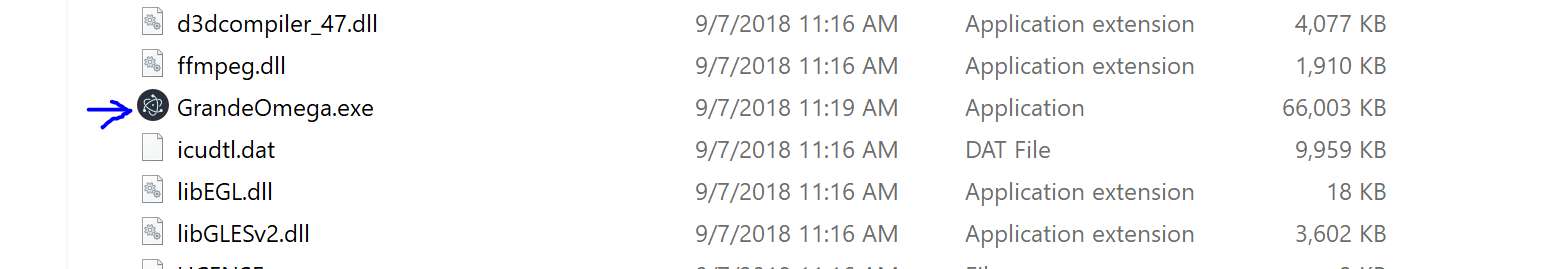
Guide to GrandeOmega (GO)

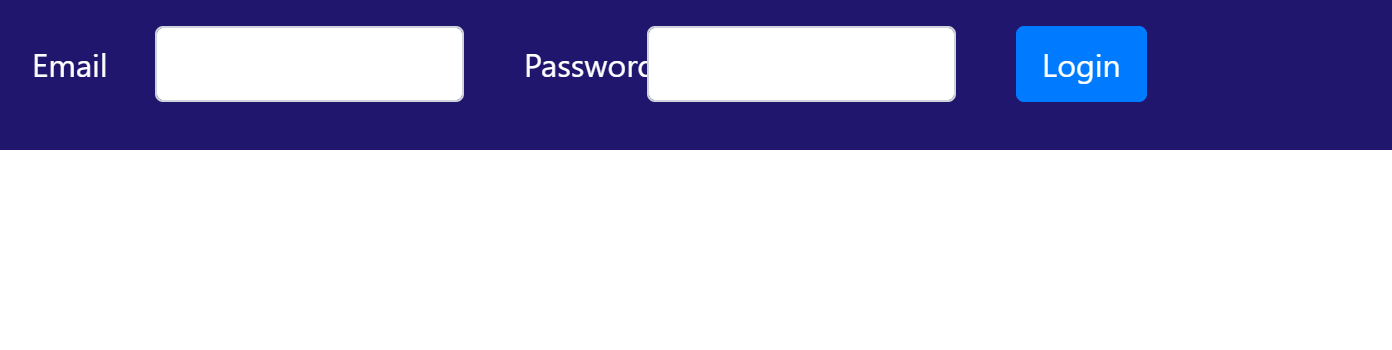
# Installation

* Download node.js (LTS) from: <https://nodejs.org/en/download/>
* Download Python from: <https://www.python.org/downloads/>
  + For Windows users, make sure to have Python added to your environment variables (<https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/> )
* Download the client of GO from: <http://grandeomega.com/go_student_win.7z>
  + Note: the file is archived with 7zip (download it, if needed)
* Unzip the compressed folder downloaded at the previous step
* Execute the **GrandeOmega.exe** file:

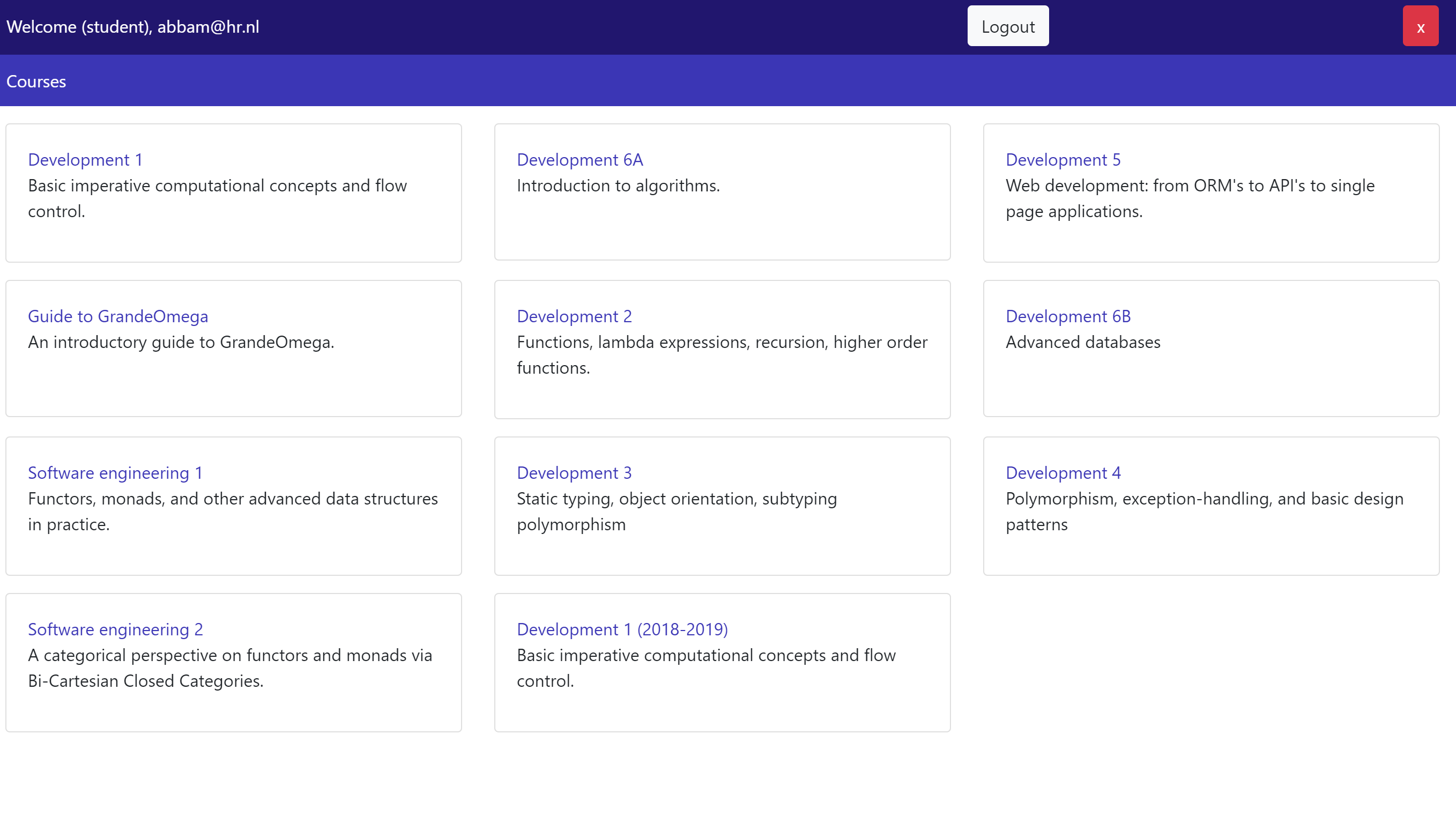


# Use

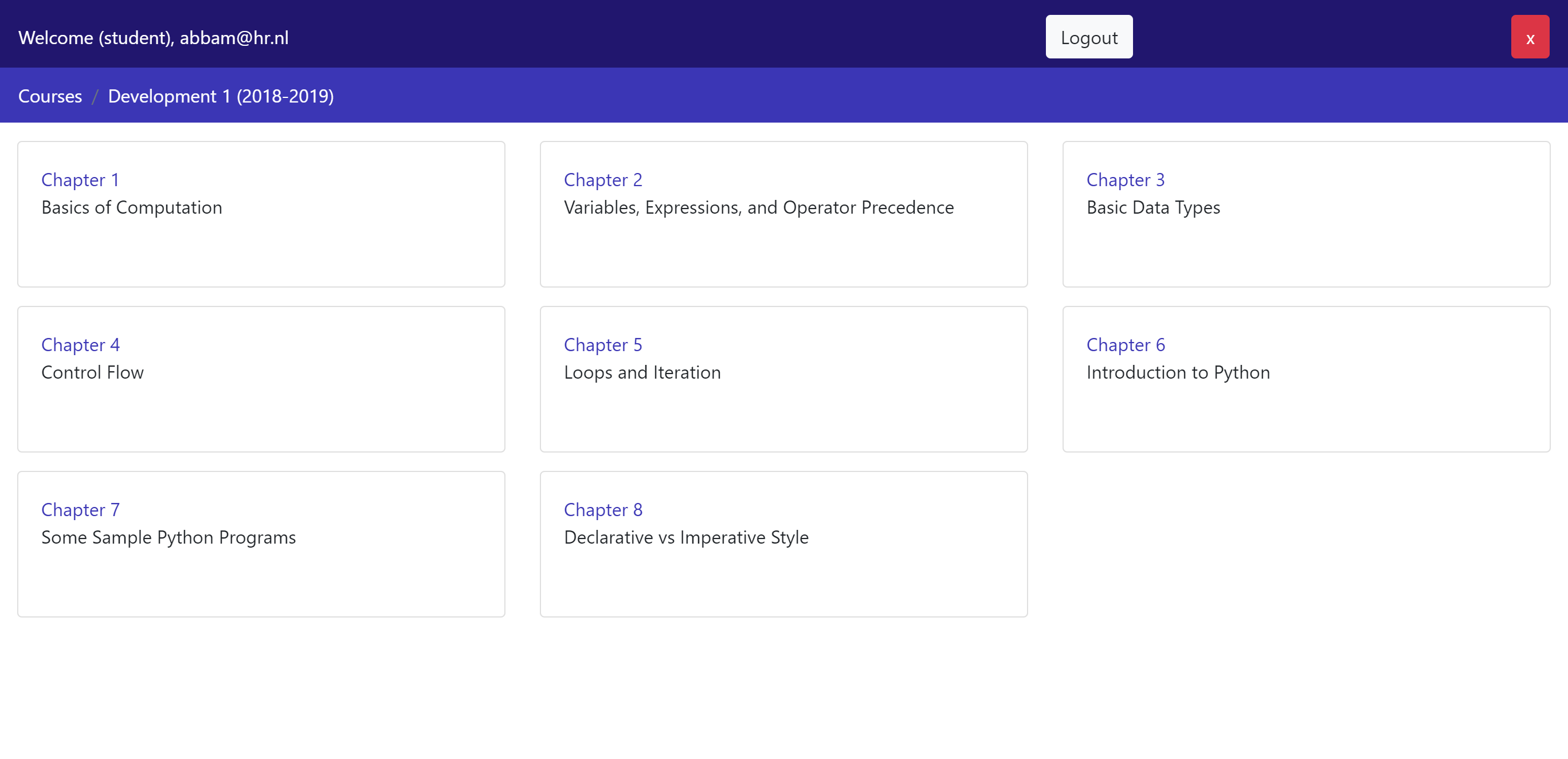
* After the client starts, you need to login with your credentials (you will receive via your student email instructions to get access):



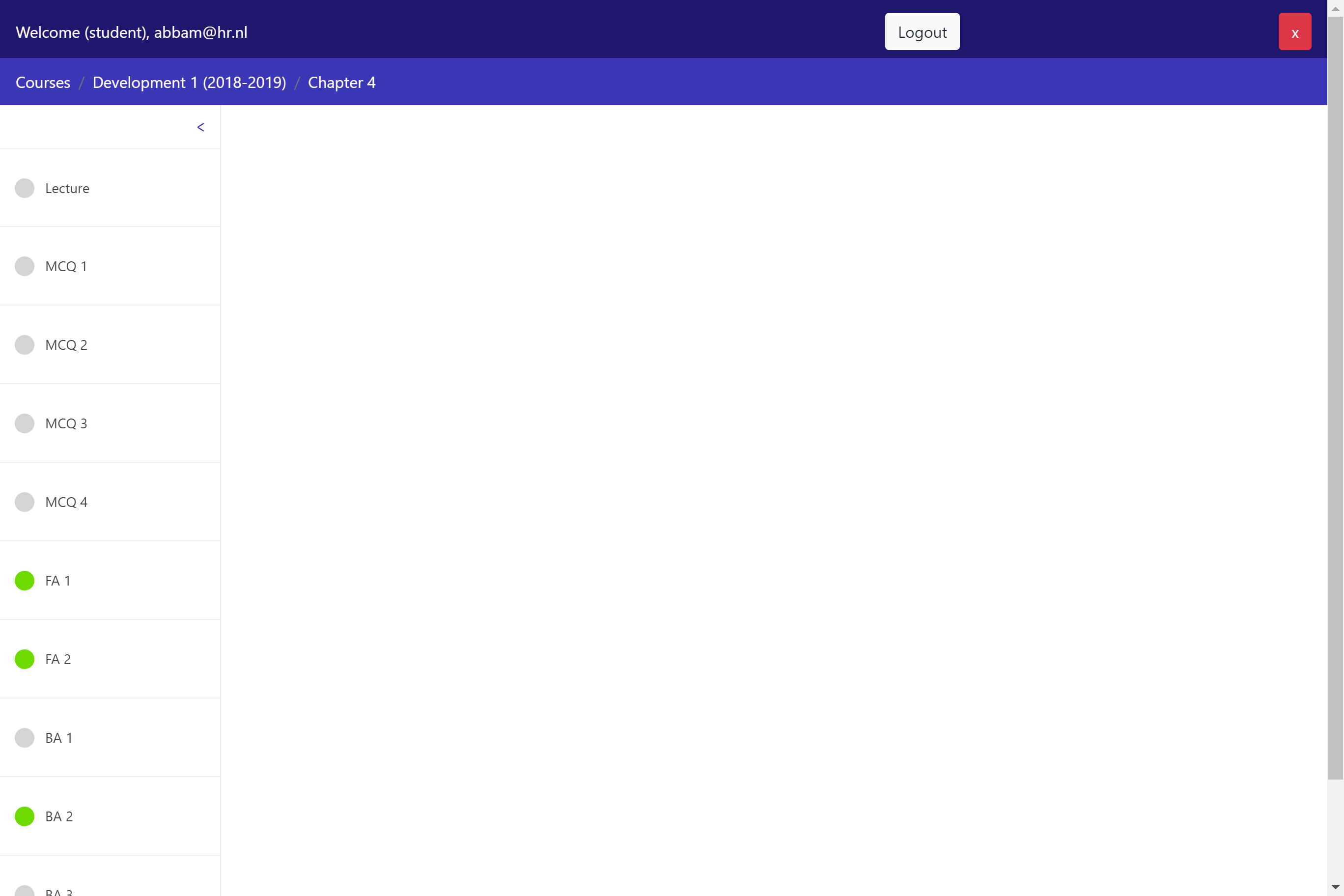
* After having logged in, you will see a screen with the courses you are subscribed to:



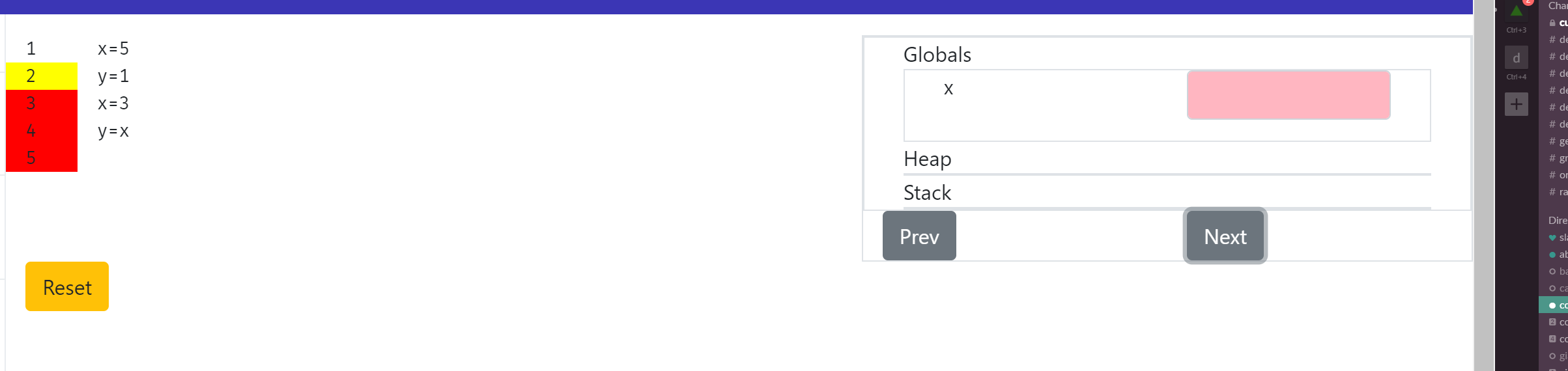
* Clicking on a course, you will see the chapters of materials available for such course:



* Clicking on a chapter, you will see the materials associated to such chapter in a column on the left of the screen. Click on the name of an item to open its associated content.

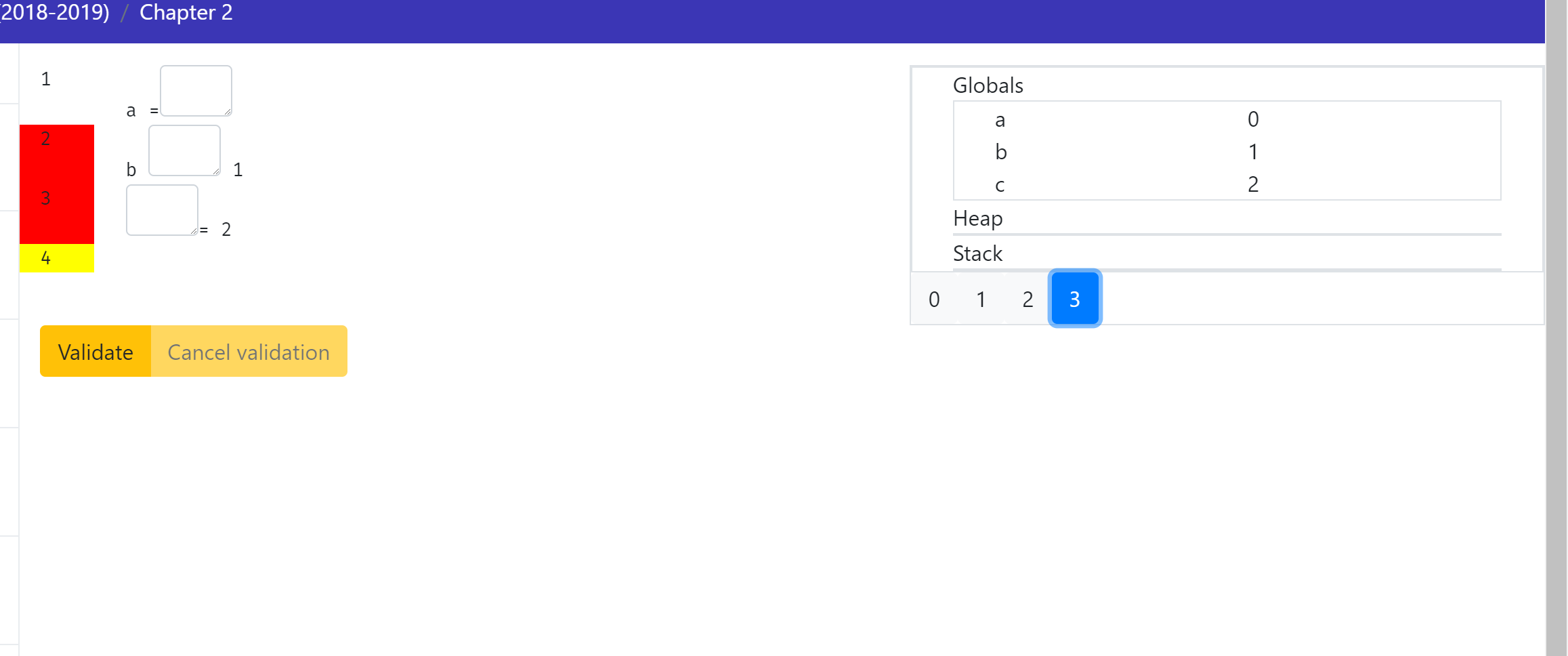


* A single chapter is usually composed by:
  + The reader of the corresponding lecture
  + A series of exercises which are a combination of:
    - Multiple Choice Questions (MCQ)
    - Forward Assignments (FA)
    - Backward Assignments (BA)
* During the practicums, the teachers will show you more in detail how to solve the Forward and Backward assignments.
* In short, a **Forward Assignment** shows you a program and the (sometimes incomplete) state associated to certain steps of the execution of such program (marked with red blocks to the left of the code). To solve a FA, you need to insert the missing values of variables in *all* incomplete states (remember to click “Next” until the last state is reached). For example:



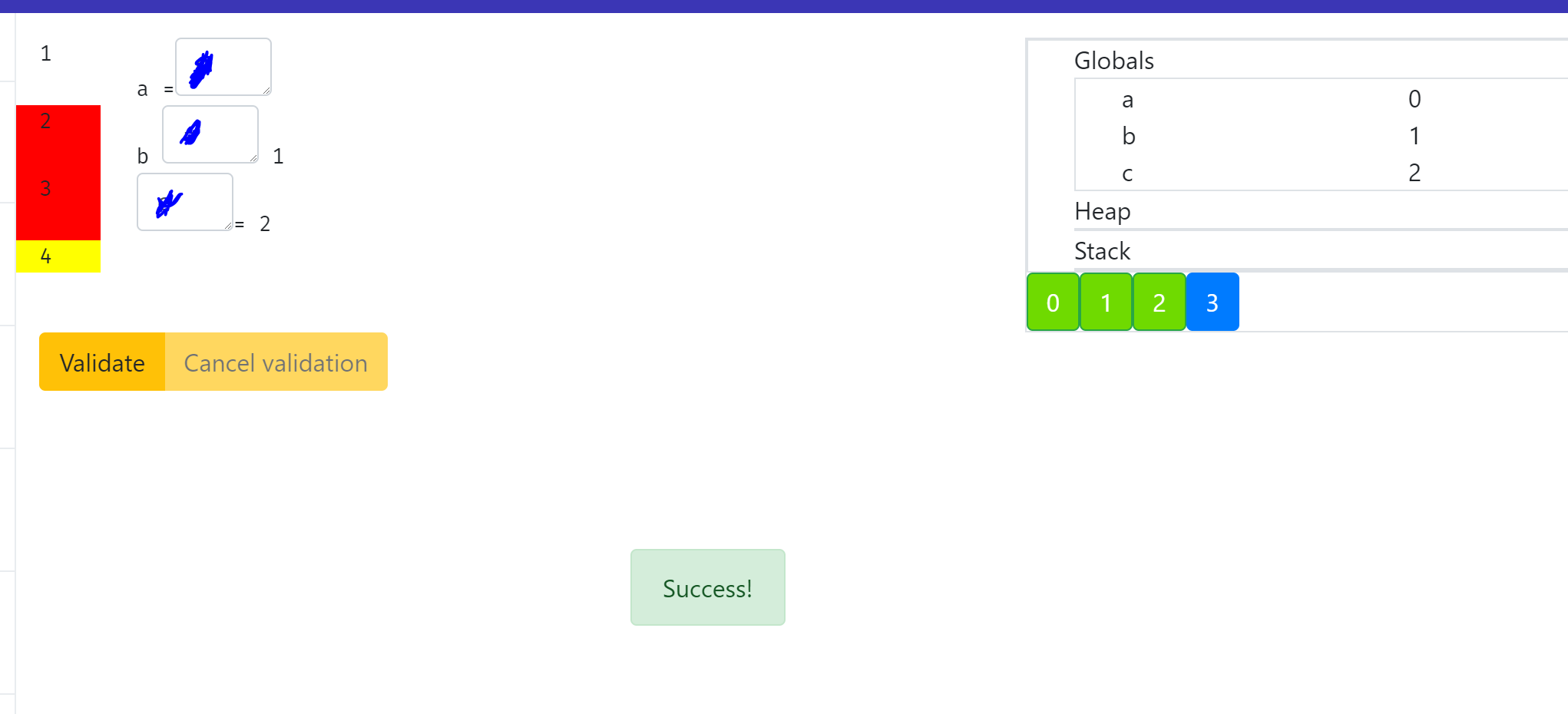
The state on the right (Globals, etc.) corresponds to the state of the program when the line of code marked with a yellow block is about to be executed (in the example above, when line 2 is about to be executed).

A **Backward assignment**, instead, shows you an incomplete program and the states associated to some steps of the execution of the complete program (again, marked by red/yellow blocks to the left of the code). By looking at such states, you should be able to fill in the missing parts of the program. For example:

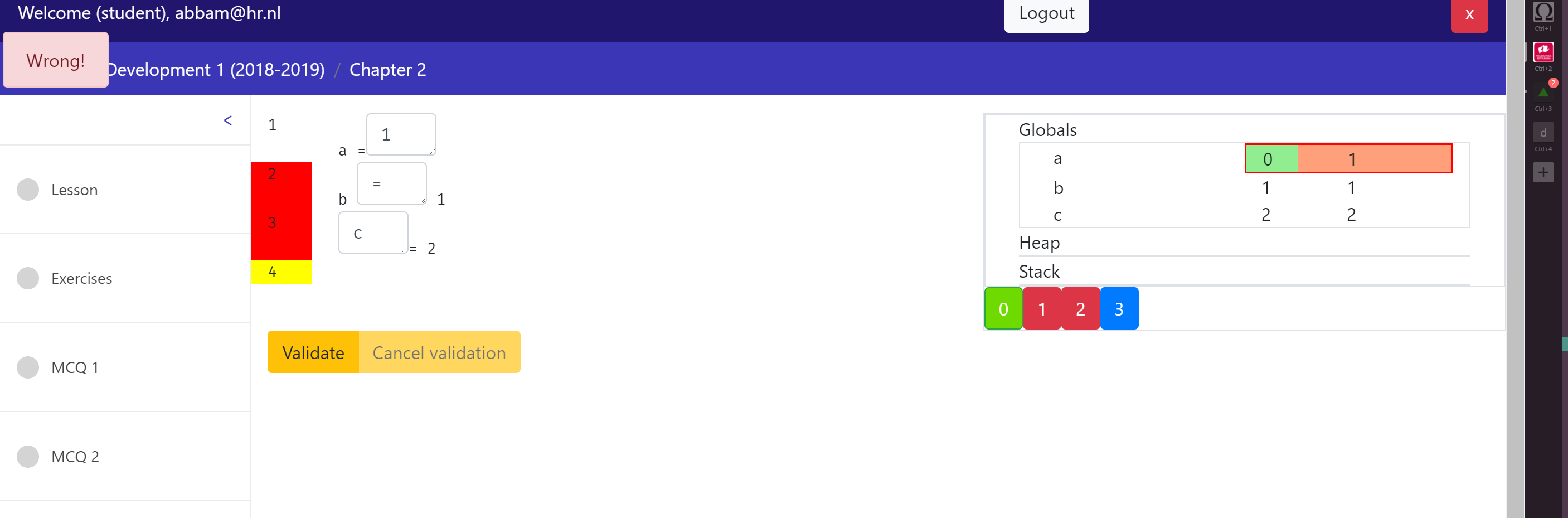


To see if your code solves the BA, click on “Validate” and you will get feedback.

When an assignment is correctly solved (both FA and BA) a “Success!” green message will appear on screen:



Otherwise, a “Wrong!” red message appears (and in BAs the wrong values of your program are shown in red close to the correct ones in green in the state):



The round icon close to the assignment name in the left column also gets such color (orange for incomplete/wrong and green for complete):

