New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Exceptions

Special Note: sometimes a **SyntaxError** is not found in the line mentioned by Python! This will occur when the line *above* the line mentioned has an open parentheses or open bracket that is not closed. If you get a SyntaxError exception and the line in question looks OK, look to the line above to see if there is an open parentheses or bracket that is not closed.

AttributeError: when you try to use a method or other attribute that doesn't exist for that object.

ImportError: when you try to import a module that doesn't exist or that Python can't find.

IndentationError: when you start an indent where it doesn't belong (or even start a line with one additional space) or when you don't have an indent where one is expected.

IndexError: when you try to access a list or tuple index element that doesn't exist.

IOError: when you try to open a file that doesn't exist in the specified location or don't have permission to open.

KeyError: when you try to access a dictionary key that doesn't exist.

NameError: when you try to refer to a variable that doesn't exist.

OSError: when you try to access a directory or learn about a file that doesn't exist in the specified location, or don't have permission to access.

SyntaxError: when the syntax of your code is incorrect (missing or misplaced commas, parentheses, brackets, etc.

> Special Note: sometimes the error is not in the line mentioned by Python! This will occur when the line *above* the line mentioned has an open parentheses or open bracket that is not closed. If you get a SyntaxError exception and the line in question looks OK, look to the line above to see if there is an open parentheses or bracket that is not closed.
>
> **invalid syntax**: this is Python's go-to when it can't make sense of the syntax in the line. Check the line's use of quotes, commas, etc carefully against tutorial examples.
>
> **EOL while scanning string literal**: Python has reached the End Of Line of a string without seeing an end quote. This indicates that the end quote is missing. It may also indicate that the start quote is different than the end quote (i.e., starts with a double quote but ends with a single quote).
>
> **Non-ASCII character**: this error usually refers to special characters that are pasted into your IDE or text editor window from another program. These are often “smart quotes” or other “styled” text formats like Word or PDF. One way to handle this is to paste the text into a plaintext editor like Notepad or TextWrangler, and then cut and paste out of that program. Or you can retype the text in your IDE / the text editor you use to write your Python programs.

TypeError: when you try to use an incorrect object type in an expression or function call.

UnboundLocalError: when you try to assign to a global variable from inside a function.

ValueError: when you try to use an incorrect value in an expression or function call.

> **invalid literal**: this refers to the value you attempted to use as inappropriate for the expression
> **invalid literal for int() with base 10**: the literal (usually string) value you used is not a valid integer ("base 10" just means a normal decimal integer – **int()** can work in other bases)

ZeroDivisionError: when you try to divide by zero! This is illegal is most languages.

> **integer division or modulo by zero**: "modulo" refers to the modulus operator (%), which divides an integer by a number, then returns the remainder. Modulo by zero is also illegal in most languages.

-