Chapter 3: if statements

Author: Vincent Delecroix
License: CC BY-SA 3.0

Comparisons and if

The comparison signs in Python and many other programming languages are as follows

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equality</td>
</tr>
<tr>
<td>!=</td>
<td>difference</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
</tr>
</tbody>
</table>

Exercise 3.1

Which number is the largest $1000^{1001}$ or $1001^{1000}$?

Exercise 3.2

Let us consider the following code:

```sage
a = # enter a value for a
if a != 2:
    print('lost')
elif a == 3:
    print('an instant, please')
else:
    print('you win')
```

What is the above program doing

- when the variable `a` is 1?
- when the variable `a` is 2?
- when the variable `a` is 3?
- when the variable `a` is 15?

Exercise 3.3

Two prime numbers $p$ and $q$ are said twin if $q = p + 2$. Find all twin prime numbers below 10000.
Exercise 3.4

Find the smallest and largest integers in the set

$$\{a^b - b^a : a \in \{1, 2, \ldots, 5\}, b \in \{1, 2, \ldots, 5\}\}$$

Exercise 3.5

Recall that the method digits of an integer returns the list of its digits:

```
sage: 1527.digits()
```

Solve Euler problem 56 by finding the maximal sum of digits of numbers of the form $a^b$ with both $a$ and $b$ lesser than 100

Exercise 3.6

Solve Euler problem 4 about palindromes.

Exercise 3.7

Let us consider the following list of integers:

```
sage: l = [123, 414, 264, 18, 689, 21, 5571, 28, 589, 12, 111, 231, ....: 158, 551, 250, 68, 5728, 2222, 4198, 571, 28, 518, 999, 444, ....: 112, 689, 672, 334, 680, 273]
```

Construct two lists `leven` and `lodd` that contain respectively the even and odd elements of `l`.

Using in and not in

The condition of an if or elif statement is not necessarily a comparison. Basically, any Python object would fit!

```
sage: a = 5
sage: if a:
....:   print(‘I am not zero’)
```

What happens under the hood is that the object `a` (here an integer) is converted to a boolean value (True or False). You can see the boolean value of an object by using `bool`

```
sage: bool(5)
sage: bool(0)
sage: bool([])
sage: bool([0])
```

A useful construction is obtained with the keyword in: the result of `a in b` is whether `a` belongs to the object `b`. For example:

```
sage: 2 in ZZ
sage: 2/3 in ZZ
sage: 2/3 in QQ
sage: 1 in [3, 5, 2, 1, 2, 8]
```

2
sage: ‘a’ in ‘Saint-Flour’
sage: ‘z’ in ‘Saint-Flour’

To check that an element is not in a given object use a not in b:

sage: 10 not in Primes()
sage: 5/2 not in ZZ

**Exercise 3.8**

Using an if statement involving in inside a for loop, count the number of vowels in the string:

sage: s = ‘How many vowels are present in this sentence?’

Count the number of consonant in the string:

sage: s = ‘How many consonants are present in this sentence?’

**Exercise 3.9 (Pythagorean triples)**

A Pythagorean triple is a triple \((a, b, c)\) of positive integers so that \(a^2 + b^2 = c^2\). An example is \(3^2 + 4^2 = 5^2\).

How many Pythagorean triples are there with \(a, b\) and \(c\) smaller than 100?

Solve Euler problem by finding the unique Pythagorean triple so that \(a + b + c = 1000\)

**Combining conditions or, and and not**

To make even more complicated tests you can combine them. The main operators for this are or, and.

sage: n = 17
sage: if n.is_prime() and (n+2).is_prime():
    ....:   print(‘a twin number!’)

**Exercise 3.10**

Let us call a positive integer \(n\) a triple twin if all of \(n\), \(n+2\) and \(n+6\) are primes. How many triple twins are there smaller than 10000?

The operator not is used for negation of a condition.

sage: not True
sage: not False

**More exercises**

For more exercises in the same vein you can challenge yourself with

- Euler problem 30 (sum of certain numbers)
- Euler problem 33 (digit cancelling fractions)
- Euler problem 34 (numbers which are sum of factorials of their digits)
- Euler problem 35 (circular primes)
• **Euler problem 36** (integers palindromic in base 2 and 10)
• **Euler problem 37** (truncatable primes)
• **Euler problem 38** (integer right triangles, aka pythagorean triples)
• **Euler problem 39** (binomials greater than a million)
• **Euler problem 40** (continued fractions)