Q-1 What is file function in python? What is keywords to create and write file?

Ans. File handling is an important activity in every web app. The types of activities that you can perform on the opened file are controlled by Access models. These describe how the file will be used after it has been opened.

These modes also specify where the file handle should be located within the file. Similar to a pointer, a file handle indicates where data should be read or put into the file.

In python , there are six methods or access modes, which are:

1. Read only (“r”) : this mode opens the text files for reading only. The start of the file is where the handle is located. It raise the I/O error if the file does not exist. This is the default mode for opening files as well.
2. Read and write (“r+”) : this method opens the file for both reading and writing . the start of the file is where the handle is located. If the file does not exist, an I/O error gets raised.
3. Write only(“w”): this mode opens the file for writing only. The data in existing files are modified and overwritten. The start of the file is where the handle is located. If the file does not already exist in the folder, a new one gets created.
4. Write and read(“w+”): this mode opens the file for both reading and writing . the text is overwritten and deleted from an existing file. The start of the file is where the handle is located.
5. Append only(“a”): this mode allows the file to be opened for writing. If the file doesn’t yet exist, a new one gets created. The handle is set at the end of the file . the newly written data will be added at the end, following the previously written data.
6. Append and read(“a+”): Using this method , you can read and write in the file. If the file doesn’t already exist, one gets created.

The newly written text will be added at the end, following the previously written data.

How to create files in python

In python , you use the open() function with one of the following options – “x” or “w” – to create a new file.

- **“X” : create:**
  This command will create a new file if and only if there is no file already in existence with that name or else it will return an error.

Example of creating a file in python using the “x” command:

f= open(“myfile.txt”,”x”)

we have now created a new empty text file. But if you retry the code above – for ex if you try create a new file with the same name as you used above you will get an error notify you that file already exists.

- **Write- “w” :**
  This command will create a new text file whether or not there is a file in the memory with the new specified name.
  It does not return an error if it finds an existing file with the same name – instead it will overwrite the existing file.

Ex :

F = open (“myfile.txt”,”w”)

Q-2 Explain Exception Handling ? what is an error in python?

Ans. When a python meets an error, it stops the execution of the program. An error in python might be either an error in the syntax of an expression or a python exception . we will see what an exception is. Also, we will see the difference between a syntax error and an exception in this tutorial. Following that we will learn about trying

and except blocks and how to raise exceptions and make assertions. After, that we will see the python exceptions list.

An exception is python is an incident that happens while executing a program that causes the regular course of the program's commands to be disrupted. When a python code comes across a condition it cannot handle , it raises an exception that describes an error is called an exception.

Errors are the problems in a program due to which the program will stop the execution . On the other hand, exceptions are raised when some internal events occur which changes the normal flow of the program.

Two types of Error occurs in python.

1. Syntax errors
2. Logical errors

Syntax errors:

When the proper syntax of the language is not followed then a syntax error is thrown

EX:

```
Amount = 10000
If (amount > 2999)
    Print("You are eligible to purchase Dsa self Paced")
```

Output:

Syntax error: invalid syntax

It returns a Syntax error message because After then if statement a colon: is missing. We can fix this by writing the correct syntax.

Logical errors

When in the runtime an error that occurs after passing the syntax is called exception or logical type. For example, when we divide any number by zero then the ZeroDivisionError exception is raised, or when we import a module that does not exist then ImportError is raised.

EX:

```
Marks = 10000

a= marks/0

print(a)
```

output:

ZeroDivisionError : division by zero

In the above example the ZeroDivisionError as we are trying to divide a number by 0.

Q-3 How many except statements can a try- except block have ? Name some built-in exception classes:

Ans . Exception are an error scenarios that alter the normal execution flow of the program. The process of taking care of the possible exceptions is called exception handling. If exceptions are not handled properly, the program may terminate prematurely. It can cause data corruption or unwanted results. Python exception handling is achieved by three keyword blocks try ,except and finally. The Try block contains the code that may raise exceptions or errors. The except block is used to catch the exceptions and handle them. The catch block code is executed only when the corresponding exception is raised. There can be multiple catch blocks. We can also catch multiple exceptions in a single catch block. The finally block code is always executed whether the program executed properly or it raised an exception .

We can also create an “else” block with try-except block. The code inside the else clock is executed if there are no exceptions raised.

Some Buit- in – Exception classes:

Some of the built- in- exception classes are in python.

ArithmeticError : this is the base class for arithmetic errors.

AssertionError : raised when an assertion fails.

AttributeError : when the attribute is not found.

BufferError:

EOFError : reading after end of file.

ImportError : when the imported module is not found.

LookupError : base exception for lookup errors.

MemoryError : when out of memory occurs

NameError: when a name is not found globally.

OSError : base class for I/O errors

RefernceError

RuntimeError:

StopIteration, StopAsyncIteration

SyntaxError : Invalid syntax

TypeError : invalid argument type

SystemError: internal error in the python interpreter.

ValueError : invalid argument value.

Q-4 When will the else part of Try- except-else be excecuted?

Ans. Try : this block will test the excepted error to occur

Except : here you can handle the error

Else: If there is no exception then this block will be executed

Finally : Finally block always gets executed either exception is generated or not

Syntax:

```
Try :
        # Some code …….
Except:
        # optional block
        # Handling of exception
else:
        # execution if no execution
Finally:
        # some code ……..(always excetued)
```

Let's first understand how the try and except works-

First try clasuse is executed the code between try and except clause.

If there is no exception the only try clause will run, except clause will not get executed.

If any exception occurs , but the except clause within the code doesn't handle it, it is passed on to the outer try statements. If the exception is left unhandled , then the execution stops.

A try statement can have more than one except clause.

Else cause the code enters the else block only if the try clause does not raise an exception.

Q-5 Can one block of except statements handle multiple exception?

Ans . It is possible to define multiple exceptions with the same except clause . It means that if the python interpreter finds a matching exception , then it will execute the code written under except clause:

In general the syntax for multiple exceptions is as follows

Except(Exception1, Exception2, ……..ExceptionN) as e:

When we define except clause in this way, we expect the same code to throw different exceptions Also we want to take the action in each case.

Ex:

```
import sys

try:

d = 8

d =d+’5’

except(typeerror,syntaxerror)      as e :

print sys.exc_inf
```

Q- 6 When is the Finally block executed?

Ans In programming , there may be some situation in which the current method ends up while handling some exceptions. But the method may require some additional steps before its termination, like closing a file or a network and so on. So in order to handle these situations, Python provides a keyword finally, which is always executes after normal termination of try block terminates due to some exception.

Syntax:

Try:

    # Some code ……

Except:

    # optional block

    # Handlimg of exceptions (if required)

Finally :

    # some code …….. (always executed)

Finally block is always executed after leaving the try statement. In case if some exception was not handled by except block , it is re-raised after execution of finally block.

Finally block is used to deallocate the system resources.one can use finally just after try without using exceptblock, but no exception is handled in that case.

Q-7 What happens when ,,l”== 1 is executed?

Ans.