Python Errors Explained

This is an attempt to translate Python errors into English that was compiled by a long-time community TA, Emily. It is by no means complete or comprehensive, but we've added it here as a resource.

SyntaxError: "I don't recognize what you just wrote as Python code."

'Syntax' refers to the rules that dictate how you're allowed to write down a coding language. For example, the rule that states that you may not begin a variable name with a digit is part of the syntax of Python. A SyntaxError will pop up if you violate any of Python's syntax rules. This could be forgetting to close a parenthesis or quote, mistakes in indentation, forgetting colons after function headers, etc. Note that syntax errors can be tricky - sometimes the real error is actually before or after the line that the error is reported on. When you go hunting for syntax errors, try looking at the entire block of code that surrounds the line where the error appears.

Example 1: Syntax Error: bad input (' ') - http://www.codeskulptor.org/#user29_RBAM7aLLFF_10.py

Example 2: Syntax Error: bad token("") - http://www.codeskulptor.org/#user29_9vki2pLzHk_1.py

Example 3a: Syntax Error: bad input('print') - http://www.codeskulptor.org/#user29_p7rilCJqyf_1.py

Example 3b: Syntax Error: 'return' outside function - http://www.codeskulptor.org/#user29_UWijfk8YnI_0.py

Example 4: Syntax Error: can't assign to literal - http://www.codeskulptor.org/#user29_GAQb6gQq0Y_2.py

Example 5: Indentation Error: unindent does not match any outer indentation - http://www.codeskulptor.org/#user29_DM85TuO1tN5P1fJ.py

NameError: "What in the world is X?"

A NameError is triggered when Codeskulptor doesn't recognize the variable, function name, etc. that you're referring to. A NameError is Codeskulptor's way of saying "I've never heard of that before!" Common causes of this are misspellings or forgetting to declare variables entirely.

Example 1: NameError: name 'some_variable_name' is not defined - http://www.codeskulptor.org/#user29_rvErPx9lFA_7.py

Example 2: NameError: name 'some_function_name' is not defined - http://www.codeskulptor.org/#user29_DSrwc3QXrb_1.py

TypeError: "You're trying to use some of those things in a way that wasn't intended."

This error can come up when you're trying to do perform an operation on something, but the operation and the thing just don't go together. One example is trying to multiply a dictionary and an integer - it just won't work. A helpful tool in debugging type errors is the Python type() function. Adding print statements like "print type(my_variable)" will help you figure out what's going wrong.

Example 1: TypeError: cannot concatenate 'str' and 'int' objects (or any other data type): http://www.codeskulptor.org/#user29_O8c065gLI1_11.py

Example 2: TypeError: function_name() takes exactly x arguments (y given) - http://www.codeskulptor.org/#user29_Xd5fyxBt6K_0.py

Example 3: TypeError: 'int' object is not iterable (could be a different data type) - http://www.codeskulptor.org/#user29_Vd3ft6yTVH_0.py

Example 4: TypeError: 'int' object is not callable (could be a different data type) - http://www.codeskulptor.org/#user29_BRRaiVlwpD_0.py

Example 5: TypeError: handler must be a function - http://www.codeskulptor.org/#user29_5JQUSZ4lDP_1.py

AttributeError: "That object doesn't know how do to what you asked it to do"

In OOP, objects have "attributes" - things that they're aware of, and/or know how to do. In terms of Python, attributes are an objects properties and the methods defined by its class. An AttributeError will be thrown when you ask an object to do something or access something that isn't in its class definition.

Example 1: Attribute Error: '[Object X] has no attribute [attribute Y] - http://www.codeskulptor.org/#user29_JWjKm5ZQV4_6.py

Example 2: Same as Example 1, but with an explicit class definition - http://www.codeskulptor.org/#user29_Qvjnv5BNTg_0.py

IndexError: "That list/dictionary/tuple (etc.) doesn't have that many items in it."

An IndexError happens when you try to access an index that doesn't actually exist. It's like telling someone to take 13 eggs out of a 12-egg carton - it won't work, because there are only 12 spaces. Printing out your index values, along with len(the_list_in_question) will help you in debugging these errors. Important to note: negative indices *are* possible in Python! See the video lecture on lists for more information.

Example 1: IndexError: list index out of range - http://www.codeskulptor.org/#user29_eget014sAQ_9.py

TokenError: "You probably forgot to close a bracket." *

Very simply, tokens are things that stand for other things - sort of like variables, except they're used at a more structural level. Some examples of tokens are EOF (End Of File) and EOL (End Of Line). These are the two most common ones you will come across in Python, but they are used everywhere in programming. The TokenErrors that you will see in this course are usually solved by remembering to close your brackets. See the example for a more in-depth explanation of why this is.

Example 1: TokenError: EOF in multi-line statement - http://www.codeskulptor.org/#user29_TwsBlWnsiG_3.py

*Not actually what it means - but a more detailed explanation is more appropriate for other courses, and this is the most common cause of TokenErrors in this class.

ValueError: "There's something wrong with the value of one of those arguments (but the type is ok)."

A ValueError is raised when a function receives an argument that looks ok on the surface (e.g., it receives a character, and it was looking for a character), but the value of that argument is unexpected (e.g., the function was only built to handle digits, and it received a letter 'a'). This type of error can be solved by checking the documentation for whatever function you're trying to use,

and making sure that whatever you put inside the parentheses, the function was built to handle it.

Example 1: ValueError: invalid literal for int() with base 10: ' ' - http://www.codeskulptor.org/#user29_v7yqnKyAPa_2.py

IndentationError: "Your code blocks aren't all indented to the proper levels."

This is a type of Syntax Error. See Syntax Errors, example # 5.

Miscellaneous section - these errors are either self-explanatory or not common in the level of programming done in IIPP

OverflowError - caused by trying to store, for example, a long inside an int. A long has too many bytes to fit inside the int data type

ZeroDivisionError - you guessed it - you're trying to divide by zero somewhere

ImportError - caused by trying to import a module that doesn't exist. Check your spelling.

JavaScript section - these are not Python errors. They are related to your browser, not your code.

HierarchyError - Internet Explorer is not supported by CodeSkulptor. Use Chrome or Firefox.