

Astrophysics III : Stellar and galactic dynamics

Exercises

The following exercises are intended to make you familiar with python 3 and working on a remote server like `noto.epfl.ch`.

Problem 1 : Hello world !

Write python script (file) that prints out “Hello world!” and execute it correctly.

Problem 2 : Functions

Write a function in a python script that takes two integers as an argument, and returns their sum as the result. Then the script should execute the function and print out the result to test that it’s working correctly.

Problem 3 : Conditionals

Write a new function that takes two integers as an argument. It should call the function you wrote in problem 2 to compute the sum of the two arguments. If the result is greater than 100, the new function should print ‘The result is > 100’ to the screen. Otherwise, it should print ‘The result is < 100’. Then the script should execute the new function to test that it’s working correctly.

Problem 4 : Writing and importing modules

Move the two functions that you’ve written in exercise 2 and 3 into a module and name it `my_first_module.py`. Also define two variables, `x_module = 20` and `y_module = 35`, in the module. Then write a script that imports your new module and then calls the function of the module that tells you whether the sum of two variables is smaller or greater than 100 with the variables `x_module` and `y_module` as arguments.

Hint : Recall that for python to find the module files you write, they need to be in specific directories where python is looking for them (not covered in this course), or in the same directory as the file that you’re trying to run (what we are trying to do here).

For the exercises below, feel free to use a jupyter notebook instead of writing a script, if you prefer to do so.

Problem 5 : Loops and lists

Write a python script that creates 2 empty lists named `even` and `odd`. Then it should run through a loop from 0 to 100 (included) and add all the odd integers to the list `odd` and all the even integers to the list `even`, unless the integer is divisible by 5. In that case, it shouldn’t be added to either list.

Hint : The modulo operator `%` returns the remainder of an integer division, e.g.

```
>>> 3 % 2
1
```

Problem 6 : Numpy essentials

Write a script that has a numpy array named `array1` containing the values

```
1 3 5 -2 -8 20 43 54 -9 0
```

Then use a numpy array method to find the following values :

1. The minimum value in the array
2. The maximum value in the array
3. The mean value of the array
4. The sum of the array
5. The standard deviation of the array
6. The position of the maximal value in the array
7. The position of the minimal value in the array

and print out the results.

Now create a second numpy array named `array2` that contains the following elements :

```
7. 16.0324 8.29342 9.234 724.243 712.321 74293.783 123.13 523.423 423.41
```

and compute the following using it and numpy functions :

1. the square root
2. the natural logarithm \ln
3. the base 10 logarithm \log_{10}
4. the exponential function e^{array2}
5. the array to the power of 2.3
6. the sine of the array, $\sin(\text{array2})$

Finally, using the arrays `array1` and `array2`, do the following :

1. add up all elements of both arrays individually and store them in a new array.
2. multiply all elements of both arrays individually and store them in a new array.
3. Create a new array `array3` that contains the elements of `array2` multiplied by 3.4. Then multiply the first two elements of `array3` with the first two elements of `array1`. Then subtract the first two elements of `array1` from the **last** two elements of `array3`.

Problem 7 : Matplotlib essentials

Using the `matplotlib` and `numpy` modules, draw a plot of the function

$$f(x) = 18 \sin(23x) \exp(-x^2) \quad \text{for } x \in [0, \pi) \quad (1)$$

Problem 8 : Matplotlib advanced

Using the `matplotlib` and `numpy` modules, draw a plot of the following functions **in the same plot** :

$$f(x) = -3\ln(x) + 5 \quad (2)$$

$$g(x) = 2\sin\left(x - \frac{\pi}{2}\right) \quad (3)$$

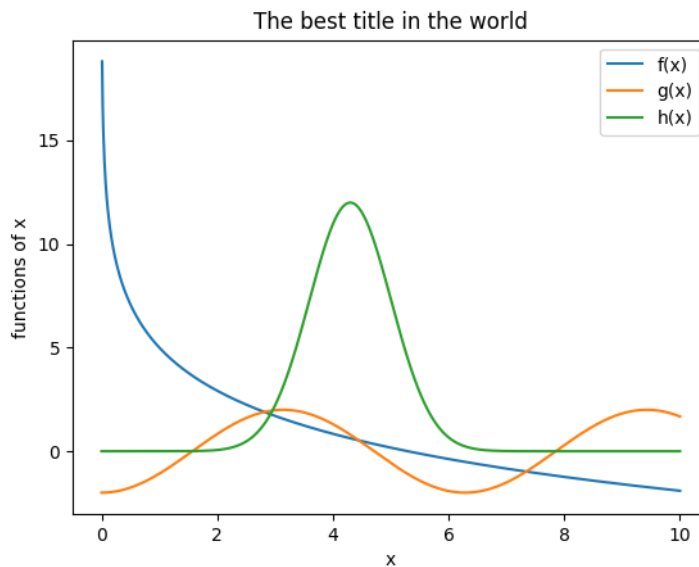
$$h(x) = 12\exp(-(x - 4.3)^2) \quad (4)$$

for $x \in [0, 10)$.

Then :

- Label each curve individually and add a legend
- Label each axis individually and add a title
- directly save the figure as `problem_8_plot.png`

The end result should look something like this :



Optional assignments :

- Instead of plotting all functions in the same plot, plot them in 3 individual subplots in the same figure instead
- Make the $f(x)$ line be dashed, $g(x)$ be dotted, and $h(x)$ be dots only
- Draw a grid for each subplot
- Suppose $h(x)$ are measurements that have an error of $\pm 10\%$ at every point of measurement. Draw the error bars.

The end result should look something like this :

