3.5. Runtime Errors

The second type of error is a runtime error, so called because the error does not appear until you run the program. These errors are also called **exceptions** because they usually indicate that something exceptional (and bad) has happened.

Runtime errors are rare in the simple programs you will see in the first few chapters, so it might be a while before you encounter one.

Check your understanding

debug-51: Which of the following is a run-time error?

- (•) A. Attempting to divide by 0.
- ( ) B. Forgetting a colon at the end of a statement where one is required.
- ( ) C. Forgetting to divide by 100 when printing a percentage amount.

Check me Compare me

✔ Python cannot reliably tell if you are trying to divide by 0 until it is executing your program (e.g., you might be asking the user for a value and then dividing by that value—you cannot know what value the user will enter before you run the program).

Activity: 1 -- Multiple Choice (question4_5_1)

debug-52: Who or what typically finds runtime errors?

- ( ) A. The programmer.
- (•) B. The interpreter.
- ( ) C. The computer.
- ( ) D. The teacher / instructor.

Check me Compare me

✔ If an instruction is illegal to perform at that point in the execution, the interpreter will stop with a message describing the exception.

Activity: 2 -- Multiple Choice (question4_5_2)

You have attempted 3 of 2 activities on this page

✔ Completed. Well Done!

3.4. Syntax errors">
3.6. Semantic Errors">

| Back to top