Lab 1

COMP9021, Session 2, 2017

1 ☞ Temperature conversion tables

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.

2 ☞ Max element and span in a list

Run and study the program `max_in_list.py`. Then write a program `span.py` that prompts the user for a seed for the random number generator, and for a strictly positive number, `nb_of_elements`, generates a list of `nb_of_elements` random integers between 0 and 99, prints out the list, computes the difference between the largest and smallest values in the list without using the builtins `min()` and `max()`, prints it out, and check that the result is correct using the builtins. Here is a possible interaction:

```
$ python span.py
Input a seed for the random number generator: 0
How many elements do you want to generate? 8

The list is: [49, 97, 53, 5, 33, 65, 62, 51]

The maximum difference between largest and smallest values in this list is: 92
Confirming with builtin operations: 92
$ python span.py
Input a seed for the random number generator: 1
How many elements do you want to generate? 12

The list is: [17, 72, 97, 8, 32, 15, 63, 97, 57, 60, 83, 48]

The maximum difference between largest and smallest values in this list is: 89
Confirming with builtin operations: 89
```

3 ☞ Classifying elements in a list

The operators `/`, `//` and `%` are used for floating point division, integer division, and remainder, respectively. Run and study the program named `modulo_4.py` . Then write a program named `intervals.py` that pPrompts the user for a strictly positive integer, `nb_of_elements`, generates a list of `nb_of_elements` random integers between 0 and 19, prints out the list, computes the number of elements strictly less 5, 10, 15 and 20, and prints those out. Here is a possible interaction:

```
$ python intervals.py
Input a seed for the random number generator: 0
How many elements do you want to generate? 1

The list is: [12]

There is no element between 0 and 4.
There is no element between 5 and 9.
There is 1 element between 10 and 14.
There is no element between 15 and 19.
$ python intervals.py
Input a seed for the random number generator: 1
How many elements do you want to generate? 3

The list is: [4, 18, 2]

There are 2 elements between 0 and 4.
There is no element between 5 and 9.
There is no element between 10 and 14.
There is 1 element between 15 and 19.
$ python intervals.py
Input a seed for the random number generator: 2
How many elements do you want to generate? 14

The list is: [1, 2, 2, 11, 5, 9, 8, 19, 6, 19, 1, 18, 5, 13]

There are 4 elements between 0 and 4.
There are 5 elements between 5 and 9.
There are 2 elements between 10 and 14.
There are 3 elements between 15 and 19.
```

mean平均数
median中位数
standard deviation标准差：
方差的正平方根

4 ☞ Statistics on numbers in a list

Write a program named `mean_median_standard_deviation.py` that prompts the user for a strictly positive integer, `nb_of_elements`, generates a list of `nb_of_elements` random integers between -50 and 50, prints out the list, computes the mean, the median and the standard deviation in two ways, that is, using or not the functions from the statistics module, and prints them out. Here is a possible interaction:

```
$ python mean_median_standard_deviation.py
Input a seed for the random number generator: 0
How many elements do you want to generate? 1

The list is: [-1]

The mean is -1.00.
The median is -1.00.
The standard deviation is 0.00.

Confirming with functions from the statistics module:
The mean is -1.00.
The median is -1.00.
The standard deviation is 0.00.
$ python mean_median_standard_deviation.py
Input a seed for the random number generator: 1
How many elements do you want to generate? 8

The list is: [-33, 22, 47, -42, -18, -35, 13, 47]

The mean is 0.12.
The median is -2.50.
The standard deviation is 34.41.

Confirming with functions from the statistics module:
The mean is 0.12.
The median is -2.50.
The standard deviation is 34.41.
$ python mean_median_standard_deviation.py
Input a seed for the random number generator: 2
How many elements do you want to generate? 13

The list is: [-43, -39, -40, -4, -29, 44, 35, -11, -18, 27, -23, 27, -46]

The mean is -9.23.
The median is -18.00.
The standard deviation is 30.92.

Confirming with functions from the statistics module:
The mean is -9.23.
The median is -18.00.
The standard deviation is 30.92.
```

To compute the median, the easiest way is to first sort the list with the builtin `sort()` method.

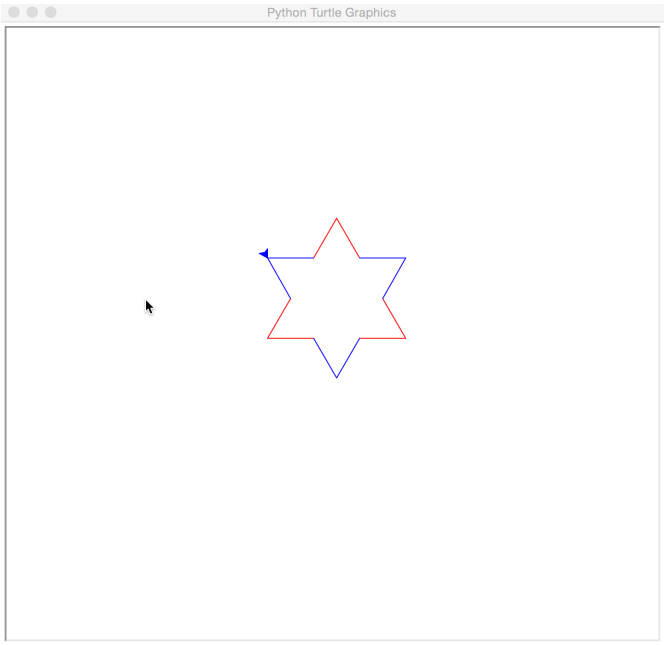
5 Drawing pictures with `turtle` (optional practice)

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

5.1 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:

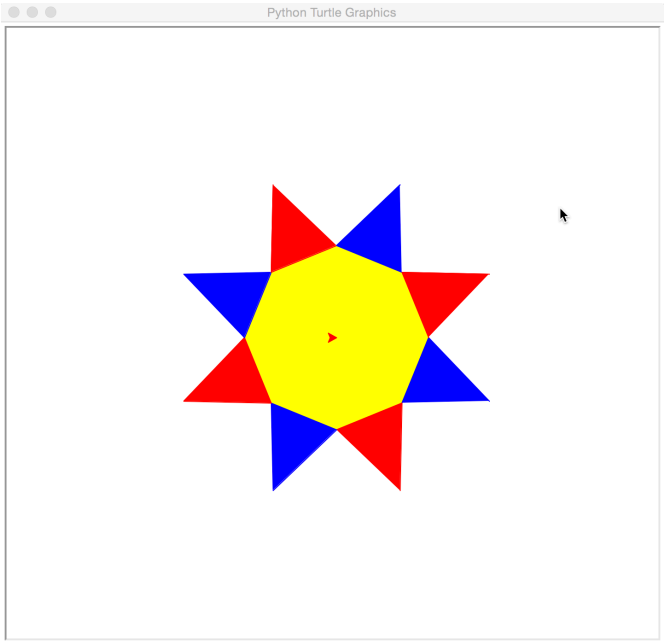


You are encouraged to draw the red part and then the blue part of the star.

5.2 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.