Compact Course Python
Exercise 2

1 Leap Year

Write a program that reads a year (a number) from the command line and checks whether or not the year is a leap year. Print the result to to the screen.

```
> python Leap.py 1999
Is not a leap year.
> python Schaltjahr.py 2008
Is a leap year.
```

Tip: with `sys.argv[i]` you can acces the `i`-th command line argument (as String; see last slide). With `int(s)` you can convert a string `s` into an integer.

Thinking Exercise: The implementation that suggests itself uses the modulo operator (`%`) to test whether a year number is divisible without remainder. Try to figure out a general rule for modelling the modulo operator with other operators.

2 Eternal Calendar

On the Wikipedia page describing *Zeller's congruence*[1], you'll find a formula that computes the weekday for an arbitrary date.

Write a Python program that reads day, month (as a number) and year from the command line and outputs the corresponding weekday, computed with Zeller's formula (use the one for the Gregorian calendar).

```
> python EternalCalendar.py 20 1 2000
Thursday
```

Mind the special treatment for January and February!

[1] http://en.wikipedia.org/wiki/Zeller%27s_congruence

3 Strings to Numbers

For No. 1 and to we use `int`, to convert Strings to numbers. Re-implement this functionality, i.e. write a program, that converts a string into a (decimal) number (e.g. `"123"` into `123`).

Tips: `ord(ch) - ord('0')` gives you the numeric value of a digit represented as a string `ch`.

4 More on operators

(a) What happens in the following code snippet? `a` and `b` are two variables that have some numeric values.

```
a = a + b
b = a - b
a = a - b
```

(b) Could you write a different algorithm that achieves the same result, but which is easier to read?