

Python, отладка

@pvavilin

2 апреля 2022 г.

Кен Томпсон

Томпсон



Wiki

print

Сейбел: Какие инструменты вы используете при отладке?

Томпсон: В основном вывод на печать. При разработке программы я размещаю очень много операторов вывода на печать.

Чуть лучше чем принт

logging

- Во время разработки можно выводить сообщения на экран
- В продакшене, пишем сообщения уже в логгер (syslog, просто в файл)
- Можем настраивать режимы отладки

PDB

myobj.py

```
from dataclasses import dataclass
@dataclass
class MyObj:
    n: int
    def go(self):
        for i in range(self.n):
            print(i)
if __name__ == "__main__":
    MyObj(5).go()
```

```
python -m pdb ./myobj.py
```

PDB

Можно вызвать в рамках Python-сессии

```
import pdb  
from myobj import MyObj  
pdb.run("MyObj(5).go()")
```

PDB

Чаще всего может потребоваться перейти в режим отладки в определённом месте программы

```
# тут какой-то код  
def main(request):  
    import pdb # ipdb  
    pdb.set_trace() # ipdb.set_trace()  
    data = request.GET  
# дальше много кода
```

PDB

```
try:
    some_code()
except:
    import sys
    import ipdb
    tb = sys.exc_info()[2]
    ipdb.post_mortem(tb)
```


PDB

`h(elp) [command]` выведет подсказку по доступным командам

`a(rgs)` напечатать аргументы текущей функции (текущего стека вызовов)

`p <expression>` напечатать содержимое переменной или результат вызова

`pp <expression>` красиво напечатать

`w(here)` покажет стектрейс как вы очутились в текущем месте кода

`l(ist) [start [end]]` выведет по 5 строк вокруг текущей или вокруг заданной, либо выведет строки со `start` по `end`

`u(p) / d(own)` перемещение по стеку вызовов

PDB

`!` выражение выполнить выражение, например
переопределить значение переменной

`s(step)` зайти внутрь вызова

`n(ext)` следующая строка кода

`b(reak) [([filename:]lineno | function) [, condition]]` добавить
точку останова

`t(break) [([filename:]lineno | function) [, condition]]` одноразовая
точка останова

`condition bpnumber [condition]` добавить условие для точки
останова

`disable / enable tbnumber` выкл / вкл точку останова

`cl(ear) (filename:lineno | bpnumber)` удалить точку останова

Дополнительное чтение

- import pdb
- pdb — интерактивный отладчик
- python debugging with pdb

PuDB

```
|python -m pudb myobj.py
```

```
PuDB 2019.2 - ?help n:next s:step into b:breakpoint !:python command line
```

```
1 # -*- coding: utf-8 -*-
2 from dataclasses import dataclass
3 @dataclass
4 class A:
5     def __init__(self):
6         for i in range(1000000):
7             pass
8     if __name__ == "__main__":
9         obj = A()
10         obj.foo()
```

Variables:

Stack:

Breakpoints:

myobj.py:7 (B Nite)

Command line: [Ctrl-X] Clear

Visual Studio Code / PyCharm

- VSC
- PyCharm

runserver_plus

installation instruction

```
| WERKZEUG_DEBUG_PIN=1234 \  
python manage.py runserver_plus
```