Common Exceptions

We'll look at some common exceptions in Python

You have seen a few exceptions already. Here is a list of the most common built-in exceptions (definitions from the Python documentation):

> - **Exception** (this is what almost all the others are built off of)
> - **AttributeError** - Raised when an attribute reference or assignment fails.
> - **IOError** - Raised when an I/O operation (such as a print statement, the built-in open() function or a method of a file object) fails for an I/O-related reason, e.g., “file not found” or “disk full”.
> - **ImportError** - Raised when an import statement fails to find the module definition or when a **from ... import** fails to find a name that is to be imported.
> - **IndexError** - Raised when a sequence subscript is out of range.
> - **KeyError** - Raised when a mapping (dictionary) key is not found in the set of existing keys.
> - **KeyboardInterrupt** - Raised when the user hits the interrupt key (normally Control-C or Delete).
> - **NameError** - Raised when a local or global name is not found.
> - **OSError** - Raised when a function returns a system-related error.
> - **SyntaxError** - Raised when the parser encounters a syntax error.
> - **TypeError** - Raised when an operation or function is applied to an object of inappropriate type. The associated value is a string giving details about the type mismatch.
> - **ValueError** - Raised when a built-in operation or function receives an argument that has the right type but an inappropriate value, and the situation is not described by a more precise exception such as IndexError.
> - **ZeroDivisionError** - Raised when the second argument of a

> division or modulo operation is zero.

There are a lot of other exceptions as well, but you probably won't see them all that often. However, if you are interested, you can go and read all about them in the Python documentation.