

Handin 1

Due 13 Sep by 18:00 **Points** 100 **Submitting** an external tool

Available after 6 Sep at 18:00

The handins in the course will consist of two parts: 1) a set of short exercises to practice a particular part of the curriculum (e.g. loops), and 2) a project part. In the project part we will consider a specific Data Science case, each week working on the same (or similar) data, and gradually building up a complete analysis of the data.

Note that the tools required to solve the handins will be generally covered in the lectures in the week where the assignment is given. Sometimes, you might have to wait until the Friday lecture before you have all the tools to complete the assignment.

Before we begin, please start by creating a new folder in VS Code for your handins, as described on the slides.

Part 1

This week, we'll look into print statements, if-else statements and loops.

1. Create a new file called `handin1_1.py`. Put the following piece of code in the file.



```
if 2<4
    print "Hello, world!"
```

There are two errors in the code. Fix the errors so that the code prints "Hello, world!" to screen.

2. Create a new file called `handin1_2.py`. In this program, you should create a variable called `message` and assign the string "Hello, world!" to this variable. The program should then print out the length of the string contained in the variable. Finally, it should open an output file called `message.txt`, and print the string to this file. Remember to close the file after you are done writing to it.
3. Create a new file called `handin1_3.py`. In this file, create a variable called `number_str`, and assign the following string to this variable: `"1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20\n"`. We will process this string in a sequence of steps:
 1. Use a string method to remove the newline character at the end of the string and save the result in a variable called `number_str_cleaned`.
 2. Use another string methods to split the string into a list of small strings, where each element is a string containing a single number. Save the result in a variable called `number_str_list`.

3. Write a for-loop that iterates over this list, and prints out the numbers to screen, such that each line contains a single number.

Part 2: Project

The project we will work on throughout the course will be about climate data. In particular, we will be analyzing the `Land_and_Ocean_summary.txt` data file (https://wouterboomsma.github.io/ppds2023/data/Land_and_Ocean_summary.txt  (https://wouterboomsma.github.io/ppds2023/data/Land_and_Ocean_summary.txt)). This file comes from the Berkeley Earth website (<http://berkeleyearth.org/data/>  (<http://berkeleyearth.org/data/>)). It contains global temperature anomalies, telling us how much colder or warmer it is compared to a reference value, which is chosen to be the average over the time period between 1951 and 1980. The header of the file contains further details.

In this first handin, we will start by reading the data into a Python list, and then process the data in a few ways. Start by downloading the `Land_and_Ocean_summary.txt` file and copying it to the directory in which your code resides (i.e. the project folder in VS Code).

1. Create a file called `handin1_project.py`. Inside this file, write code that reads the `Land_and_Ocean_summary.txt` into a list of strings called `list_of_lines`, where each element in the list corresponds to a line in the file. In this first exercise, we will include all lines (including comments).
2. In the same `handin1_project.py` file, write a for-loop that iterates over the `list_of_lines` list. For each line, start by removing the white space at the beginning and end of the line. Skip the line if it is empty or if it contains a comment (i.e. starts with a '%'). Finally, split the string into small pieces as we did in point 2) in the previous exercise, and then use `print` to write out the first entry of each line (i.e. the year values) - but ONLY if the year is larger or equal then 2000.

When you are done, click on the "Load Handin1 in a new window" button below, which will take you to the CodeGrade server. Here, please submit the `handin1_1.py`, `handin1_2.py`, `handin1_3.py` and `handin1_project.py` files. CodeGrade will then automatically check the code for you, and upgrade your grade for the assignment within Absalon. You can submit as many times as you want.

This tool needs to be loaded in a new browser window

Load Handin 1 in a new window

