

The deadline for this exercise sheet is **Monday, 11.06.2018, 12:00**.

Introductory Words

Remember that you need proper documentation to pass the homework. The documentation doesn't need to be *perfect*, but everything that needs a docstring, should have a docstring.

1 Warm-Up: The Final Countdown

Write a script that uses the `datetime` module to make a countdown from 10 seconds to 0 seconds in which each second the remaining time is printed.

Bonus: Make the countdown more dramatic. We don't care how exactly, you can get creative! (One idea would be to include sounds).

2 BirthdayCalc

Usually, when asked how old we are, we can only answer in years (if even). Let's get some more information on how much time we've already spent on this lovely planet.

Write a module named `birthday_calc` using which one can create `BirthdayCalc` objects that can return how much time has passed since one's birthday in different units.

A `BirthdayCalc` object is initialized with a day, a month and a year and a day (of the birthday) and store this as a `datetime` object. There have to be following methods in the class:

- `birthday_calc`, which returns your birthday as a `datetime` object
- `years_since_birth`,
- `months_since_birth`,
- `days_since_birth`,
- `hours_since_birth`,
- `minutes_since_birth` and
- `seconds_since_birth`, each of which will return an **integer** containing how much time has passed since the birthday in the respective unit.

Attached you will find a module named `birthday_fun`. This is a test class which you can run to test your implementation.

Bonus: Extend `BirthdayCalc` by a function that returns the days left until the user's next birthday. Also extend `birthday_fun` such that the user is able to use this new function. *Note:* You do not need to take care of error handling

this time. The test class will do this for you.

Note 2: Instead of looking out for leap years, you can assume that a year has exactly 365.25 days.