

Hello, in this video you will see how to create and test your Python code in an online environment called replit. Many of you already used replit last year, when you learnt C.

Please, notice that the use of replit is not compulsory. It makes interaction easier in online learning environments and I will use it to build the examples shown in the videos, but if you want to use any other Python developing environment, please feel free to do so. In that case, please make sure you are using Python 3.7.

Also, please notice that support on installing and using programming environments will be not provided in this module. As a Year 2 student, we expect you to be able to solve such technical problems on your own.

[SLIDE 2] For those of you who might not know replit, this is an online development environment that provides the text editor (to write your programmes) and the interpreter all in one place.

Notice that I have said “interpreter”, not compiler.

[SLIDE 3] That’s an important difference between C and Python. C is a compiled language. Instead, Python is an interpreted language.

[SLIDE 4] Both, the compiler in C and the interpreter in Python, translate the instructions written in the programming language to machine code.

[SLIDE 5] But while C translates the whole programme into machine code before executing it, Python translates an instruction and then executes it immediately; then the next instruction is translated and, if no error is found in that instruction, it is executed and so on. That makes interpreted languages slower than compiled languages, but also easier to debug.

When you use a compiler, all the errors in your programme are reported at once. Instead, when you use an interpreter, only the first error encountered will be reported. After you fix that, the next error will be shown and so on.

We will shortly see how this difference is manifested when we execute our Python programmes.

[SLIDE 6] OK, if you don’t have an account in replit yet and want to work in that environment, please, pause the video and create your account. When you are ready, please come back.

[SLIDE 7] This is the screen I see when I enter to my Replit account. If this is your first time in Replit, probably you will only have the option to create a new replit. Please, select the Python language.

[SLIDE 8] You should now see this screen

[SLIDE 9] On the left part, you can see the programming language you have selected and a brief description of it.

[SLIDE 10] On the right part, you can write the name for your project. For example, Programme1. Of course, you can use that name or any other name you want.

[SLIDE 11] After that, please click on Create replit.

[SLIDE 12] You should now see in the screen a working space like this.

[SLIDE 13] On the top you will see the URL of your project. If, for example, you would like me to have a look at your project, you can simply copy this URL and send it to me. I can then paste the URL in my browser and see your project.

[SLIDE 14] A bit below the URL, you will find your replit space. If you click here, you will be taken to a page listing all your replits (sometimes, you need to scroll down to see them).

[SLIDE 15] Next to the name of your replit space, you will see the name of the project you are working on.

[SLIDE 16] Now, let's move to the main areas. On the left you will see the list of files associated to your project. For now, we will only use one file, called main.py.

[SLIDE 17] The extension .py refers to files containing code written in Python

[SLIDE 18] The name "main" is the by-default name for our Python files in repl.it. For now, please, do not change it as I want you to focus on how to programme in Python rather than other aspects related to the management of files.

[SLIDE 19] The central part is the text editor, where you write your Python code. If you have several files, you will see several tabs here. But since we only have one file, the file main.py, you will see only one tab with the name of the file whose content is shown in the text editor below.

[SLIDE 20] Finally, the right part has two tabs. The Console tab is the tab shown by default and this is where the result of executing your Python code is shown.

Now, here is where the fact that Python is an interpreted language makes a difference with respect to C. You can write one or more Python instructions here, in the console space, and see the results of the execution of those instructions immediately.

[SLIDE 21] For example, if you write `print("Hello")`, you will see the word "Hello" displayed on the console space.

[SLIDE 22] If you want to do something a bit longer, like creating two variables and printing the value of one of those on screen, you can also do it.

[SLIDE 23] You can even write flow control instructions, like the if-else shown here.

[SLIDE 24] Notice that these 3 dots signal that the interpreter is waiting for indentation. So, make sure you press the tab key where indentation is needed. Otherwise, you will get a syntax error.

[SLIDE 25] Now, usually, we will not write instructions in the console space, but in the text editor space. In that way, you can save your work and retrieve it later.

[SLIDE 26] Here, you can see the same piece of code written in the text editor area.

To execute it, you simply

[SLIDE 27] press on the on the green button with the word Run on it.

[SLIDE 28] If your code does not have any errors, you will get your results printed on screen, like I am showing you here.

[SLIDE 29] With this, you are ready to start writing your own Python programmes as you learn more about the programming language in the next videos. Enjoy!