

# Control Flow

## Chapter 8

# This chapter covers

- Repeating code with a while loop
- Making decisions: the if-elif-else statement
- Iterating over a list with a for loop
- Using list and dictionary comprehensions
- Delimiting statements and blocks with indentation
- Evaluating Boolean values and expressions

```
1  u, v, x, y = 0, 0, 100, 30
2
3  while x > y:
4      u = u + y
5      x = x - y
6      if x < y + 2:
7          v = v + x
8          x = 0
9      else:
10         v = v + y + 2
11         x = x - y - 2
12 print(u, v)
```

## The while loop

```
1  x = 5
2
3  if x < 5:
4      y = -1
5      z = 5
6  elif x > 5:
7      y = 1
8      z = 11
9  else:
10     y = 0
11     z = 10
12
13 print(x, y, z)
```

# The if-elif-else statement

# The for loop

```
1 item_list = [3, "string1", 23, 14.0, "string2", 49, 64, 70]
2
3 for x in item_list:
4     if not isinstance(x, int):
5         continue
6     if not x % 7:
7         print("found an integer divisible by seven: %d" % x)
8         break
```

```
43  # Using the range() function:
44  ✓ for x in range(6):
45      |     print(x)
46
47  # Using the start parameter:
48  ✓ for x in range(2, 6):
49      |     print(x)
50
51  # Increment the sequence with 3 (default is 1):
52  ✓ for x in range(2, 30, 3):
53      |     print(x)
54
55  # Print all numbers from 0 to 5, and print a message when the
    # sequence is
    # ended:
56  ✓ for x in range(6):
57      |     print(x)
58  ✓ else:
59      |     print("Finally finished!")
```

# The range function



## QUICK CHECK

---

- Suppose that you have a list  $x = [1, 3, 5, 0, -1, 3, -2]$ , and you need to remove all negative numbers from that list. Write the code to do this.
- How would you count the total number of negative numbers in a list  
 $y = [[1, -1, 0], [2, 5, -9], [-2, -3, 0]]$ ?
- What code would you use to print very low if the value of  $x$  is below -5, low if it's from -5 up to 0, neutral if it's equal to 0, high if it's greater than 0 up to 5, and very high if it's greater than 5?

# Boolean values and expressions

- Python has a Boolean object type that can be set to either True or False. Any expression with a Boolean operation returns True or False.
  - The numbers 0, 0.0, and 0+0j are all False; any other number is True.
  - The empty string "" is False; any other string is True.
  - The empty list [] is False; any other list is True.
  - The empty dictionary {} is False; any other dictionary is True.
  - The empty set set() is False; any other set is True.
  - The special Python value None is always False.



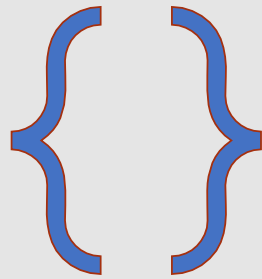
# Writing a simple program to analyze a text file

## Example

# Lab

Refactor word\_count

# Summary



- Python uses indentation to group blocks of code.
- Python has loops using while and for, and conditionals using if-elif-else.
- Python has the Boolean values True and False, which can be referenced by variables.
- Python also considers any 0 or empty value to be False and any nonzero or nonempty value to be True.