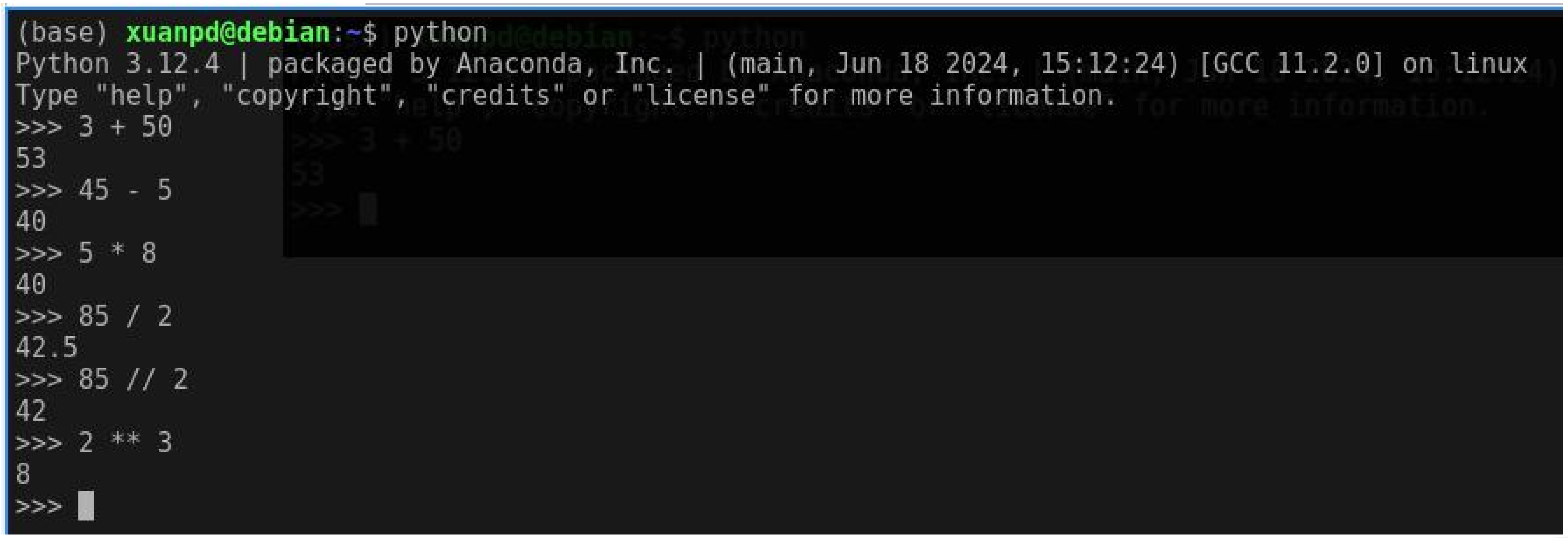
Chapter 1
Programming as a Way of Thinking

A. Downey, *Think Python: How to Think Like a Computer Science,* 3rd ed., O'Reilly, 2024.

Contents

1. Arithmetic Operators

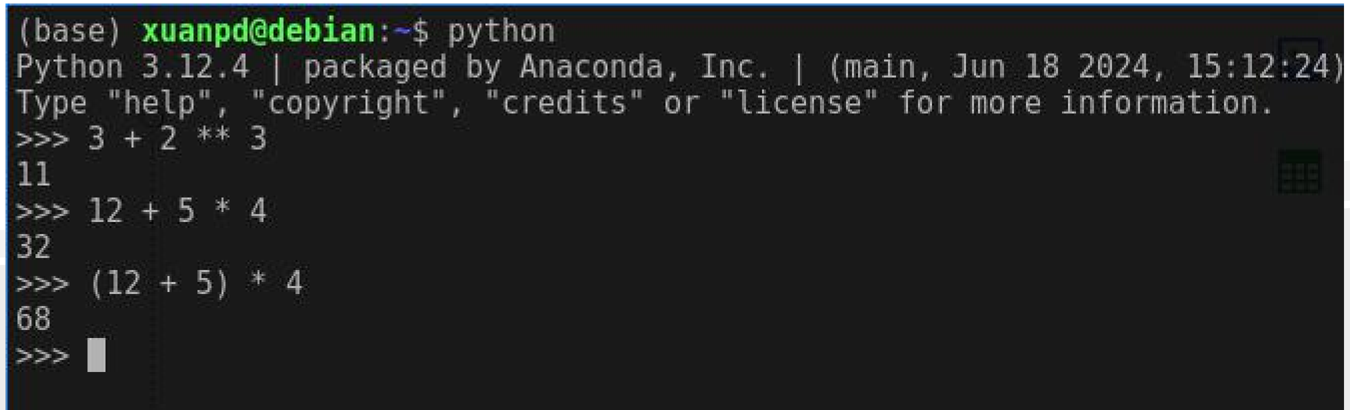
- An arithmetic operator is a symbol that represents an arithmetic computation. For example, the plus sign, +, performs addition:



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:24) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 + 50
53
>>> 45 - 5
40
>>> 5 * 8
40
>>> 85 / 2
42.5
>>> 85 // 2
42
>>> 2 ** 3
8
>>> 
```

2. Expressions

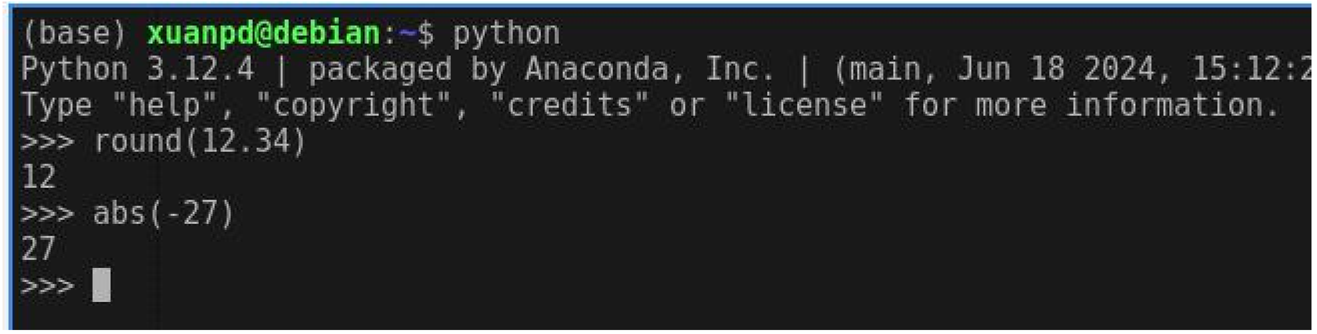
- A collection of operators and numbers is called an expression. An expression can contain any number of operators and numbers.
- Every expression has a value.
- Precedence of operators
  - Python follows the order of operations you might have learned in a math class: exponentiation operator has a higher precedence than multiplication and division, which have higher precedence than addition and subtraction.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:24)
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 + 2 ** 3
11
>>> 12 + 5 * 4
32
>>> (12 + 5) * 4
68
>>> 
```

3. Arithmetic Functions

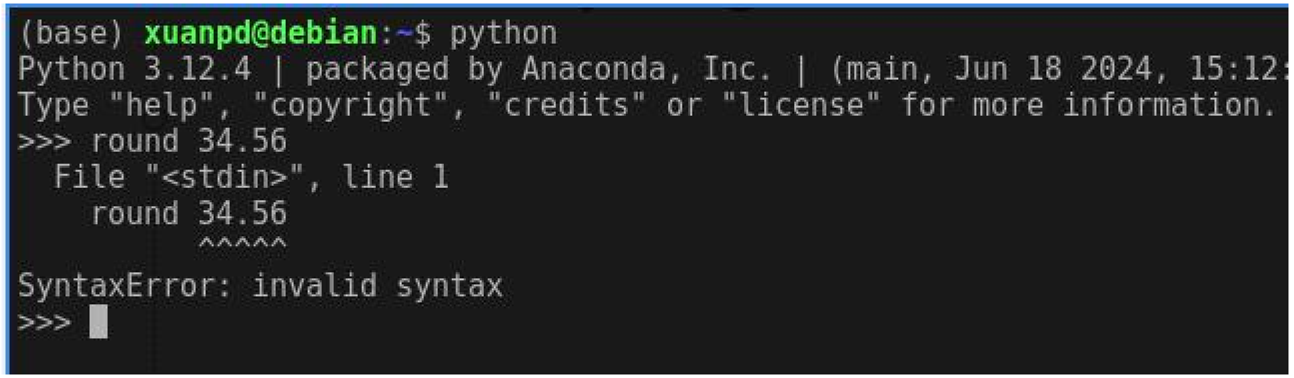
- In addition to the arithmetic operators, Python provides a few functions that work with numbers.
  - For example, the round function takes a floating-point number and rounds it off to the nearest whole number.
  - The abs function computes the absolute value of a number.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:2
Type "help", "copyright", "credits" or "license" for more information.
>>> round(12.34)
12
>>> abs(-27)
27
>>> 
```

3. Arithmetic Functions

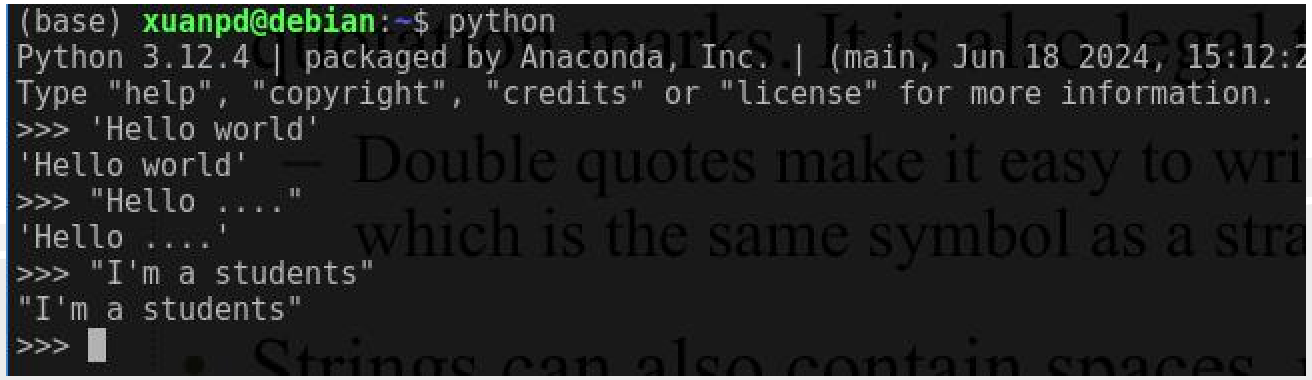
- When we use a function like this, we say we're **calling** the function. An expression that calls a function is a **function call**.
- When you call a function, the parentheses are required. If you leave them out, you get an error message:



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:
Type "help", "copyright", "credits" or "license" for more information.
>>> round 34.56
  File "<stdin>", line 1
    round 34.56
          ^^^^^
SyntaxError: invalid syntax
>>> 
```

4. Strings

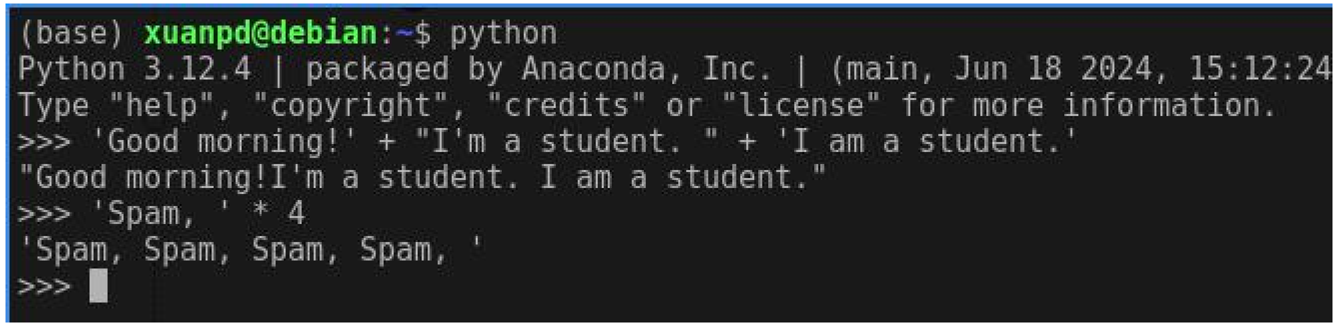
- In addition to numbers, Python can also represent sequences of letters, which are called strings because the letters are strung together like beads on a necklace.
- To write a string, we can put a sequence of letters inside straight quotation marks. It is also legal to use double quotation marks.
  - Double quotes make it easy to write a string that contains an apostrophe, which is the same symbol as a straight quote.
- Strings can also contain spaces, punctuation, and digits.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:2
Type "help", "copyright", "credits" or "license" for more information.
>>> 'Hello world'
'Hello world'
>>> "Hello ...."
'Hello ....'
>>> "I'm a students"
"I'm a students"
>>> 
```

4. Strings

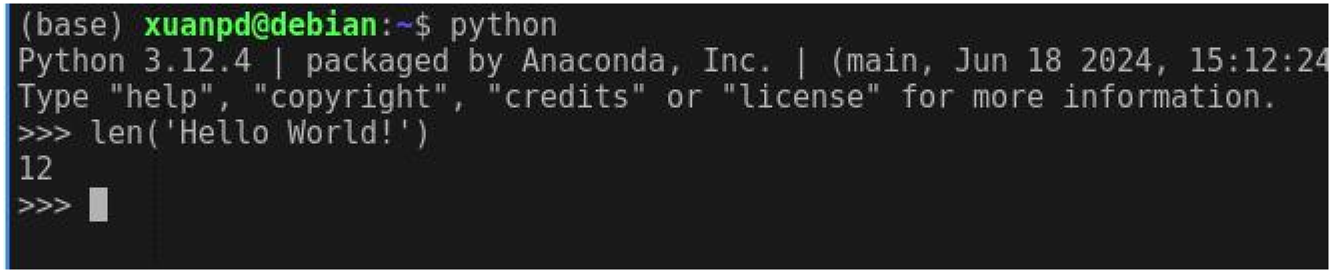
- The + operator works with strings; it joins two strings into a single string, which is called **concatenation**.
- The * operator also works with strings; it makes multiple copies of a string and concatenates them.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:24
Type "help", "copyright", "credits" or "license" for more information.
>>> 'Good morning!' + "I'm a student. " + 'I am a student.'
"Good morning!I'm a student. I am a student."
>>> 'Spam, ' * 4
'Spam, Spam, Spam, Spam, '
>>> 
```

4. Strings

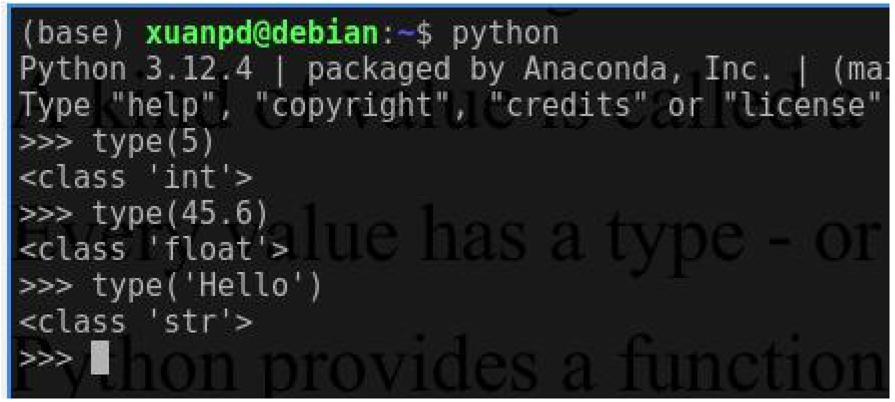
- Python provides a function called **len** that computes the length of a string.
  - Notice that **len** counts the letters in the string.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:24
Type "help", "copyright", "credits" or "license" for more information.
>>> len('Hello World!')
12
>>> 
```

5. Values and Types

- So far we've seen three kinds of values:
  - 2 is an integer,
  - 42.0 is a floating-point number
  - 'Hello' is a string.
- A kind of value is called a **type**.

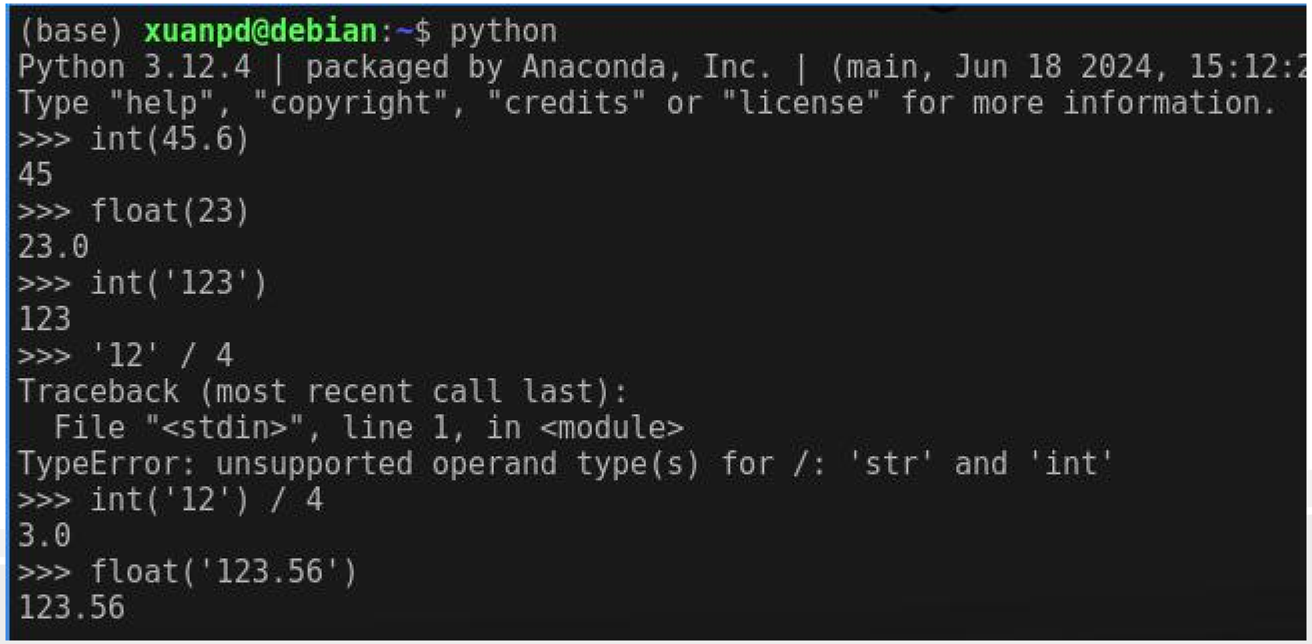


```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (ma
Type "help", "copyright", "credits" or "license"
>>> type(5)
<class 'int'>
>>> type(45.6)
<class 'float'>
>>> type('Hello')
<class 'str'>
>>>
```

- Every value has a type - or we sometimes say it "belongs to" a type.
- Python provides a function called **type** that tells you the type of any value.
  - The type of an integer is **int**.

5. Values and Types

- The types int, float, and str can be used as functions.
  - For example, int can take a floating-point number and convert it to an integer (always rounding down).
  - And float can convert an integer to a floating-point value.



```
(base) xuanpd@debian:~$ python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:12:2
Type "help", "copyright", "credits" or "license" for more information.
>>> int(45.6)
45
>>> float(23)
23.0
>>> int('123')
123
>>> '12' / 4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for /: 'str' and 'int'
>>> int('12') / 4
3.0
>>> float('123.56')
123.56
```

5. Values and Types

- When you write a large integer, you might be tempted to use commas between groups of digits, as in 1,000,000. This is a legal expression in Python, but the result is not an integer.
  - Python interprets 1,000,000 as a comma-separated sequence of integers.
- You can use underscores to make large numbers easier to read:
  - 1_000_000