место обязательной остановки выполнения программы. Место обязательной остановки программы разработчики называют **«точка останова»** или **«брейкпоинт»**.

- **c (continue)** — выполнять программу до тех пор, пока она не отработает полностью или пока её выполнение не будет прервано ошибкой. Также выполнение этой команды будет прервано достижением точки останова — брейкпоинта.
- **p имя_переменной** — вывести текущее значение переменной в терминал.
- **q (quit)** — остановить работу отладчика.

The Python Debugger через интерфейс VSCode

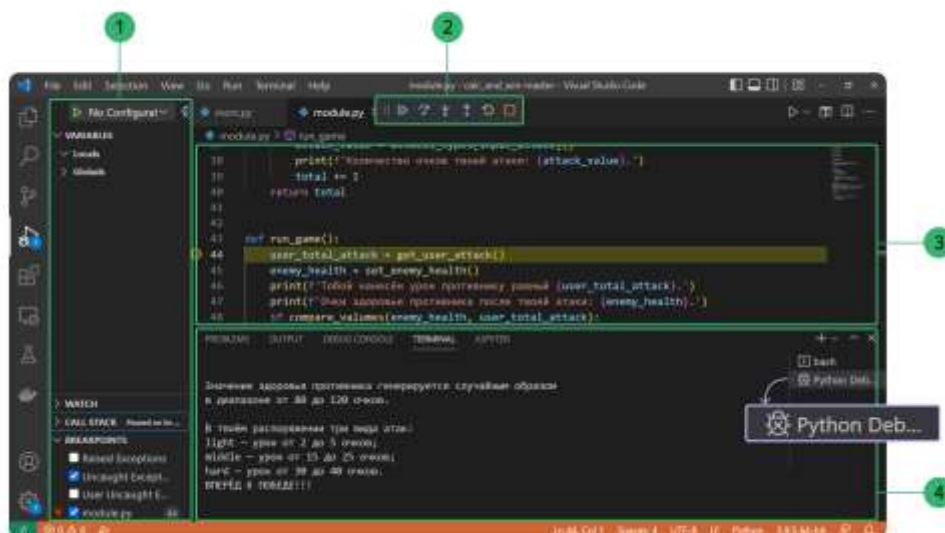
Чтобы запустить отладчик в VSCode, в левой боковой панели выберите пункт Run and Debug:



Чтобы запустить отладку:

1. Нажмите кнопку **Run and Debug**.
2. Выберите конфигурацию отладки, например «Файл Python».

Окно VSCode в режиме отладки делится на четыре части:



1. Боковая Debug-панель с информационным меню.
2. Debug Toolbar — панель с кнопками управления процессом отладки:
 - continue (F5) — **c (continue)**;
 - step over (F10) — **n (next)**;

- step into (F11) — **s (step)**;
- step out (Shift+F11) — **r (return)**;
- restart (Shift+Ctrl+F5) — перезапуск отладчика;
- stop (Shift+F5) — остановка отладки.

3. Рабочая область с кодом. Здесь появился указатель, который останавливается на строке кода, на которой Python находится в данный момент.

4. Терминал, запущенный в режиме отладки.

Чтобы поставить точку останова в VSCode, нужно кликнуть рядом с номером строки, на которой вы хотите остановить выполнение программы, — брейкпоинт обозначится красной точкой.