```
>>> import keywords
```

---

and	or	**not**	if
elif	else	for	**while**
break	continue	def	**class**
try	except	with	**yield**

35

PY KEYS

GETTING BETTER AT PYTHON,
ONE KEYWORD AT A TIME

35 PY KEYS

Getting Better at Python, One Keyword at a Time

How to Read This Book

Python has exactly 35 reserved keywords. This book meets every one of them three times, in the same order each time, never once explaining the same thing twice.

Pass One — Meet It. A definition and the smallest honest example. Fast, shallow, on purpose — the job here is recognition, not mastery.

Pass Two — Break It. The same 35 words, same order, each one now a program that does not work. You will see the broken code, the exact failure Python produces, the diagnosis, and the fix. This is where the gotchas live, and only here.

Pass Three — Use It. Third contact. A diagram of the real mechanism, two or three named production use cases, and a one line rule for when to reach for this keyword over its neighbor.

Between Pass One and Pass Two, four short interludes connect ideas that span several keywords at once — truthiness, identity versus equality, scope resolution, when a class is the wrong tool. Four more sit between Pass Two and Pass Three, covering error-handling philosophy, generators, how `import` actually works, and the difference between concurrency and parallelism.

The spacing is the point. You meet `yield` as a definition, meet it again as a bug you diagnose, and meet it a third time as a diagram and three working examples — thirty pages apart each time. Three different questions about the same word, never one long explanation cut into thirds.

Read it start to finish, or treat any single pass as a standalone reference once you have been through it once. Every code sample in this book runs as printed, on Python 3.12 or newer, using nothing beyond the standard library.

Contents

PASS ONE — MEET IT **9**

Part I

1.01 `True` 11

1.02 `False` 12

1.03 `None` 13

Part II

1.04 `and` 15

1.05 `or` 16

1.06 `not` 17

1.07 `if` 18

1.08 `elif` 19

1.09 `else` 20

1.10 `in` 21

1.11 `is` 22

Part III

1.12 `for` 24

1.13 `while` 25

1.14 `break` 26

1.15 `continue` 27

1.16 `pass` 28

Part IV

1.17 `def` 30

1.18 `return` 31

1.19 `lambda` 32

1.20 `yield` 33

Part V

1.21 `global` 35

1.22 `nonlocal` 36

1.23 `del` 37

Part VI

1.24 `class` 39

Part VII

1.25 `try` 41

1.26 `except` 42

1.27 `finally` 43
1.28 `raise` 44
1.29 `assert` 45
1.30 `with` 46
Part VIII
1.31 `import` 48
1.32 `from` 49
1.33 `as` 50
Part IX
1.34 `async` 52
1.35 `await` 53
Interludes
Truthiness .....
`is` vs `==` .....
LEGB .....
When Not to Write a Class .....
PASS TWO — BREAK IT .....
Part I
2.01 `True` .....
2.02 `False` 67
2.03 `None` 69
Part II
2.04 `and` 72
2.05 `or` 74
2.06 `not` 76
2.07 `if` 78
2.08 `elif` 80
2.09 `else` 82
2.10 `in` 84
2.11 `is` 86
Part III
2.12 `for` 89
2.13 `while` 91
2.14 `break` 93
2.15 `continue` 95
2.16 `pass` 97
Part IV