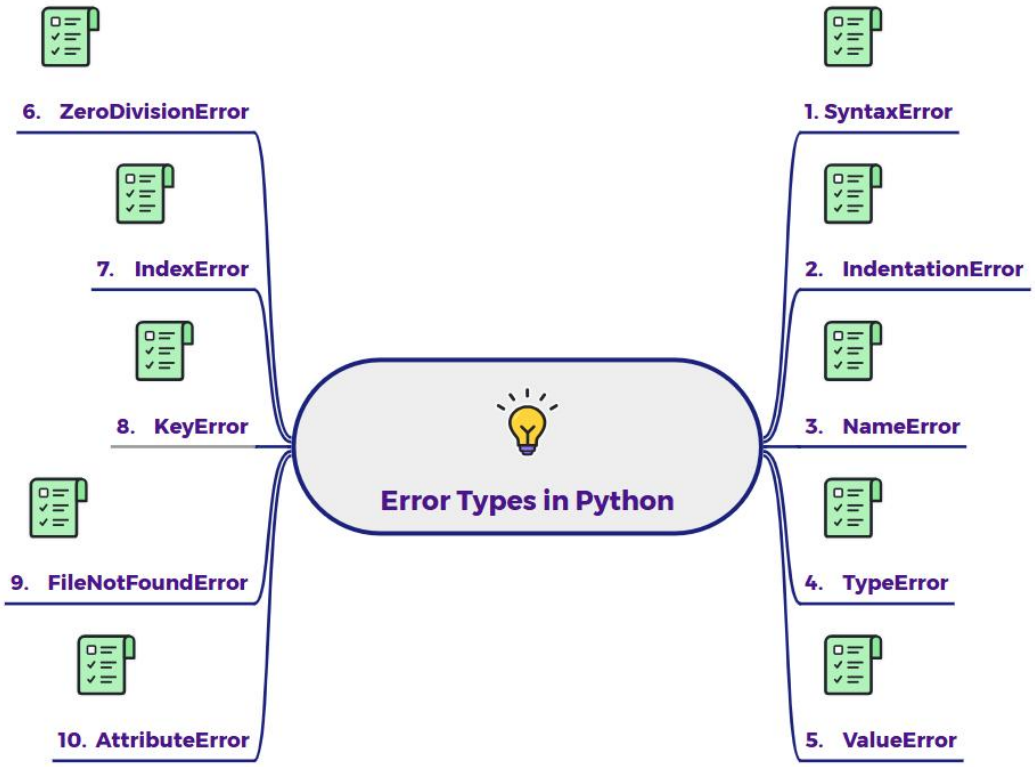
Different types of exceptions in python?



Python has a rich hierarchy of built-in exceptions that represent different types of errors that can occur during program execution. All built-in exceptions are derived from the Exception class.

Here are some of the most common and important types of exceptions in Python:

1. **SyntaxError**:
    - **What it means:** Occurs when the interpreter finds a grammatical mistake in your code, preventing it from understanding or parsing the code. This error usually happens *before* the program even starts running.
    - **Example:** Forgetting a colon at the end of an if statement or having unmatched parentheses.

2. **IndentationError**:
    - **What it means:** A type of SyntaxError that is specific to Python's strict use of indentation to define code blocks. It means your code's spacing is inconsistent or incorrect.
    - **Example:** Mixing spaces and tabs, or incorrectly indenting a line within an if or for block.
3. **NameError**:
    - **What it means:** Raised when you try to use a variable, function, or module name that hasn't been defined or properly imported in the current scope.
    - **Example:** print(my_undefined_variable)
4. **TypeError**:
    - **What it means:** Occurs when an operation or function is applied to an object of an inappropriate type.
    - **Example:** Trying to add a number to a string (5 + "hello") or calling a non-callable object.
5. **ValueError**:
    - **What it means:** Raised when an operation receives an argument of the correct type, but an inappropriate value.
    - **Example:** Trying to convert a non-numeric string to an integer (int("hello")) or using a value outside a valid range for a function.
6. **ZeroDivisionError**:
    - **What it means:** Occurs when you try to divide a number by zero.
    - **Example:** 10 / 0
7. **IndexError**:
    - **What it means:** Raised when you try to access an index that is outside the valid range of a sequence (like a list or tuple).
    - **Example:** my_list = [1, 2, 3]; print(my_list[5])

8. **KeyError**:
    - **What it means:** Raised when you try to access a dictionary key that doesn't exist.
    - **Example:** my_dict = {'a': 1}; print(my_dict['b'])
9. **FileNotFoundError**:
    - **What it means:** Raised when you try to open or access a file that does not exist at the specified path.
    - **Example:** open("non_existent_file.txt", "r")
10. **AttributeError**:
    - **What it means:** Occurs when you try to access an attribute (like a variable or method) that an object or class does not possess.
    - **Example:** my_string = "hello"; my_string.append('x') (strings don't have an append method).