

PDB Cheatsheet

一图流速查表 | Generated from pdb.md

如何开始调试

插入代码启动调试

```
import pdb; pdb.set_trace()
```

启动默认调试器pdb 不需要
import pdb

```
breakpoint()
```

如不想修改代码 可修改运行指令

```
python -m pdb script.py
```

lineno是什么

lineno (行号) 指源代码文件中的行号 (从 1 开始)
在命令中使用时 它指定文件中的特定行

在当前文件的第 10 行设置断点

```
b 10
```

执行到当前函数的第 15 行

```
until 15
```

执行控制

执行当前行 进入被调用函数

```
s(tep)
```

执行到当前函数的下一行 不进入函数内部

```
n(ext)
```

继续运行直到断点

```
c(ont(inue))
```

执行到当前函数返回

```
return)
```

执行到指定行号或当前行之后
若无参数, 执行完当前行 (含
循环)

指定行号指的是当前函数中的
行号

```
unt(il) [lineno]
```

保留断点重启程序

重启时不会重新编译、加载模
块 不会检测代码变更

```
run [args] / restart  
[args]
```

断点管理

设置断点 支持条件

无参数则会列出所有当前断点的
详细信息

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

myfile.py 的第 20 行设置断点

```
b myfile.py:20
```

mymod/utils.py 的 add 函数入
口设置断点

```
b mymod.utils.add
```

在第 10 行设置断点 当 x > 5
时触发

```
b 10 x > 5
```

在 add 函数入口设置断点 当 x
== 10 时触发

```
b add x == 10
```

在 myfile.py 第 10 行设置条件
断点 (x > 5)

```
b myfile.py:10 x > 5
```

临时断点, 命中一次后删除

```
tbreak [([filename:]lineno | function) [, condition]]
```

清除断点

无参数则会清除调试会话中设
置的所有断点

不会修改或删除源代码中的
breakpoint()

```
cl(ear) [filename:lineno | bpnumber ...]
```

禁用断点 (保留但不触发)

```
disable bpnumber ...
```

启用断点

```
enable bpnumber ...
```

设置断点忽略次数

```
ignore bpnumber [count]
```

为断点添加/删除条件

```
condition bpnumber [condition]
```

检查和显示

显示堆栈 (箭头>表示当前帧)

```
w(here) [count]
```

在堆栈中上移动帧

```
u(p) [count]
```

在堆栈中下移动帧

```
d(own) [count]
```

列出源码 (当前行周围 11 行
默认延续上次)

```
l(ist) [first[, last]]
```

列出当前函数全部源码

```
l1 | longlist
```

显示当前函数参数

```
a(rgs)
```

打印表达式值

```
p expression
```

美观打印表达式

```
pp expression
```

显示表达式类型

```
whatis expression
```

例: <class 'int'> / <class
'list'>

```
whatis x / [1, 2]
```

显示源代码 (函数/类)。无法
获取源码 (如内置函数) 会报
错

```
source expression
```

每次停止时显示表达式 (变化
时提示)

```
display [expression]
```

停止显示表达式

```
undisplay [expression]
```

显示当前函数返回值。需先执
行 return 到函数结束, 否则
报错

```
retval
```

其他功能

执行任意 Python 语句

```
! statement
```

显示帮助 (help pdb 显示完整
文档)

```
h(elp) [command]
```

退出调试器 (程序中止)

```
q(uit)
```

注意事项

PDB 支持 Tab 补全 快捷变量
.pdbrc 文件配置别名

检查当前模块

```
p $__name__
```

检查当前文件位置

```
p $__file__
```

查看函数文档

```
p $__doc__
```

查看局部变量

```
p $__frame.f_locals
```

查看全局变量

```
p $__frame.f_globals
```

输出当前行号

```
p $__frame.f_lineno
```

查看异常

```
p $__exception__
```

查看堆栈

```
p $__traceback__
```

实用技巧

快速查看当前帧的局部变量字
典

```
p locals()
```

快速查看当前帧的全局变量字
典

```
p globals()
```

进入交互式解释器 (带当前帧
上下文), 适合临时多步探索

```
interact
```

小习惯: 先用 where/w 看堆
栈, 再用 u/d 切到目标帧, 然
后用 args/locals()/p 观察变量