Preliminaries

Aksel Hiorth

University of Stavanger, Institute for Energy Resources, Norway

Dec 11, 2023

Contents

0.1 Why should you learn programming?

A quick google search will tell you that you should learn to code because it will lead to job opportunities and boost your career. This is almost certainly true. I would highlight that it will let you test out ideas much more efficiently. When you have a lot of domain knowledge of a certain phenomena, you will most likely have ideas that can lead to innovation. As an example, maybe you have observed something that makes you believe that phenomena that most people think are unrelated actually are related. To prove or support you claim you need to collect data from these phenomena and present them together. Since the phenomena are unrelated the data are most likely located different places. With

some basic knowledge of Python you can easily access different files, folders, web pages, scrap data from them, filter the data and visualize them. Furthermore as you are using a computer program there will be much less manual errors. The first step in any data exploratory phase is usually to make some plots and look for patterns visually. The next steps is quantify correlations by e.g. regression analysis or to use more advanced machine learning techniques. Using Python it is almost trivial to get started, it takes of course time to master, but with basic Python knowledge you can easily run through tutorials yourself and become quite advanced within weeks.

1 About this course

The aim of this course is to start with practical applications and gradually move to basic operations. We choose a top-down approach, because we hope it will be more engaging. Thus, we will start with some advanced features and move to basic programming concepts. What we want to do is to get some useful applications up and running fast, and then investigate from a programming perspective what is going on.

This has the consequence that we will introduce basic programming concepts such as types, lists, dictionaries *when it is needed, and only the minimal amount of information*. The challenge with this approach is that there is always more to learn about the basic programming concepts, thus if you feel that you would like to know more about the different concepts you should explore this on your own. See the next section for where to find resources.

2 Online programming resources

This course is supposed to be self contained, but there are of course plenty of online courses, youtube videos, and books that you should take advantage of to improve your understanding. These resources are extremely valuable if you know exactly what your are looking for. As a complete beginner with little or no knowledge of Python it can be confusing if you do not know what you are looking for. Great online sources that cover much of Python basics are w3schools, and A Whirlwind tour of Python. These resources explains quite briefly important concepts and give examples, such as

- specific Python syntax
- data types (float, int, Boolean, etc.)
- data structures (lists, dictionaries, tuples, etc.)
- control flow (if, else, while, for loops etc.)
- functions and classes

3 Stuff you need to do

1. You need to install Python, even if you have installed Python before we recommend you to install the Anaconda distribution. It is straight forward to install, just follow the instructions and choose default options that are suggested.

2. Install an integrated development environment (IDE). An IDE is simply where you write the Python code. After installing Anaconda you should already have Spyder installed, if not you can install it by opening the Anaconda Navigator. You will find the Anaconda Navigator in the start menu in the Anaconda folder, but most likely there will already be a program called Spyder in your program folder. Another IDE is Visual Studio Code or VS Code for short, see figure 1 for two examples. An IDE will help you to write code, because it will give information about the code you write and also help you to find errors.

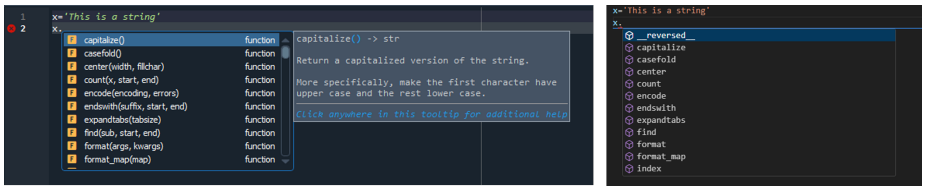

Figure 1: Two IDEs to write Python code (left) Spyder (right) VS Code.

4 Concepts you should know before the course starts

Here we have collected stuff that will make your life easier, and increase the speed of understanding. We have tried to explain some concepts below, if this is too little information there are plenty of online resources that you can check out. The purpose of this section is to introduce you to some concepts that are key to any programming language, but can take some time to master or to get under your skin. If you understand these concepts, coding will be easier. Do not focus on how we use these concepts in coding, that is what the course is all about, rather try to understand the meaning of the concepts.

4.1 Variable

Coding is very much about passing information around and do something with that information. In Python we can easily import an Excel sheet, then we want to pass the Excel sheet around in the code and maybe do some mathematical

operations on the different columns. To pass data around in our code we use *variables*. You can think of a variable as a box that contains data, e.g.:

```
x=13
y='Dog'
```

In the examples above we have two variables `x` and `y`, we have assigned the value 13 and the string `Dog` to them, respectively. The value 13 (or the string `Dog`) is stored somewhere in the computer memory, we can think of this piece of memory as a box with a label, as illustrated in figure 2.

Figure 2: A visualization of a variable.

The illustration in figure 2 also indicate that the size of the box may vary dependent on the content. Note that `x` and `y` are labels, it does not matter what kind of label we use, it is the content of the box that is important not the label you put on it. Normally you would use a more descriptive name than `x` or `y` to simply help other humans to better understand your code.

4.2 Functions

A function is several lines of code that perform a specific task. We can think of a function as a recipe, e.g. a cake recipe. To make a cake we need a certain input, eggs, flour, sugar, chocolate, then we follow a specific set of operations to produce the cake. A function in Python operates in the same way, it takes something as input (different variables), follow certain steps and returns a product (the cake).

Functions are useful because it allows us to wrap several lines of code that we believe we will use many times into reusable functions. Thus, we write the function (recipe) once and every time we want to make the same cake, we invoke the function to produce the output (cake).

4.3 Objects and Classes

In Python everything is an object. An object is a variable (a box) that contains data and functions. That means that the boxes in figure 2 is more than just

pieces of memory. To continue the with the recipe example above, we can think of an object as a cookbook that also contains ingredients, in Norwegian "Matkasse" or in English a "Meal Kit". There will be many recipes in this meal kit and many ingredients. In Python the syntax for accessing the functions (recipes) or data (ingredients) is by using the `.` syntax. In figure 1, this is illustrated. When we write `x='This is a string'`, we can e.g. do `x.capitalize()`, which will (not surprisingly) transform all the small letters to capital letters, `'THIS IS A STRING'`.

Thus in Python there will be a lot of ready made functions that you can use to quickly perform simple operations on your variables.

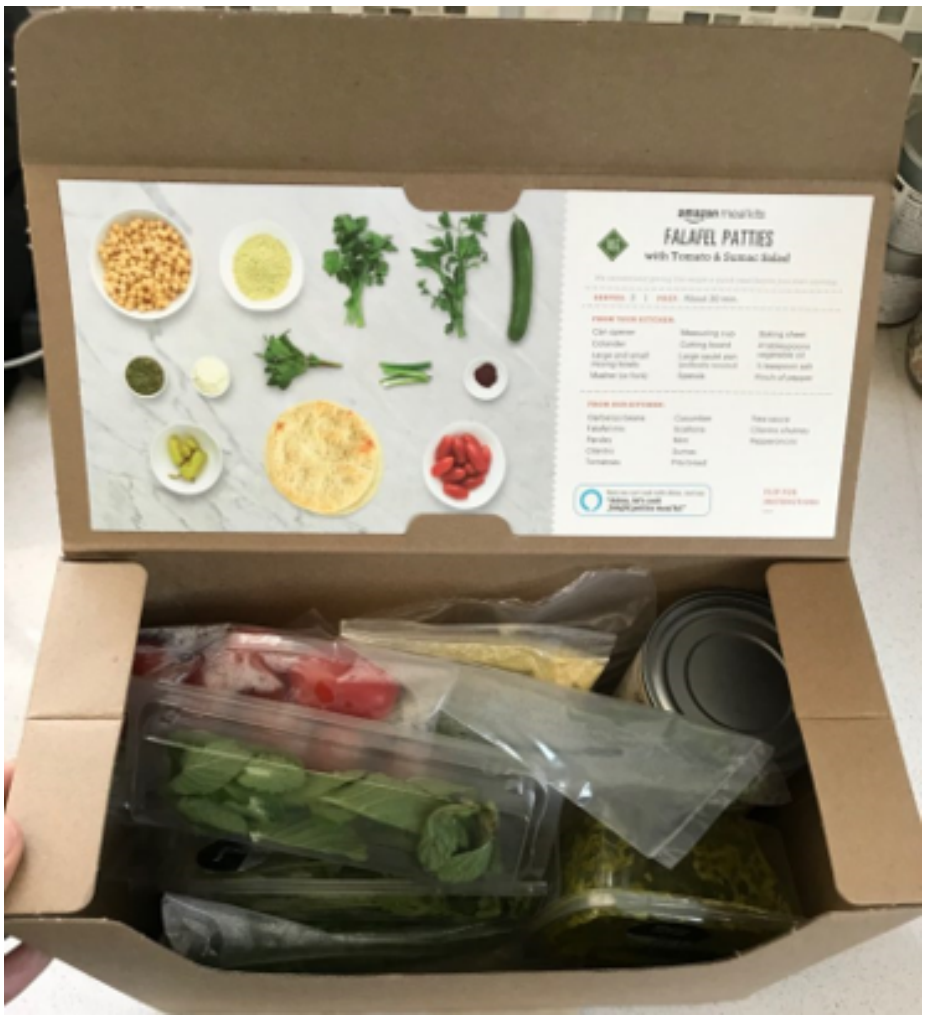

Figure 3: A visualization of an object, containing functions (recipes) and variables (ingredients).

Objects vs Classes.

A class is a blueprint and objects are an instance of the class. We create a class by writing lines of code, to create objects we execute the code in the class. In many ways you can say that objects are physical whereas a class is logical. For the food kit case a class can be a description of the food kit on paper, describing how many recipes, how much potatoes, meat etc. should be included in each food kit, whereas all the food kits delivered to the customers are the objects.

4.4 Library

Python has a lot of libraries, which is one of the reasons why Python is so popular. These libraries are free and you can import them into your code. You can think of a library as a collection of cookbooks or meal kits as they also contains data. A Python library consists of a collection of objects, functions and or variables. We will discuss in detail how we import libraries in Python.

References