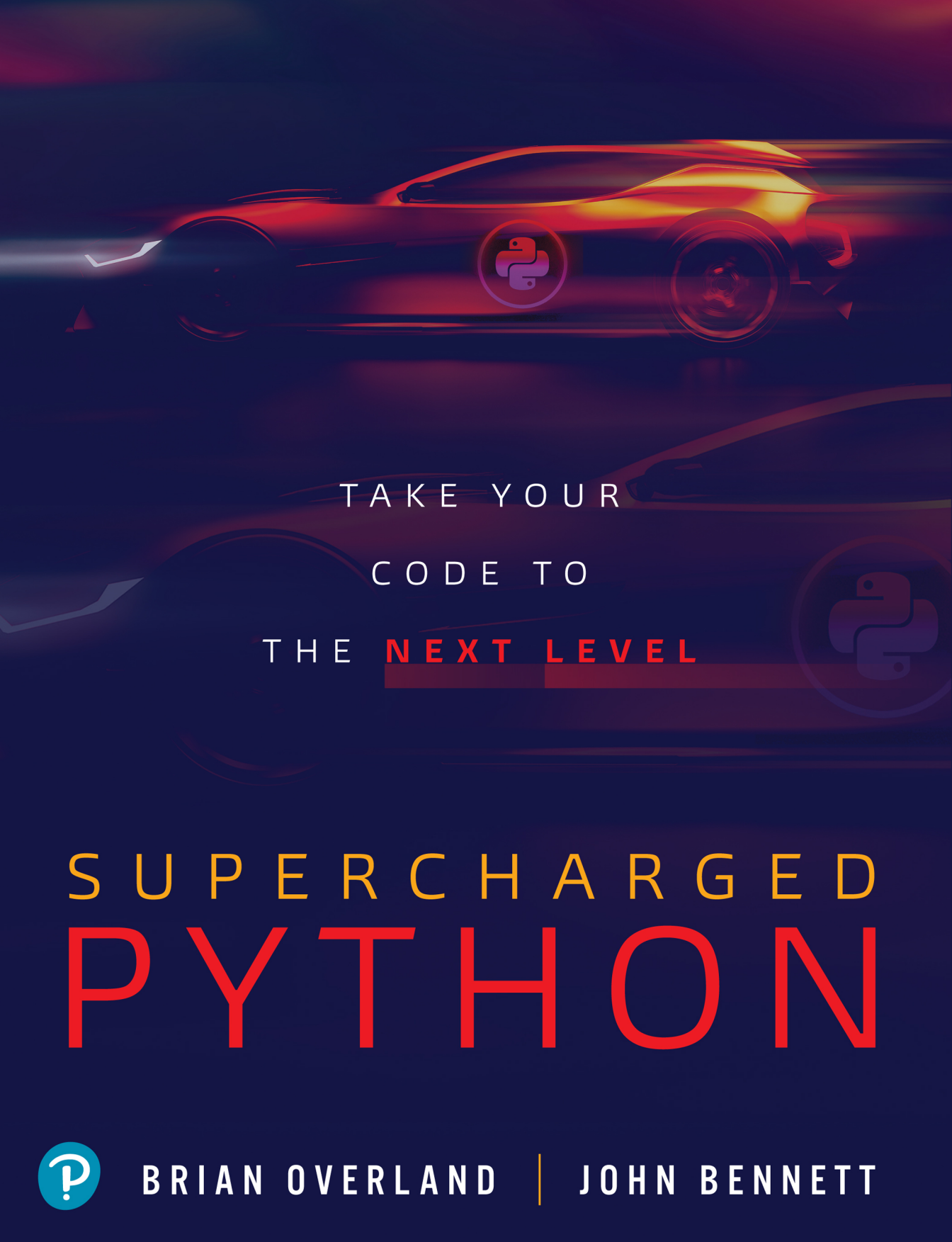




TAKE YOUR
CODE TO
THE **NEXT LEVEL**

SUPERCHARGED PYTHON



BRIAN OVERLAND

JOHN BENNETT

Supercharged Python

Supercharged Python: Take Your Code to the Next Level

Table of Contents

Cover

Title Page

Copyright Page

Contents

Preface

What Makes Python Special?

Paths to Learning: Where Do I Start?

Clarity and Examples Are Everything

Learning Aids: Icons

What Youll Learn

Have Fun

Acknowledgments

About the Authors

Chapter 1 Review of the Fundamentals

1.1 Python Quick Start

1.2 Variables and Naming Names

1.3 Combined Assignment Operators

1.4 Summary of Python Arithmetic Operators

1.5 Elementary Data Types: Integer and Floating Point

1.6 Basic Input and Output

Table of Contents

- 1.7 Function Definitions
- 1.8 The Python if Statement
- 1.9 The Python while Statement
- 1.10 A Couple of Cool Little Apps
- 1.11 Summary of Python Boolean Operators
- 1.12 Function Arguments and Return Values
- 1.13 The Forward Reference Problem
- 1.14 Python Strings
- 1.15 Python Lists (and a Cool Sorting App)
- 1.16 The for Statement and Ranges
- 1.17 Tuples
- 1.18 Dictionaries
- 1.19 Sets
- 1.20 Global and Local Variables
- Summary
- Review Questions
- Suggested Problems

Chapter 2 Advanced String Capabilities

- 2.1 Strings Are Immutable
- 2.2 Numeric Conversions, Including Binary
- 2.3 String Operators (+, =, *, >, etc.)
- 2.4 Indexing and Slicing
- 2.5 Single-Character Functions (Character Codes)
- 2.6 Building Strings Using join
- 2.7 Important String Functions
- 2.8 Binary, Hex, and Octal Conversion Functions

Table of Contents

2.9 Simple Boolean (is) Methods

2.10 Case Conversion Methods

2.11 Search-and-Replace Methods

2.12 Breaking Up Input Using split

2.13 Stripping

2.14 Justification Methods

Summary

Review Questions

Suggested Problems

Chapter 3 Advanced List Capabilities

3.1 Creating and Using Python Lists

3.2 Copying Lists Versus Copying List Variables

3.3 Indexing

3.3.1 Positive Indexes

3.3.2 Negative Indexes

3.3.3 Generating Index Numbers Using enumerate

3.4 Getting Data from Slices

3.5 Assigning into Slices

3.6 List Operators

3.7 Shallow Versus Deep Copying

3.8 List Functions

3.9 List Methods: Modifying a List

3.10 List Methods: Getting Information on Contents

3.11 List Methods: Reorganizing

3.12 Lists as Stacks: RPN Application

3.13 The reduce Function

3.14 Lambda Functions

Table of Contents

3.15 List Comprehension

3.16 Dictionary and Set Comprehension

3.17 Passing Arguments Through a List

3.18 Multidimensional Lists

3.18.1 Unbalanced Matrixes

3.18.2 Creating Arbitrarily Large Matrixes

Summary

Review Questions

Suggested Problems

Chapter 4 Shortcuts, Command Line, and Packages

4.1 Overview

4.2 Twenty-Two Programming Shortcuts

4.2.1 Use Python Line Continuation as Needed

4.2.2 Use for Loops Intelligently

4.2.3 Understand Combined Operator Assignment (+= etc.)

4.2.4 Use Multiple Assignment

4.2.5 Use Tuple Assignment

4.2.6 Use Advanced Tuple Assignment

4.2.7 Use List and String Multiplication

4.2.8 Return Multiple Values

4.2.9 Use Loops and the else Keyword

4.2.10 Take Advantage of Boolean Values and not

4.2.11 Treat Strings as Lists of Characters

4.2.12 Eliminate Characters by Using replace

4.2.13 Dont Write Unnecessary Loops

4.2.14 Use Chained Comparisons ($n < x < m$)

4.2.15 Simulate switch with a Table of Functions

4.2.16 Use the is Operator Correctly

Table of Contents

- 4.2.17 Use One-Line for Loops
- 4.2.18 Squeeze Multiple Statements onto a Line
- 4.2.19 Write One-Line if/then/else Statements
- 4.2.20 Create Enum Values with range
- 4.2.21 Reduce the Inefficiency of the print Function Within IDLE
- 4.2.22 Place Underscores Inside Large Numbers

4.3 Running Python from the Command Line

- 4.3.1 Running on a Windows-Based System
- 4.3.2 Running on a Macintosh System
- 4.3.3 Using pip or pip3 to Download Packages

4.4 Writing and Using Doc Strings

4.5 Importing Packages

4.6 A Guided Tour of Python Packages

4.7 Functions as First-Class Objects

4.8 Variable-Length Argument Lists

- 4.8.1 The *args List
- 4.8.2 The **kwargs List

4.9 Decorators and Function Profilers

4.10 Generators

- 4.10.1 Whats an Iterator?
- 4.10.2 Introducing Generators

4.11 Accessing Command-Line Arguments

Summary

Questions for Review

Suggested Problems

Chapter 5 Formatting Text Precisely

- 5.1 Formatting with the Percent Sign Operator (%)
- 5.2 Percent Sign (%) Format Specifiers

Table of Contents

5.3 Percent Sign (%) Variable-Length Print Fields

5.4 The Global format Function

5.5 Introduction to the format Method

5.6 Ordering by Position (Name or Number)

5.7 Repr Versus String Conversion

5.8 The spec Field of the format Function and Method

5.8.1 Print-Field Width

5.8.2 Text Justification: fill and align Characters

5.8.3 The sign Character

5.8.4 The Leading-Zero Character (0)

5.8.5 Thousands Place Separator

5.8.6 Controlling Precision

5.8.7 Precision Used with Strings (Truncation)

5.8.8 Type Specifiers

5.8.9 Displaying in Binary Radix

5.8.10 Displaying in Octal and Hex Radix

5.8.11 Displaying Percentages

5.8.12 Binary Radix Example

5.9 Variable-Size Fields

Summary

Review Questions

Suggested Problems

Chapter 6 Regular Expressions, Part I

6.1 Introduction to Regular Expressions

6.2 A Practical Example: Phone Numbers

6.3 Refining Matches

6.4 How Regular Expressions Work: Compiling Versus Running

6.5 Ignoring Case, and Other Function Flags

Table of Contents

6.6 Regular Expressions: Basic Syntax Summary

6.6.1 Meta Characters

6.6.2 Character Sets

6.6.3 Pattern Quantifiers

6.6.4 Backtracking, Greedy, and Non-Greedy

6.7 A Practical Regular-Expression Example

6.8 Using the Match Object

6.9 Searching a String for Patterns

6.10 Iterative Searching (findall)

6.11 The findall Method and the Grouping Problem

6.12 Searching for Repeated Patterns

6.13 Replacing Text

Summary

Review Questions

Suggested Problems

Chapter 7 Regular Expressions, Part II

7.1 Summary of Advanced RegEx Grammar

7.2 Noncapture Groups

7.2.1 The Canonical Number Example

7.2.2 Fixing the Tagging Problem

7.3 Greedy Versus Non-Greedy Matching

7.4 The Look-Ahead Feature

7.5 Checking Multiple Patterns (Look-Ahead)

7.6 Negative Look-Ahead

7.7 Named Groups

7.8 The re.split Function

7.9 The Scanner Class and the RPN Project

Table of Contents

7.10 RPN: Doing Even More with Scanner

Summary

Review Questions

Suggested Problems

Chapter 8 Text and Binary Files

8.1 Two Kinds of Files: Text and Binary

8.1.1 Text Files

8.1.2 Binary Files

8.2 Approaches to Binary Files: A Summary

8.3 The File/Directory System

8.4 Handling File-Opening Exceptions

8.5 Using the with Keyword

8.6 Summary of Read/Write Operations

8.7 Text File Operations in Depth

8.8 Using the File Pointer (seek)

8.9 Reading Text into the RPN Project

8.9.1 The RPN Interpreter to Date

8.9.2 Reading RPN from a Text File

8.9.3 Adding an Assignment Operator to RPN

8.10 Direct Binary Read/Write

8.11 Converting Data to Fixed-Length Fields (struct)

8.11.1 Writing and Reading One Number at a Time

8.11.2 Writing and Reading Several Numbers at a Time

8.11.3 Writing and Reading a Fixed-Length String

8.11.4 Writing and Reading a Variable-Length String

8.11.5 Writing and Reading Strings and Numerics Together

8.11.6 Low-Level Details: Big Endian Versus Little Endian

8.12 Using the Pickling Package

Table of Contents

8.13 Using the shelve Package

Summary

Review Questions

Suggested Problems

Chapter 9 Classes and Magic Methods

9.1 Classes and Objects: Basic Syntax

9.2 More About Instance Variables

9.3 The `__init__` and `__new__` Methods

9.4 Classes and the Forward Reference Problem

9.5 Methods Generally

9.6 Public and Private Variables and Methods

9.7 Inheritance

9.8 Multiple Inheritance

9.9 Magic Methods, Summarized

9.10 Magic Methods in Detail

9.10.1 String Representation in Python Classes

9.10.2 The Object Representation Methods

9.10.3 Comparison Methods

9.10.4 Arithmetic Operator Methods

9.10.5 Unary Arithmetic Methods

9.10.6 Reflection (Reverse-Order) Methods

9.10.7 In-Place Operator Methods

9.10.8 Conversion Methods

9.10.9 Collection Class Methods

9.10.10 Implementing `__iter__` and `__next__`

9.11 Supporting Multiple Argument Types

9.12 Setting and Getting Attributes Dynamically

Summary

Table of Contents

Review Questions

Suggested Problems

Chapter 10 Decimal, Money, and Other Classes

10.1 Overview of Numeric Classes

10.2 Limitations of Floating-Point Format

10.3 Introducing the Decimal Class

10.4 Special Operations on Decimal Objects

10.5 A Decimal Class Application

10.6 Designing a Money Class

10.7 Writing the Basic Money Class (Containment)

10.8 Displaying Money Objects (`__str__`, `__repr__`)

10.9 Other Monetary Operations

10.10 Demo: A Money Calculator

10.11 Setting the Default Currency

10.12 Money and Inheritance

10.13 The Fraction Class

10.14 The Complex Class

Summary

Review Questions

Suggested Problems

Chapter 11 The Random and Math Packages

11.1 Overview of the Random Package

11.2 A Tour of Random Functions

11.3 Testing Random Behavior

11.4 A Random-Integer Game

11.5 Creating a Deck Object

Table of Contents

- 11.6 Adding Pictograms to the Deck
- 11.7 Charting a Normal Distribution
- 11.8 Writing Your Own Random-Number Generator
 - 11.8.1 Principles of Generating Random Numbers
 - 11.8.2 A Sample Generator
- 11.9 Overview of the Math Package
- 11.10 A Tour of Math Package Functions
- 11.11 Using Special Values (π)
- 11.12 Trig Functions: Height of a Tree
- 11.13 Logarithms: Number Guessing Revisited
 - 11.13.1 How Logarithms Work
 - 11.13.2 Applying a Logarithm to a Practical Problem

Summary

Review Questions

Suggested Problems

Chapter 12 The numpy (Numeric Python) Package

- 12.1 Overview of the array, numpy, and matplotlib Packages
 - 12.1.1 The array Package
 - 12.1.2 The numpy Package
 - 12.1.3 The numpy.random Package
 - 12.1.4 The matplotlib Package
- 12.2 Using the array Package
- 12.3 Downloading and Importing numpy
- 12.4 Introduction to numpy: Sum 1 to 1 Million
- 12.5 Creating numpy Arrays
 - 12.5.1 The array Function (Conversion to an Array)
 - 12.5.2 The arange Function
 - 12.5.3 The linspace Function

Table of Contents

12.5.4 The empty Function

12.5.5 The eye Function

12.5.6 The ones Function

12.5.7 The zeros Function

12.5.8 The full Function

12.5.9 The copy Function

12.5.10 The fromfunction Function

12.6 Example: Creating a Multiplication Table

12.7 Batch Operations on numpy Arrays

12.8 Ordering a Slice of numpy

12.9 Multidimensional Slicing

12.10 Boolean Arrays: Mask Out That numpy!

12.11 numpy and the Sieve of Eratosthenes

12.12 Getting numpy Stats (Standard Deviation)

1 2.13 Getting Data on numpy Rows and Columns

Summary

Review Questions

Suggested Problems

Chapter 13 Advanced Uses of numpy

13.1 Advanced Math Operations with numpy

13.2 Downloading mat plotlib

13.3 Plotting Lines with numpy and matplotlib

13.4 Plotting More Than One Line

13.5 Plotting Compound Interest

13.6 Creating Histograms with matplotlib

13.7 Circles and the Aspect Ratio

13.8 Creating Pie Charts

Table of Contents

13.9 Doing Linear Algebra with numpy

13.9.1 The Dot Product

13.9.2 The Outer-Product Function

13.9.3 Other Linear Algebra Functions

13.10 Three-Dimensional Plotting

13.11 numpy Financial Applications

13.12 Adjusting Axes with xticks and yticks

13.13 numpy Mixed-Data Records

13.14 Reading and Writing numpy Data from Files

Summary

Review Questions

Suggested Problems

Chapter 14 Multiple Modules and the RPN Example

14.1 Overview of Modules in Python

14.2 Simple Two-Module Example

14.3 Variations on the import Statement

14.4 Using the `__all__` Symbol

14.5 Public and Private Module Variables

14.6 The Main Module and `__main__`

14.7 Gotcha! Problems with Mutual Importing

14.8 RPN Example: Breaking into Two Modules

14.9 RPN Example: Adding I/O Directives

14.10 Further Changes to the RPN Example

14.10.1 Adding Line-Number Checking

14.10.2 Adding Jump-If-Not-Zero

14.10.3 Greater-Than (>) and Get-Random-Number (!)

14.11 RPN: Putting It All Together

Table of Contents

Summary

Review Questions

Suggested Problems

Chapter 15 Getting Financial Data off the Internet

15.1 Plan of This Chapter

15.2 Introducing the Pandas Package

15.3 stock_load: A Simple Data Reader

15.4 Producing a Simple Stock Chart

15.5 Adding a Title and Legend

15.6 Writing a makeplot Function (Refactoring)

15.7 Graphing Two Stocks Together

15.8 Variations: Graphing Other Data

15.9 Limiting the Time Period

15.10 Split Charts: Subplot the Volume

15.11 Adding a Moving-Average Line

15.12 Giving Choices to the User

Summary

Review Questions

Suggested Problems

Appendix A: Python Operator Precedence Table

Appendix B: Built-In Python Functions

abs(x)

all(iterable)

any(iterable)

ascii(obj)

bin(n)

Table of Contents

bool(obj)
bytes(source, encoding)
callable(obj)
chr(n)
compile(cmd_str, filename, mode_str, flags=0, dont_inherit=False,
optimize=1)
complex(real=0, imag=0)
complex(complex_str)
delattr(obj, name_str)
dir([obj])
divmod(a, b)
enumerate(iterable, start=0)
eval(expr_str [, globals [,locals]])
exec(object [, global [,locals]])
filter(function, iterable)
float([x])
format(obj, [format_spec])
frozenset([iterable])
getattr(obj, name_str [,default])
globals()
hasattr(obj, name_str)
hash(obj)
help([obj])
hex(n)
id(obj)
input([prompt_str])
int(x, base=10)

Table of Contents

int()
isinstance(obj, class)
issubclass(class1, class2)
iter(obj)
len(sequence)
list([iterable])
locals()
map(function, iterable1 [, iterable2])
max(arg1 [, arg2])
max(iterable)
min(arg1 [, arg2])
min(iterable)
oct(n)
open(file_name_str, mode='rt')
ord(char_str)
pow(x, y[, z])
print(objects, sep=" ", end=''
 ', file=sys.stdout)
range(n)
range(start, stop [, step])
repr(obj)
reversed(iterable)
round(x [,ndigits])
set([iterable])
setattr(obj, name_str, value)
sorted(iterable [, key] [, reverse])
str(obj="")

Table of Contents

`str(obj=b" [, encoding='utf-8'])`

`sum(iterable [, start])`

`super(type)`

`tuple([iterable])`

`type(obj)`

`zip(*iterables)`

Appendix C: Set Methods

`set_obj.add(obj)`

`set_obj.clear()`

`set_obj.copy()`

`set_obj.difference(other_set)`

`set_obj.difference_update(other_set)`

`set_obj.discard(obj)`

`set_obj.intersection(other_set)`

`set_obj.intersection_update(other_set)`

`set_obj.isdisjoint(other_set)`

`set_obj.issubset(other_set)`

`set_obj.issuperset(other_set)`

`set_obj.pop()`

`set_obj.remove(obj)`

`set_obj.symmetric_difference(other_set)`

`set_obj.symmetric_difference_update(other_set)`

`set_obj.union(other_set)`

`set_obj.union_update(other_set)`

Appendix D: Dictionary Methods

`dict_obj.clear()`

Table of Contents

dict_obj.copy()
dict_obj.get(key_obj, default_val = None)
dict_obj.items()
dict_obj.keys()
dict_obj.pop(key [, default_value])
dict_obj.popitem()
dict_obj.setdefault(key, default_value=None)
dict_obj.values()
dict_obj.update(sequence)

Appendix E: Statement Reference

Variables and Assignments

Spacing Issues in Python

Alphabetical Statement Reference

assert Statement
break Statement
class Statement
continue Statement
def Statement
del Statement
elif Clause
else Clause
except Clause
for Statement
global Statement
if Statement
import Statement
nonlocal Statement
pass Statement

Table of Contents

raise Statement
return Statement
try Statement
while Statement
with Statement
yield Statement

Index