Python 101

Exception Handling and FIle Handling Basics

Contents

Errors & Exceptions

In Python Errors is some undesired condition.

Exceptions are a means of breaking out of the normal flow of control of a code block in order to handle errors or other exceptional conditions.

Mind that Exceptions are not always caused by some undesired condition they can also be due to exceptional conditions.(eg. StopIteration)

Some errors are SyntaxErrors, NameError, TypeError, ValueError etc.

The try-except...n-else-finally

Python's exception handling sequence

```
try:
    STATEMENTS_TO_TRY_TO_EXECUTE_WHICH _MAY_PRODUCE_ERRORS
except EXCEPTION as VARIABLE_NAME_TO_HOLD_EXCEPTION:
    DO_SOMETHING_WITH_EXCEPTION
else:
    STATEMENTS_TO_EXECUTE_WHEN_EVERYTHING _WENT_FINE
finally:
    STATEMENTS_TO_ALWAYS_EXECUTE
```

Errors or Exception ?

In Python most of the Exceptions end with the suffix **Error** but the reason behind it is PEP8 section Exception Names.

Its also because of PEP20 as Error suffix clarifies better than the Exception suffix that the Exception is caused due to some mal-written code.

Try out the following code on the interpreter

```
>>> try:
. . . .     1/0
. . . .except: ← Catch all exceptions
. . . .     print('Exception occured')
. . . .else:
. . . .     print('Everything went fine')
. . . .finally:
. . . .     print('I will get executed no matter what')
. . . .
Exception occured

>>> try:
. . . .     1/0
. . . . except ZeroDivisionError as e: ← Catch only ZeroDivisionError
. . . .     print(e)
. . . .
division by zero
```

Some Types of Exceptions

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.

NameError- Raised when a local or global name is not found.

See the link below for all exceptions and exception hierarchy-

https://docs.python.org/3/library/exceptions.html#exception-hierarchy

File Handling

Stored Files are nothing but a bunch of bytes packed together at a certain location in secondary memory.

When we open these files we get a data-stream from them. This data stream can have 3 modes read, write or append.

- Write discards all of the existing data and writes the new data.
- Append adds keeps the existing data and writes the new data after it.
- Read simply reads the data from the data-stream.

Handling these file-data-streams or file-streams is called File Handling.

Opening a file stream

```
open(RELATIVE_PATH_TO_THE_FILE, mode, buffering=num)
```

There can be following values of mode-

- r - read, w-write, a - append as character.
- rb, wb, ab means read write append as binary.
- r+, w+, a+ open for read and write.

REMEMBER THAT w+ and w and wb and wb+ will delete all file content.

A nonzero buffering allows buffering of content on stream. -ve means use system default and +ve(!=1) means use this many bytes to buffer.

Methods on a stream

.read()- To read content from file

.write(CONTENT_TO_WROITE)- To write content to file

.readline() - To read content line by line from file

.writelines() - To write content to file from an iterable.

.tell() - Tell current cursor position.

.seek(NEW_POSITION) - Move cursor position to NEW_POSITION

Methods on a stream

.flush - flush the residual bytes from a file stream. This is automatically invoked by python on file stream close.

.writable - <Boolean**>** if file is writeable.

.readable - <Boolean**>** if file is readable.

.seekable - <Boolean**>** if file is seekable.

Attributes of a stream

.name - filename

.mode - file-stream mode

.encoding - file-stream encoding (eg. UTF-8)

Try out the following code on the interpreter

```
>>> stream = open(‘new_file’, ‘w’)
>>> stream.write(‘write this content to file’)
>>> stream.close()
>>> stream = open(‘new_file’, ‘r’)
>>> stream.read()
write this content to file
>>> stream.close() ← remember to always close the stream.
```

Interview Questions

Is TypeError in python an Exception or Error?

In python we MIGHT call it as an error but in typical sense it is an exception.

How does else work with try except?

If everything went ie. no exception was raised from try block then upon completing execution from try block the control goes to else. Followed by finally.

What is the purpose of finally?

Finally can be used for clean up actions. eg. closing file if a wrong character encoding exception occurred in middle of reading the file.

Whats the difference between w+ and w?

W+ means open for read and write and then your stuff.

W means open for writting and writting exclusively

What do tell and seek do?

They tell and move the cursor to position in a file.