**Common errors in Python** – English

1. SyntaxError

If you write the syntax of your code incorrectly, you have written it wrong. In our case, it is the Python syntax. Python has a syntax and you should write it exactly as it is.

Example: "pint" instead of "print", "swatch" instead of "switch", "wile" instead of "while".

1. IndentationError

Python has a code standard; you need to write your Python code using the correct spaces and gaps. If you put gaps and spaces in the wrong place, basically misplacing the code, an IndentationError will occur.

1. NameError

NameError occurs when you use a variable that has not been initialized. Example: C = C + 1. You need to declare your variable first and initialize it with C = 0.

1. ValueError

This happens when you pass the correct type of argument to a function, but use the wrong value.

Example: Python has a function that returns the square root of a number you give it.

So if you give the function -10, you are giving it the correct type of argument, which is a number, but you are giving it the wrong value, because a negative number does not have a square root.

1. UnboundLocalError

UnboundLocalError occurs when you try to use or print a variable that is not initialized, i.e., is not defined. This error is almost the same as NameError, but UnboundLocasError occurs more often when you use try and exception.

def example\_function():

try:

result = some\_operation()

except Exception as e:

print(&quot;An error occurred:&quot;, e)

print(result)

# Calling the function

example\_function()

1. TypeError

TypeError occurs when you use a variable incorrectly, you try to do things with it that its type does not allow, like printing a sentence with a number, when you have to print a string.

1. UnicodeError

The UnicodeError occurs when Python's Unicode decoder encounters an invalid Unicode: such as

"C:\Users\Clau\English Classes\Class 03.06"

"C:\Users\Eric\Desktop\beeline.txt"

Basically, Python has a structure that you need to follow.

1. ZeroDivisionError

A ZeroDivisionError in Python occurs when you attempt to divide a number by zero. This operation is mathematically undefined, and Python raises this exception to signal the error.

1. FileNotFoundError

FileNotFoundError in Python indicates that the program cannot locate the file it is trying to access.

The specified path to the file is wrong, either due to typos or an incorrect directory structure.

1. ModuleNotFoundError

This error occurs when you try to import a file or library into your code and Python cannot find that library or file, or cannot open the file.

1. MemoryError

A MemoryError in Python occurs when the interpreter runs out of memory to allocate to your program. This can happen due to several reasons. Basically, you dont have enouth RAM memory to run a program

1. PermissionError

Is when python try to read or write a file but it lacks the necessary permissions to access a file or a directory

1. IndexError

When you attempt to access an index in a sequence (like a list, tuple, or string) that is outside the valid range of indices for that sequence.

1. KeyError

Specifically, a KeyError exception is raised when a programmer tries to access a key that does not exist in a dictionary. A dictionary, for reference, is a data structure that stores data in key-value pairs, and the value in a dictionary is accessed through its key - hence the name, KeyError.

1. AttributeError

An AttributeError in Python happens when you try to access or use an attribute that an object or class does not possess. This error typically occurs when you attempt to reference an attribute using the dot notation (e.g., object.attribute), and Python cannot find a matching attribute