

Exercise set 8

Using Python extra modules (colorama, Pillow etc.)

Note: Check out the guide on Moodle on how to install extra modules in PyCharm!

1. Colorama –extra module – coloring console output, exercise 1

Create an application that asks the user to provide the birth years of five different people. **Save all birth years into a single list** either by using a loop or other methods. After this, loop through the birth years, and determine each person's age based on the birth year.

Each given age has to be between 0 and 125.

If the age is within the accepted limits, print "**age OK!**" in green text. If the age is not within the accepted limits, print "**incorrect age.**" in red text.

Note: You can get the current year by using the **datetime**-module and calculate the age by subtracting the birth year from the current year.

Note: You need two loops in this application, the first one asks the birth years from user (input()) and adds them to a list (append()), and the second loop just goes through the birth year list and prints whether the age is okay or not.

After all birth years have been processed, print the text "**All done!**" in normal text.

Example of the application running:

```
Give the birth year of person 1:
1977
Give the birth year of person 2:
2002
Give the birth year of person 3:
1856
Give the birth year of person 4:
1988
Give the birth year of person 5:
2154

Let's process all birth years...
44 years old, age OK!
19 years old, age OK!
165 years old, incorrect age.
33 years old, age OK!
-133 years old, incorrect age.
All done!
```



Filename of the exercise = *exercise8_1.py*

Typical code amount : **12-18 lines** (empty lines/comments not included)

2. Colorama –extra module – coloring console output, exercise 2

Create an application where the user has to guess a random number generated by Python between 1 and 20. **Only generate the number once in the application (i.e. don't create the random number inside the loop, but before that!).**

If the user's guess is smaller than the random number, print the text "Too low." with blue background. If the user's guess is higher than the random number, print the text "Too high." with red background. If the user guesses the random number correctly, print the text "CONGRATULATIONS!" with green background. The application should keep on asking a new number as long as the user guesses it correctly.

Example of the application running:

```
Guess the number (1-20):  
15  
Too high.  
  
Guess the number (1-20):  
10  
Too low.  
  
Guess the number (1-20):  
13  
CONGRATULATIONS!!
```



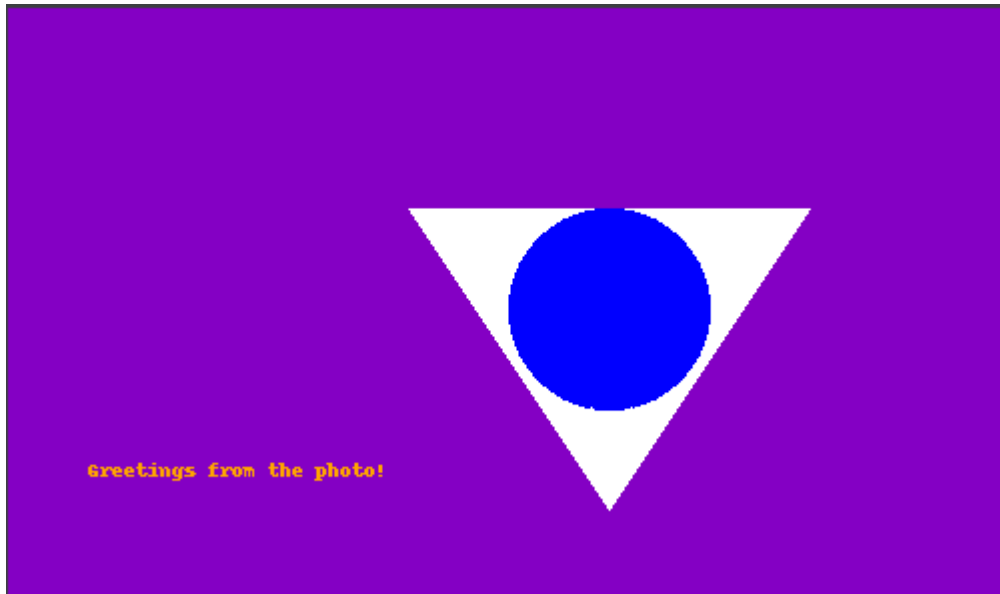
Filename of the exercise = *exercise8_2.py*

Typical code amount : **12-20 lines** (empty lines/comments not included)

3. Creating images by using the Pillow-module

By using Python, create a picture that has the size of 500 x 300 and has purple as the background color. Ask the user for some text, and draw the text into the lower left corner in orange color. Draw also a white triangle that contains a blue circle inside of it.

Example of the application running:



Filename of the exercise = *exercise8_3.py*

Typical code amount : **8-16 lines** (empty lines/comments not included)

The code amount can be even more, if extra tasks are done!

You'll find small examples in Moodle on how to use Pillow to create different shapes!

Note: Pillow accepts color in RGB-format (Red, Green, Blue).

You can use this tool to get RGB-values of any color you want.

<https://yuilibrary.com/yui/docs/color/rgb-slider.html>

Remember: the coordinate (0, 0) in this case refers to **top left corner**. Since the size of the image is 500 x 300, then the bottom right corner will be in the coordinate (500, 300). Bottom left corner will be (0, 300). **(x, y)**

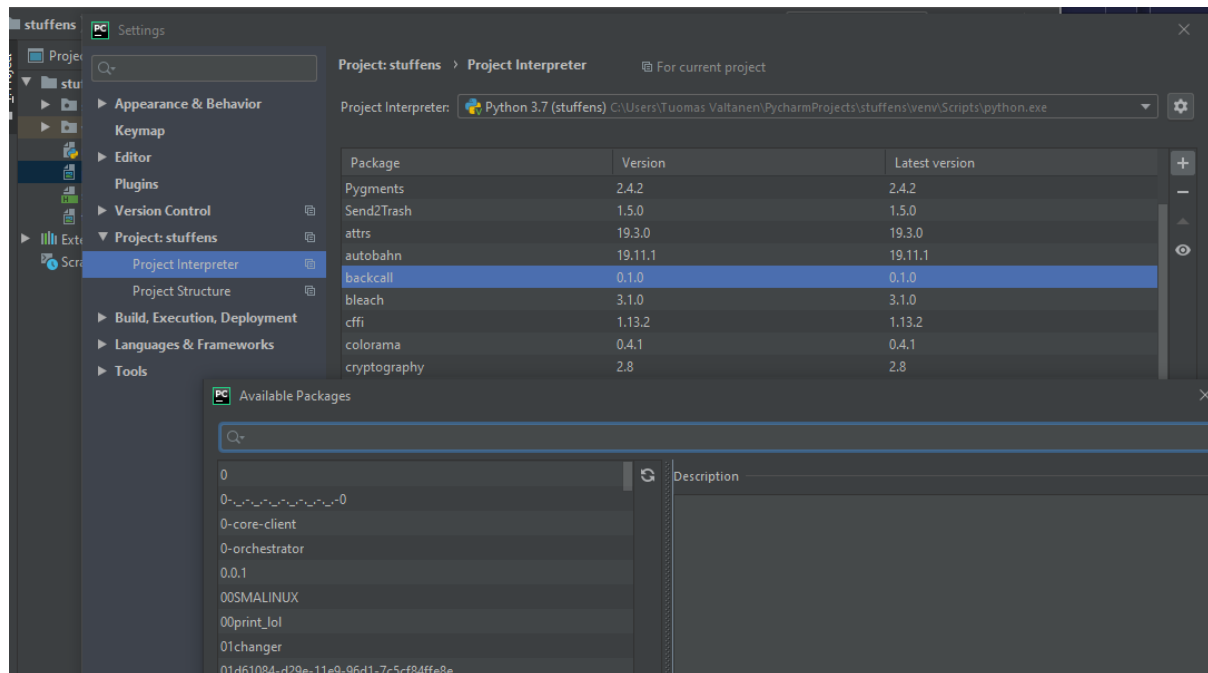
Small extra task – Change the font of the text into "Arial", and increase the size to 16 points. **Tip:** Copy the "arial.ttf" –file from Moodle, and place it in your PyCharm project. Then load the font in your Pillow-code, and use it!

Extra task: Let the user decide the usable colors in the application. Tip: Ask the user for a number where different numbers represent different colors, for example, 1 = blue, 2 = red etc.

Instructions on how to install Pillow-module in PyCharm:

While installing the Pillow-module, don't install the older PIL-module by accident! (the new module doesn't always work with the current Python – version)

The easier way to install Pillow in PyCharm:



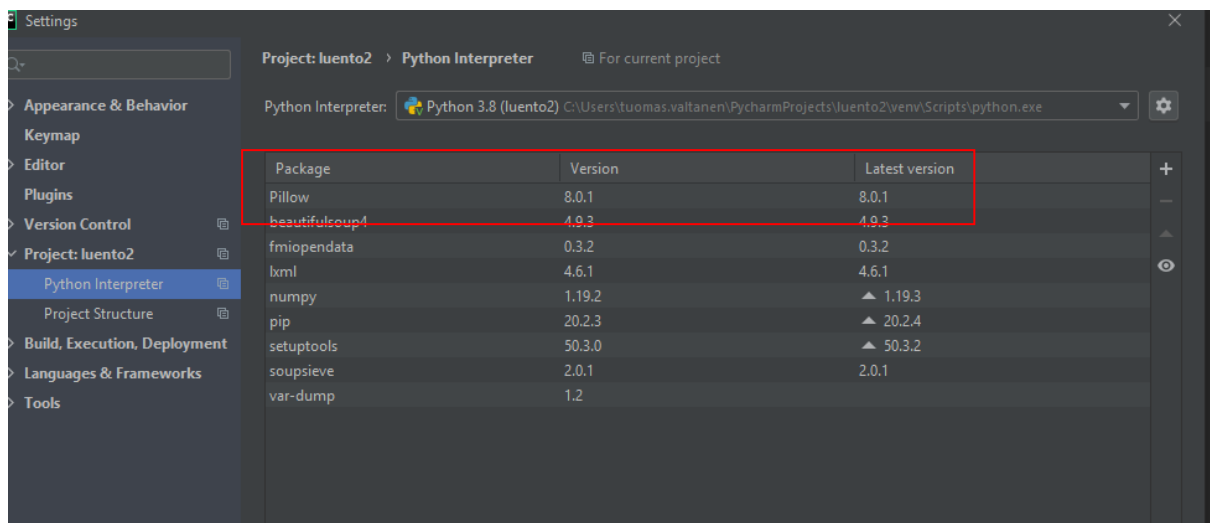
Top menu in PyCharm:

File -> Settings -> Project: [project_name] -> Project Interpreter

Click the "+" –sign in the top right corner, which opens the "Available packages" –window.

Find "Pillow" in the list, and install it. **If you installed the older PIL-module, you can also delete it from here!**

After successfully installing the correct Pillow-version, you should see it in the Project interpreter –window like this:



After this you can use the Pillow code examples from Moodle, and they should work. Here's another example as well:

<https://code-maven.com/create-images-with-python-pil-pillow>

Advanced tasks!

Note: These exercises are completely optional, and are not necessarily required for a good grade. Pick those that interest you! Some of these exercises might get quite challenging, and they will be worth of even more extra points for grading!

4. Create an application that searches for prime numbers between the range of 2 and 100. Print each number, but use green background color for prime numbers, and red background color for other numbers.

Note: a prime number is an integer that can only be divided by itself and 1. For example, 7 is a prime number, but 9 is not (because 9 is also divisible by 3).

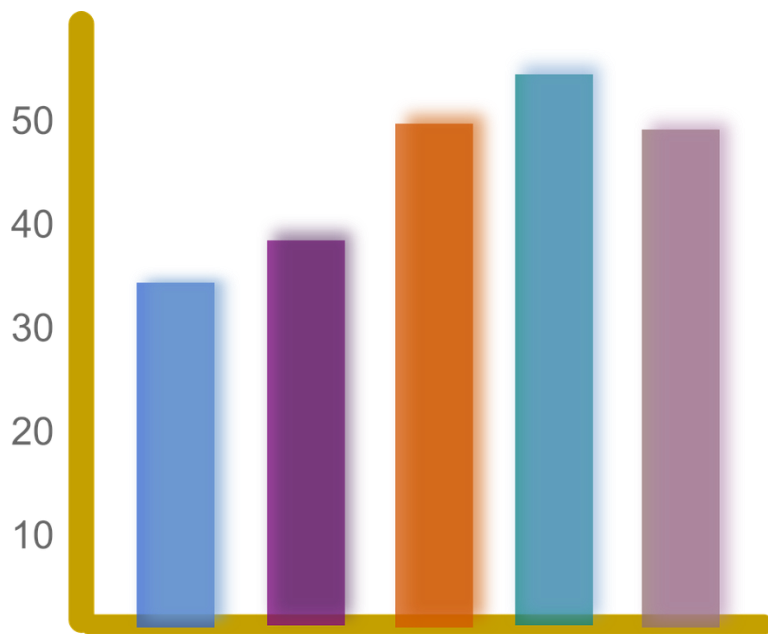


Filename of the exercise = *exercise8_4.py*

Typical code amount : **7-15 lines** (empty lines/comments not included)

5. Ask the user for a list of numbers, and create a bar chart based on them. You can use any charting module in Python you wish. Common ones are ***seaborn***, ***plotly*** and ***matplotlib***.

Tip: seaborn and matplotlib are very common tools in the field of data analytics and machine learning 😊



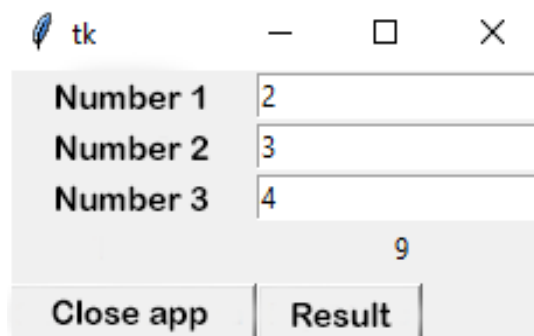
Return this exercise into the "Extra exercises return box" in Moodle!

6. Create an application that contains a graphical user interface shown in the example below. The user should be able to input different numbers in the textboxes, and if the user clicks the button, the application should sum the textbox numbers together. Show the result in the user interface as well. You can either use the default user interface module ***tkinter***, or any other common GUI-module in Python!

A good way to start is to Google around a little bit, for example:

python 3 tkinter simple calculator example

The internet is full of examples for this exercise, some of them are more suitable than others in this case 😊



Return this exercise into the "Extra exercises return box" in Moodle!

7. Try out one of the image recognition modules in Python, and test its capabilities with a number photos. The technology is called "image object recognition", if you need to google more.

You can Google, for example: *python 3 image recognition module*



Note: We're going to do image recognition in the course "Machine Learning in the Cloud" 😊

Return this exercise into the "Extra exercises return box" in Moodle!

Extra exercise X:

Is there a Python module you would like to try out? Talk with your instructor about grading these exercises. 😊