1 Basics

Exercise 1.1: The interpreter

Open the Python interpreter. What happens when you input the following statements:

(a) `3 + 1`

(b) `3 * 3`

(c) `2 ** 3`

(d) `"Hello, world!"`

Exercise 1.2: Scripts

Now copy the above to a script, and save it as `script1.py`. What happens if you run the script? (try: `python script1.py`). Can you fix this? (hint: use the `print` function)

Exercise 1.3: More interpreter

Explain the output of the following statements if executed subsequently:

(a) `'py' + 'thon'`

(b) `'py' * 3 + 'thon'`

(c) `'py' - 'py'`

(d) `'3' + 3`

(e) `3 * '3'`

(f) `a`

(g) `a = 3`

(h) `a`

Exercise 1.4: Booleans

Explain the output of the following statements:

(a) `1 == 1`

(b) `1 == True`

(c) `0 == True`

(d) `0 == False`

(e) `3 == 1 * 3`

(f) `(3 == 1) * 3`

(g) `(3 == 3) * 4 + 3 == 1`

(h) `3**5 >= 4**4`

Exercise 1.5: Integers

Explain the output of the following statements:

(a) `5 / 3`

(b) `5 % 3`

(c) `5.0 / 3`

(d) `5 / 3.0`

(e) `5.2 % 3`

(f) `2001 ** 200`

Exercise 1.6: Floats

Explain the output of the following statements:

(a) `2000.3 ** 200` (compare with above)

(b) `1.0 + 1.0 - 1.0`

(c) `1.0 + 1.0e20 - 1.0e20`

Exercise 1.7: Variables

Write a script where the variable `name` holds a string with your name. Then, assuming for now your name is *John Doe*, have the script output: `Hello, John Doe!` (and obviously, do not use `print "Hello, John Doe!"`.

Exercise 1.8: Type casting

Very often, one wants to "cast" variables of a certain type into another type. Suppose we have variable `x = '123'`, but really we would like `x` to be an integer.

This is easy to do in Python, just use `desiredtype(x)`, e.g. `int(x)` to obtain an integer.

Try the following and explain the output

(a) `float(123)`

(b) `float('123')`

(c) `float('123.23')`

(d) `int(123.23)`

(e) `int('123.23')`

(f) `int(float('123.23'))`

(g) `str(12)`

(h) `str(12.2)`

(i) `bool('a')`

(j) `bool(0)`

(k) `bool(0.1)`

2 Control flow

Disclaimer: Some of the following problems are inspired by problems from www.projecteuler.net. Have a look if you are interested, there are some great challenges and Python is an excellent tool for solving them.

Exercise 2.1: Range

Type `range(5)` in the interpreter, what does the interpreter return? So what does `for i in range(5)` mean?

Let's also find out whether the interpreter can help us understand the object '`range(5)`' better. Type `type(range(5))` in the interpreter. More on this soon!

Exercise 2.2: For loops

Use a `for` loop to:

(a) Print the numbers 0 to 100

(b) Print the numbers 0 to 100 that are divisible by 7

(c) Print the numbers 1 to 100 that are divisible by 5 but not by 3

(d) Print for each of the numbers $x = 2, \ldots 20$, all numbers that divide x, excluding 1 and x. Hence, for `18`, it should print `2 3 6 9`.

Hint: see https://docs.python.org/2.7/library/functions.html#range.

Exercise 2.3: Simple while loops

Instead of using a for loop, use a while loop to:

(a) Print the numbers 0 to 100

(b) Print the numbers 0 to 100 that are divisible by 7

Exercise 2.4: While loops

Use a `while` loop to find the first 20 numbers that are divisible by 5, 7 and 11, and print them Hint: store the number found so far in a variable.

Pseudo-code:

```
number found = 0
x = 11
while number found is less than 20:
    if x is divisible by 5, 7 and 11:
        print x
        increase number found by 1
    increase x by 1
```

Exercise 2.5: More while loops

The smallest number that is divisible by 2, 3 and 4 is 12. Find the smallest number that is divisible by all integers between 1 and 10.

Exercise 2.6: Collatz sequence

A Collatz sequence is formed as follows: We start with some number x_0, and we find the next number in the sequence by

$$x_{i+1} = \begin{cases} x_i/2 & \text{if } x_i \text{ is even} \\ 3x_i + 1 & \text{if } x_i \text{ is odd} \end{cases}$$

If $x_i = 1$, we stop iterating and have found the full sequence.

For example, if we start with $x_0 = 5$, we obtain the sequence:

```
5 16 8 4 2 1
```

It is conjectured, though not proven, that every chain eventually ends at 1.

Print the Collatz sequence starting at $x_0 = 103$.