# Python for webscraping module

Gabriele Rovigatti
email: gabriele.rovigatti@gmail.com
Room 41, $1^{st}$ floor, building B

The aim of the course is both to provide the students with the basics of Pyhton (3.X.X) and to let them start working on simple webscraping tasks. At the end of the module, they are expected to be able to scrape standard websites, extract usable information and store it in machine-readable format - i.e., to obtain datasets for statistical analyses starting from plain webpages.

**Course prerequisites** Students are supposed to have a basic knowledge of the concepts of `function`, `lists` and websites. Being able to use text software, manage comma separated files and general usage of Windows is a prerequisite.

**Material** All the material, references, code and presentations will be posted on a dedicated page on my personal webpage: https://sites.google.com/view/gabrielerovigatti/home

**Program - overview** Python is a flexible high-level programming language: it is used for statistical and scientific computing, as a scripting language for web applications or in artificial intelligence tasks. Below an outline of the goals of the course:

1) Providing the basics of Python syntax, presenting its working environment and main features will cover the first part of the course. I will present Python Shell (IDLE), the concepts of modules and how to install/import them while running scripts, the peculiar notions of lambda functions, lists, dictionaries. I will then move to the presentation of programming best practices, indentation and code examples $\Rightarrow$ 2-3 hours

2) Describing the modules and the main techniques aimed at dealing with webpages in general - *Inspect*, html language - and with Python in particular - `GET` and `POST` methods, `Selenium webdriver`, `BeautifulSoup`. Analysis of example codes and example pages $\Rightarrow$ 2-4 hours

3) Webscraping with Python: application of all above techniques to a chosen webpage. Information extraction, step-by-step dataset building, file management, final outcome in the form of readable files (.csv, .xlsx, .xml, .json, etc.) $\Rightarrow$ 4-6 hours

**Resources:** Below some useful resources available online.

- Detailed guide for webscraping and data analysis with `BeautifulSoup` (with Python 3) - https://www.dataquest.io/blog/web-scraping-tutorial-python/
- Quick guide for webscraping with `lxml` and `requests` - http://python-guide-pt-br.readthedocs.io/en/latest/scenarios/scrape/
- An open source and collaborative framework for webscraping in a Python environment and quickly writing spiders - https://scrapy.org/