What we did in lesson 1

1. Introductions

   If you couldn't be there, I would be very grateful if you could send me an email with answers to these questions, to add to the others:

   a) What's your department?
   b) What year of PhD are you on?
   c) What's your research topic?
   d) Do you have any previous experience programming?
   e) What do you expect from this course?

2. Installation of a Python environment

   You can use whichever Python distribution you want, or you can even make your own environment piece by piece. For people starting from scratch we installed Anaconda (which sucks for many reasons, but is also very easy to set up for the first time). Once installed, you can use the Jupyter application. You can write code however you prefer, but during lessons and group work we'll mostly use this software.

3. (very brief) Theory about programming

   The contents of this point and the next one are also explained (better) in the readings for next time, further on in this document.

4. Introduction to base Python commands and concepts

   Variables, some data types (`int`, `float`, `string`, `list`), some constructs (`while` loops and conditional `if-else` statments), how to use simple built-in functions (`type()` and `print()`). These commands and more are explained in the readings for next time. Also attached is Jupyter notebook called `lesson1.ipynb` that you can play with.

5. Exercise: find bugs in code

   In the slides `lesson1_findTheBug.pdf` there are some very short programs written in Python, each with its own task to perform. However, they all contain one mistake that prevents them from working as intended. Even if you have no experience with Python, try to understand what the code does and imagine the result. Then check your answer in the following slide, and try to find the mistake in the code. The answer is in the third slide.

Readings before lesson 2

As you could expect from a computer course, we won't rely on textbooks too much. Instead, we will use a selection of the many (many!) freely available resources on the web. Hopefully they'll also be useful for your future works.

Most of the stuff today comes from the book *Introduction to Python for Geographic Data Analysis* by Henrikki Tenkanen, Vuokko Heikinheimo & David Whipp (still a work-in-progress). The authors chose to publish its contents online with a very permissive CC 4.0 license, giving us the opportunity to enjoy their work as it progresses. If you don't see a separate attribution in the following, it means that the links point to their site.
I suggest everybody who's interested (and has time) to read Part I - Python essentials of this book in its entirety. I will link here to specific parts of it that will be important for the next lesson.

Basic Python commands

All the commands that we saw during the lesson are explained in Chapter 2. Basic programming concepts. If you want, you can read the whole thing, but I will list here only the ones you'll need for next time:

- 2.1 Getting started with Python
- 2.2 Lists and indices
- 2.4 `for` loops (*)
- 2.5 Conditional statements
- 2.6 Functions

(*) the `for` loop is a bit different from the `while` loop that we saw during the lesson, but I think that it is even more useful. The basic concept, however, is the same: we want the computer to perform a set of operations over and over again.

I suggest you try to take some time to test the commands on your own (you can make a new Jupyter notebook, or you can modify the one we used during the lesson). Next time you will write programs using all these concepts.

Extra things if you want or need them

Info about programming and Python: paraghraph What is a computer? and following.

How to use JupyterLab: the Get Started section from Project Jupyter. NOTE: there's a short video that's very useful but starts with stuff that may seem too complicated. If you want you can go directly to minute 1:46, when they start talking about the notebooks.

PEP 8, the official Style Guide for Python Code. It's the official document that explains the etiquette for writing in Python, things like how many characters to use in a line, or what names to give variables and functions. *We will not use it.* Really, open this at your own risk: depending on the type of person you are, and your level of interest, you may end up thinking that these people are crazy, or you may start obsessing about these rules.