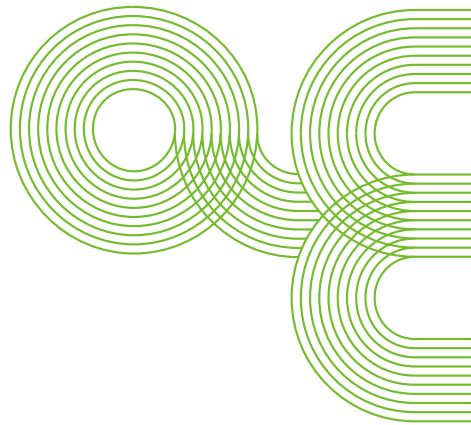


Universidade de Vigo

Computer Science / Informatics – Exercises



Escola de Enxeñaría Aeronáutica e do Espazo
Pavillón Manuel Martínez-Risco
Campus universitario
32004 Ourense

<http://aero.uvigo.es>
<mailto:aero.info@uvigo.es>



Reference: 1.0
Document: practicas-inf
Date: 11/10/2018
Pages: 8

Contents

1	Week no. 1	3
2	Week no. 2	4
3	Week no. 3	5
4	Week no. 4	6
5	Week no. 5	7

1 Week no. 1

Objectives: Work with python. Get to know simple data types. Perform simple computations with simple data types. Enter and visualize data using keyboard and console. Use strings of characters.

1. Run IDLE (Python 3.5), type the line `print("hello world")`. Save the file as `E1_exercise1.py`, run the program.
2. Try the following program that visualizes some data types of python:

```
age = 20
price = 49.5
dni = "12345678Z"
message = "I've told him 'hi'"
fellowship = True
complex = 1.3 + 2.5j
print(age, price, dni, message fellowship, complex)
```

Modify the previous program changing the values of some of the variables and running it again on the console. Check that it is possible to change the type of a variable by simply assigning a value of a different type.

3. Write a program that performs the following operations:
 - (a) Compute the mean value of two floating point numbers.
 - (b) Show this mean value on the console.
 - (c) To compute the final price of an article, assign to a variable a base price (neto) of, for instance, 100 euros, and to another variable the tax of 21%, compute the final price of the article as `base + tax`.
 - (d) Show the price on the terminal.
 - (e) Assign to a variable an age, and in another variable store whether the age stands for an adult or not.
 - (f) Show the result on the terminal.
 - (g) Compute the rest that remains by an integer division, show it on the terminal.
 - (h) Compute some exponential of a value and visualize its value.
4. The following program asks the user to enter the radius of a circle using the keyboard and then calculates its area:

```
r = float(input("Circle radius: "))
a = 3.1416*r**2
print("Area: " + str(a))
```

Write a similar program that asks for the base price (without taxes) of a ticket for the cinema, and then calculates the prices applying 10% and 21% VAT. Estimate the savings due to the reduction of tax rates.

2 Week no. 2

Learning goals: Continue with the previous week's learning goals.

1. Complete the following program to obtain the letter from a dni number:

```
letterset_dni = "TRWAGMYFPDXBNJZSQVHLCKE"  
dni = .....  
posicion = dni%23  
letter_dni = .....  
NIF = .....  
print("NIF = ",NIF)
```

2. Write a python program that asks for a value in degrees Fahrenheit and shows the value converted to Celsius. You should use the following formula:
$$\text{degrees_celsius} = (\text{degrees_fahrenheit} - 32) / 1.8$$
3. Write a python program that reads-in a first name, a name, and a dni of a person and outputs the data on the terminal.
4. Write a python program that reads-in a name and surname and generates (and visualizes) the corresponding email address. The address is built by the first letter of the name followed by the surname and ended by @alumnos.uvigo.es.
5. Write a python program that reads-in the coordinates x and y of a point and visualizes them in the format (x,y) on screen.
6. Write a python program that computes the distance between two points.
7. These lines will generate a integer random number between 0 and 9:

```
import random  
num = random.randint(0,9)
```

Write a python program that, after generating a random number, asks the user to input a number using the keyboard. The program must tell if the user has guessed the number or, if he didn't, tell if the secret number is smaller or greater than the number typed by the user.

3 Week no. 3

Learning goals: Implement simple algorithms using flow control and loops.

1. Write a program that reads-in a numerical mark and shows its textual value on the terminal.
2. Write a program that reads-in a numerical value and visualizes whether the value is odd or even.
3. Write a program that visualizes which one of two input numbers is the larger one. Extend the program to work for three numbers.
4. Write a small calculator, i.e., a program that accepts three parameters: two numbers and a symbol. If the symbol is a + the values are added, if it is a - the values are subtracted, likewise *, /, and ^ (exponential).
5. Write a program that computes the mean value of a series of numbers given by the user. Think about a method how to determine that the user wants the program to stop accepting numbers.

4 Week no. 4

Learning goals: Work with simple algorithms including flow control and loops. First functions.

1. Write a program that asks for a number and prints the squares of the numbers ranging from one to the given value.
2. Write a program that generates ten random numbers between 1 and 100 and prints the greatest of them.
3. Write a program that asks for an integer number and prints its multiplication table.
4. Modify the previous program so that the output is nicely formatted in columns. This can be achieved using the `format` functionality. To see how this works, try the following program:

```
print("{} y {} are even numbers".format(2,4))
print("{:4d} y {:4d} are even numbers".format(2,4))
print("{:04d} y {:04d} are even numbers".format(2,4))
```

5. Write a program that shows whether a given number is prime or not.
6. The program included below uses a function to calculate the final mark from three partial marks using three weighting factors.

```
def calculate_mark(test1, test2, lab):
    return test1*0.35 + test2*0.35 + lab*0.3

print(calculate_mark(7.5, 8.5, 7))
print(calculate_mark(4.7, 7.2, 6.8))
```

Modify this program so that the weighting factors are also passed as input arguments to the function.

5 Week no. 5

Learning goals: Using functions in python. Reinforcement of loops. More input/output.

1. Observe the following program. It uses two functions: one to encapsulate the input operation, and another one to calculate and print the grade.

```
def ask_for_grades():
    exam1 = float(input("Grade 1. exam: "))
    exam2 = float(input("Grade 2. exam: "))
    exercises = float(input("Grade exercises: "))
    return exam1, exam2, exercises

def show_grades(test1, test2, exercises):
    final = test1*0.35 + test2*0.35 + exercises*0.3
    print("Grade final: {:.2f}".format(final))

grade1, grade2, grade3 = ask_for_grades()
show_grades(grade1, grade2, grade3)
```

2. The following program asks for new input and visualizes the grades until the user wishes to finish the program.

```
def ask_for_grades():
    exam1 = float(input("Grade 1ª exam: "))
    exam2 = float(input("Grade 2ª exam: "))
    exercises = float(input("Grade exercises: "))
    return exam1, exam2, exercises

def show_grades(test1, test2, exercises):
    final = test1*0.35 + test2*0.35 + exercises*0.3
    print("Final grade: {:.2f}".format(final))

repeat = "yes"
while (repeat != "no"):
    grade1, grade2, grade3 = ask_for_grades()
    show_grades(grade1, grade2, grade3)
    repeat = input("keep going? (yes/no) ")
    repeat = repeat.lower()
else:
    print("\nEnd of program")
```

3. Modify the previous program such that any negative input terminates implicitly

the input loop (clearly, this last entry should not participate in the average).

4. Write a function to compute the Collatz sequence of a positive integer number a_0 , where for such a sequence we have $a_{n+1} = 1/2 \cdot a_n$, if a_n is even, and $a_{n+1} = 3 \cdot a_n + 1$, if a_n is odd. The sequence terminates whenever we arrive at the number 1. For instance, for $a_0 = 3$ we obtain the sequence: 3, 10, 5, 16, 8, 4, 2, 1.

Note: the Collatz conjecture is that no matter what value of n , the sequence will always reach 1. This conjecture has not been proven yet¹.

5. We now that in Spain, the height of young men follows roughly a normal distribution (mean 172 cm and standard deviation 8.1 cm). We want to calculate the percentage of the male population that is taller than 185 cm. To obtain this percentage we will use the Monte Carlo method: we create data points following the given distribution, and we count how many points lie above the specified height (185 cm). Write a function to compute this percentage.

Hint 1: we will consider that 10000 points is a big enough set of data to have an accurate estimation.

Hint 2: the random library provides a function called `gauss` that can be used to generate the data. See python documentation²

6. Modify the previous program so that the mean, standard deviation and height are asked to be entered by the user.

¹https://en.wikipedia.org/wiki/Collatz_conjecture

²<https://docs.python.org/3/library/random.html>