Hi! In this video you will learn about a very useful feature built-in in Pyhton to avoid your code from crashing due to different types of unexpected events. This is known as exception handling.

[SLIDE 2] An exception is an error that happens during the execution of the code.

[SLIDE 3] For example, when the user enters abnormal input,

[SLIDE 4] when the programme attempts to write or read an inexistent file or

[SLIDE 5] when the programme divides by zero.

[SLIDE 6] Exception handling is the process that allows you to consider such errors and deal with them to avoid the programme crashing.

[SLIDE 7] Let's have a look at one example where the programme will crash. In this programme, it is expected that the user enters a number. If everything goes ok, the user enters a number which is stored in variable x. Then, the number is increased in 6 units and printed on screen. But, what happens if the user enters a piece of text instead of a number?

[SLIDE 8] Well, in that case, an exception is raised and the programme crashes. Here you can see the error message produced in repl.it when the user enters the word hello into this programme.

[SLIDE 9] These lines tell you what lines of code caused the error

[SLIDE 10] In the last line, you can identify the type of exception that has been raised.

[SLIDE 11]There are many types of built-in exceptions in Python. Here, I am showing you 3 examples. The NameError exception is raised if you want to use a variable that has not been created. The ValueError exception, like the one in this example, happens when a variable has the wrong value and the ZeroDivisionError, as its name implies, is raised when there is an attempt to divide by zero. You can read about all types of built-in exceptions in the link at the bottom of this slide.

[SLIDE 12] Finally, details about the error are given immediately after the type of exception.

If we want to avoid this abnormal termination of our programme, we use the try statement.

[SLIDE 13] Here, you can see the previous code with two additional instructions: try and except.

[SLIDE 14] Inside the try instruction we have written our previous code. In this way, the system is going to execute these instructions under try mode. If everything goes fine, that is, if there is no error when executing the instructions inside the try statement, the statement except is ignored.

[SLIDE 15] Instead, if the try block rises an error, the instructions in the except block will be executed.

[SLIDE 16] Here, you can see how the programme stops from crashing when the user enters a wrong input. Instead of crashing, a message indicating the problem is printed on screen and the programme ends normally.

[SLIDE 17] Now that you know the basics of handling errors, please use them as much as possible to prevent your programmes from crashing due to execution errors.

[SLIDE 18] Now, there are cases where you want your programme to raise an exception specific to the problem you are solving. For example, you want the user to enter a number in a given range. If the user enters a number out of range, you would like to raise an exception.

[SLIDE 19] In that case, we use the reserved work raise, followed by the type of error that has occurred. Let's see an example.

[SLIDE 20] This code is the same code we just visited that has been extended to ask for the user to enter a number in a range. The new lines of code have been highlighted in red. So, the main differences with respect to the previous code are:

[SLIDE 21] We instruct the user to enter a number in a range: between 1 and 10.

[SLIDE 22] We check whether the user has complied with entering a number between 1 and 10. If not, we raise an exception of type IndexError. You can choose any built-in type here, but it is good to select one close to the error occurring. Since this is an out of range error, and there is no error in that category, I have chosen this that refers to the case when an array index is out of range.

[SLIDE 23] Finally, I have now two excepts, one for each type of error. In that case, a different message will be displayed on the screen depending on what error has occurred.

[SLIDE 24] OK, that's all for now. If you want to read more about exceptions, you can visit this webpage from the official Python website.

In the next, and last video for this week, we will review 4 extra aspects you will need to know for the first assessment. In that way, if you want to start working on that early, you will have all the information you need on time.