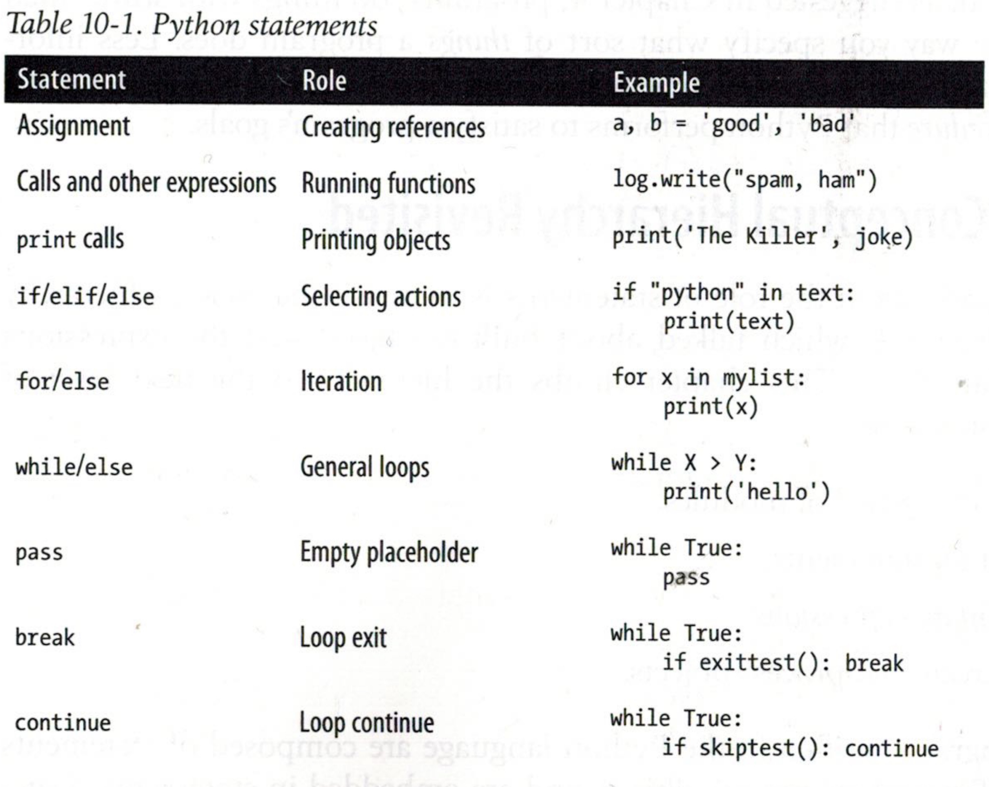
# Python Statements



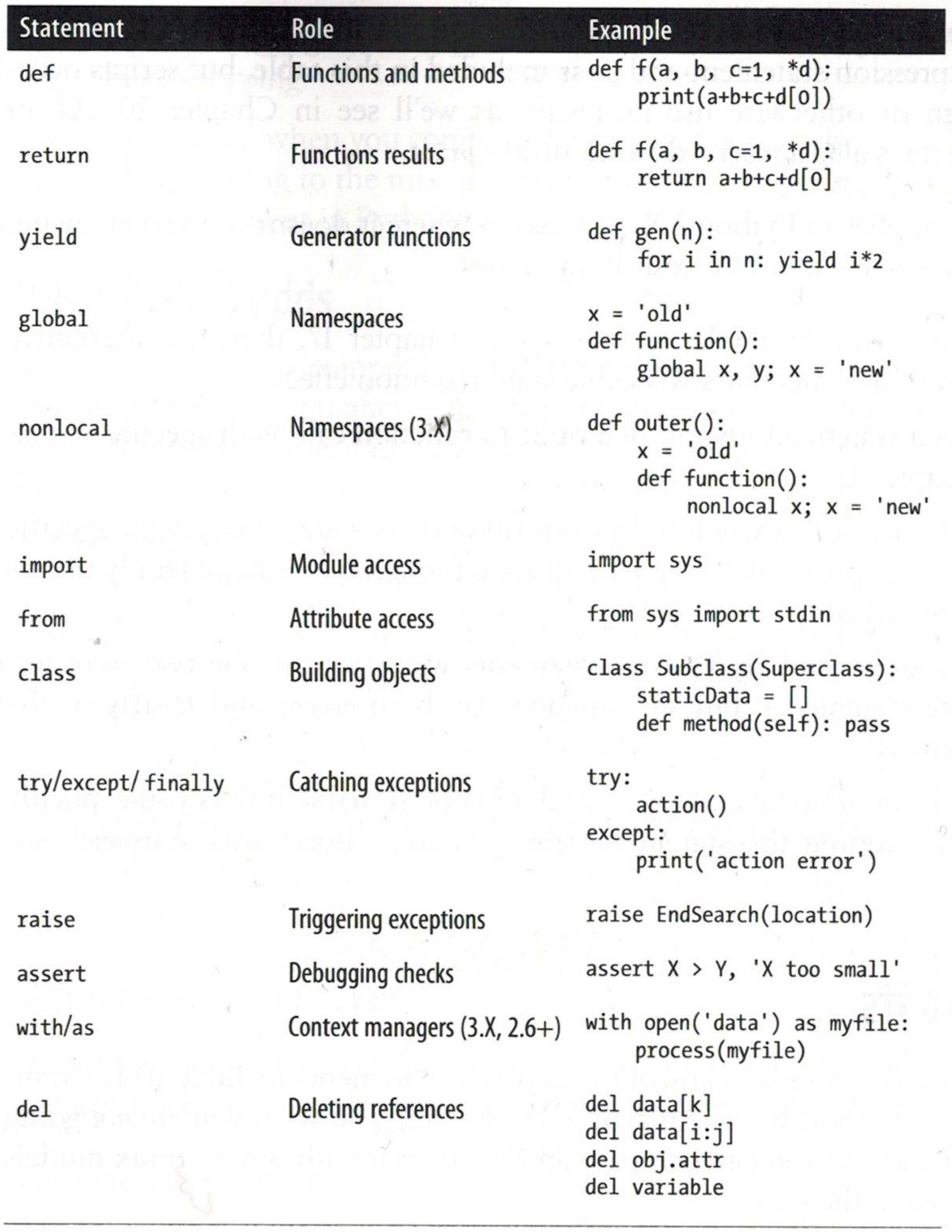


*Table 10-1. Python statements*

| Statement | Role | Example |
|---|---|---|
| Assignment | Creating references | `a, b = 'good', 'bad'` |
| Calls and other expressions | Running functions | `log.write("spam, ham")` |
| print calls | Printing objects | `print('The Killer', joke)` |
| if/elif/else | Selecting actions | `if "python" in text:`<br>`    print(text)` |
| for/else | Iteration | `for x in mylist:`<br>`    print(x)` |
| while/else | General loops | `while X > Y:`<br>`    print('hello')` |
| pass | Empty placeholder | `while True:`<br>`    pass` |
| break | Loop exit | `while True:`<br>`    if exittest(): break` |
| continue | Loop continue | `while True:`<br>`    if skiptest(): continue` |
| def | Functions and methods | `def f(a, b, c=1, *d):`<br>`    print(a+b+c+d[0])` |
| return | Functions results | `def f(a, b, c=1, *d):`<br>`    return a+b+c+d[0]` |
| yield | Generator functions | `def gen(n):`<br>`    for i in n: yield i*2` |
| global | Namespaces | `x = 'old'`<br>`def function():`<br>`    global x, y; x = 'new'` |
| nonlocal | Namespaces (3.X) | `def outer():`<br>`    x = 'old'`<br>`    def function():`<br>`        nonlocal x; x = 'new'` |
| import | Module access | `import sys` |
| from | Attribute access | `from sys import stdin` |
| class | Building objects | `class Subclass(Superclass):`<br>`    staticData = []`<br>`    def method(self): pass` |
| try/except/ finally | Catching exceptions | `try:`<br>`    action()`<br>`except:`<br>`    print('action error')` |
| raise | Triggering exceptions | `raise EndSearch(location)` |
| assert | Debugging checks | `assert X > Y, 'X too small'` |
| with/as | Context managers (3.X, 2.6+) | `with open('data') as myfile:`<br>`    process(myfile)` |
| del | Deleting references | `del data[k]`<br>`del data[i:j]`<br>`del obj.attr`<br>`del variable` |

Lutz (2013), Ch. 10, pp330-331

# Combined Assignment Operators

Examples: **a = 5 & b= 2**

| Assignment operator | Name | Example | Meaning | Result |
|---|---|---|---|---|
| = | Simple assignment | a = b | Set a as b | 2 |
| += | Add AND | a += b | a = a +b | 7 |
| -= | Subtract AND | a -= b | a= a- b | 3 |
| *= | Multiply AND | a *= b | a = a * b | 10 |
| /= | Divide AND | a /= b | a = a /b | 2 |
| %= | Modulus AND | a % = b | a = a % b | 1 |
| **= | Exponent AND | a ** = b | a = a ** b | 25 |

# Multiple Assignments & Statements

- The following “multiple assignments” are possible

```
>>> a = b = c = 0
>>> a1, b1, c1 = 1, 1.0, ‘c1’
>>> (a2, b2, c2) = 2, 2.0, ‘c2’
>>> print a,b,c,a1,b1,c1,a2,b2,c2
0 0 0 1 1.0 c1 2 2.0 c2
```

- Multiple statements on one line

```
>>> a3 = 3; b3 = 3.0 ; c3 = ‘c3’
```