

Lab 1

COMP9021, Session 1, 2016

If you have not done so already, create two sequences of directories,

- `~/COMP9021/Labs` and
- `~/COMP9021/Lectures`.

One way to do this is to, in your home directory, execute the Unix command

```
mkdir -p COMP9021/Labs COMP9021/Lectures
```

Download from the course's website the compressed archives

- `Lab_1.tar.gz` and
- `Lecture_1.tar.gz`,

and save them in the appropriate directories. Decompress the archives. One way to do this is to, in the directory `~/COMP9021/Labs`, execute the Unix command

```
tar xzf Lab_1.tar.gz
```

and in the directory `~/COMP9021/Lectures`, execute the Unix command

```
tar xzf Lecture_1.tar.gz
```

This will create a subdirectory `Lab_1` of `~/COMP9021/Labs` and a subdirectory `Lecture_1` of `~/COMP9021/Lectures`, each of which will contain the provided material. The archives can then be deleted. One way to do this is, in the directory `~/COMP9021`, execute the Unix command

```
rm Labs/*gz Lectures/*gz
```

You will do the same for the following labs and lectures; recall then to refer to these instructions if needed.

1 Running python code

Experiment with the different ways of running python code as described in the pdf document `Running python code.pdf`, which is part of the material for the first lecture.

2 Text-based programs

Run and study the program named `fahrenheit_to_celsius.py`. Then write a program named `celsius_to_fahrenheit.py` that displays a conversion table from Celsius degrees to Fahrenheit degrees, with the former ranging from 0 to 100 in steps of 10.

Run and study the program `max_in_list.py`. Then write a program `span.py` that generates a list of 10 random integers between -50 and 50 (included), prints out the list, computes the difference between the largest and smallest values in the list, and prints it out. Here is a possible interaction:

```
$ python3 span.py
The list is: [-32, 39, 18, -17, 40, 18, 30, 44, -42, 23]
    The maximum difference between largest and smallest values in this list is: 86
Confirming with builtin operations: 86
$ python3 span.py
The list is: [3, 1, 47, -47, 10, 42, -23, 42, 24, 6]
    The maximum difference between largest and smallest values in this list is: 94
Confirming with builtin operations: 94
```

Run and study the program named `intervals.py`. Then write a program named `modulo_4.py` that prompts the user for a strictly positive integer N , generates a list of N random integers between 0 and 99 (included), prints out the list, computes the number of elements equal to 0, 1, 2 3 modulo 4, and prints these numbers out. Here is a possible interaction:

```
$ python3 modulo_4.py
How many elements do you want to generate? 1
The list is: [40]
    There is 1 element equal to 0 modulo 4.
    There is 0 element equal to 1 modulo 4.
    There is 0 element equal to 2 modulo 4.
    There is 0 element equal to 3 modulo 4.
$ python3 modulo_4.py
How many elements do you want to generate? 3
The list is: [79, 34, 6]
    There is 0 element equal to 0 modulo 4.
    There is 0 element equal to 1 modulo 4.
    There are 2 elements equal to 2 modulo 4.
    There is 1 element equal to 3 modulo 4.
$ python3 modulo_4.py
How many elements do you want to generate? 14
The list is: [49, 0, 15, 62, 45, 51, 51, 67, 42, 28, 37, 63, 29, 63]
    There are 2 elements equal to 0 modulo 4.
    There are 4 elements equal to 1 modulo 4.
    There are 2 elements equal to 2 modulo 4.
    There are 6 elements equal to 3 modulo 4.
```

Write a program named `mean_median_standard_deviation.py` that prompts the user for a strictly positive integer N , generates a list of N random integers between -50 and 50 (included), prints out the list, computes the mean, the median and the standard deviation, and prints them out. Here is a possible interaction:

```
$ python3 mean_median_standard_deviation.py
How many elements do you want to generate? 1
The list is: [29]
    The mean is 29.00.
    The median is 29.00.
    The standard deviation is 0.00.
Confirming with functions from the statistics module:
    The mean is 29.00.
    The median is 29.00.
    The standard deviation is 0.00.
$ python3 mean_median_standard_deviation.py
How many elements do you want to generate? 8
The list is: [-37, -12, 18, -15, -35, -33, -45, 10]
    The mean is -18.62.
    The median is -24.00.
    The standard deviation is 21.58.
Confirming with functions from the statistics module:
    The mean is -18.62.
    The median is -24.00.
    The standard deviation is 21.58.
$ python3 mean_median_standard_deviation.py
How many elements do you want to generate? 13
The list is: [-15, 23, 29, 5, -1, 45, 22, 1, 47, -30, 32, -50, -38]
    The mean is 5.38.
    The median is 5.00.
    The standard deviation is 30.17.
Confirming with functions from the statistics module:
    The mean is 5.38.
    The median is 5.00.
    The standard deviation is 30.17.
```

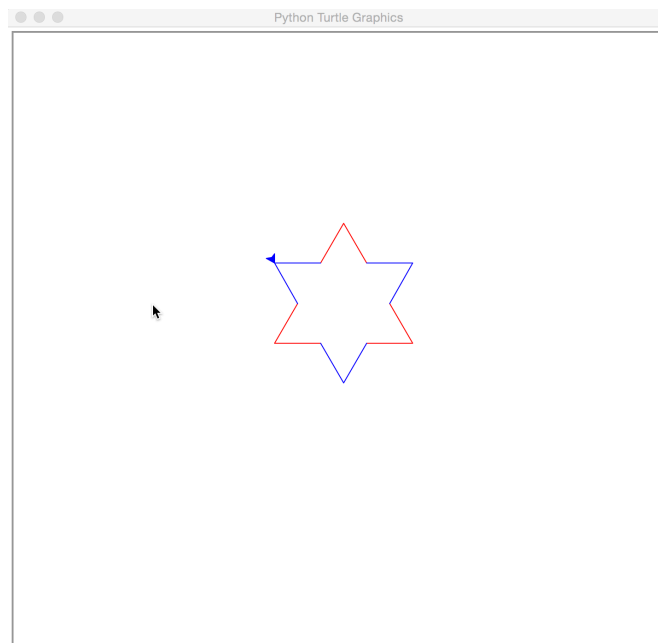
3 Drawing pictures with turtle

For the following exercises, you can refer to the [Turtle graphics](#) documentation, but you can complete the exercises by just studying the sample programs.

3.1 An hexagram

Run and study the program `dodecagrams.py`.

Then write a program `hexagram.py` that draws an hexagram that is centred horizontally in the window that displays it, with the colour of the tips alternating red and blue:

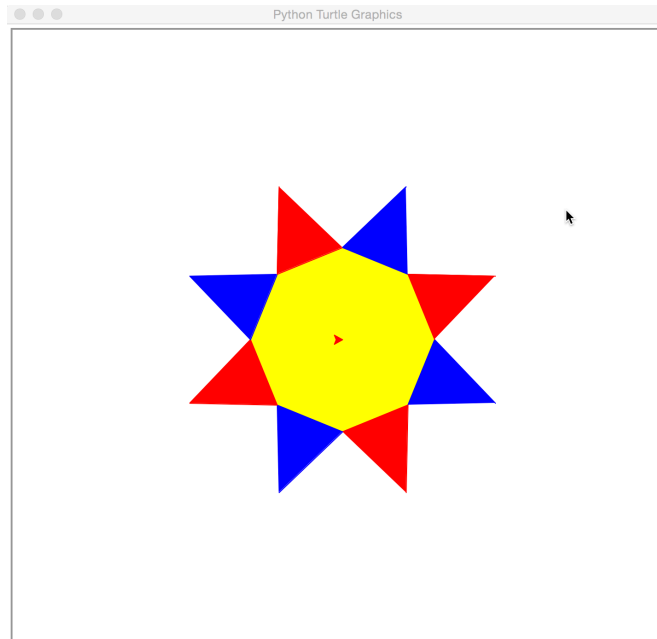


You are encouraged to draw the red part and then the blue part of the star.

3.2 An octagram

Run and study the program `dodecagon.py`.

Then write a program `octagram.py` that draws an octagram, the inscribed octagon being coloured yellow, and the colour of the triangles alternating red and blue:



You can set the distance from the centre to an edge of the inscribed octagon to 100 pixels, and the distance from the centre to the tip of a triangle to 180 pixels.