WEEK 5 : EXCEPTION HANDLING, I/O FILE, FILE HANDLING, STRING PROCESSING

LECTURE 1 : EXCEPTION HANDLING

1. Syntax Error
2. Run-time Errors :

Name error : name ‘x’ is not defined

Zero Division Error : Division by zero

Index Error : list assignment index out of range

1. We can have as many as except cases.

Try:

Except IndexError: #single

Except (NameError,INdexError,KeyError): #double

Except: #works for all type of errors

Else:

1. If(condition):

Else(condition):

1. Flow Of Control :

LECTURE 2 : STANDARD INPUT AND OUTPUT :

Sep = “ “, when specified it gives spaces in print() function.

LECTURE 3 : FILE HANDLING :

Open : fh = open(“filename.py”,”r”)

R = only read.

W = write (for only empty files)(if files are not empty, it overwrites).

A = append to an existing file.

Contents = fh.read() : read entire file in one string and has ‘\n’ at end of string.

Contents = fh.readlines() :reads entire file as list of strings. Each string has ‘\n’ at end.

Fh.write(s)

It returns number of characters written.Include ‘\n’ explicitly to go to new line.

Fh.writelines(l)

Must include ‘\n’ explicitly for each string.

Close = flush = disconnect =

Fh.close()

Fh.flush() : all pending writes are copied to disk, without closing the file, it manually forces to write disk.

Get rid of trailing ‘\n’ :

For line in contents:

s = line[:-1]

use rstrip to remove trailing whitespace :

for line in contents:

s = line.rstrip()

LECTURE 4: STRING FUNCTIONS :

S.RSTRIP() : trailing whitespace

s.lstrip() : leading whitespace

s.strip() : leading and trailing whitespace

Search for text in string :

s.find(pattern) : returns first position if pattern occurs, -1 if not found.

s.index(pattern) : returns index if found else

returns valueerror if not found

REPLACE :

s.replace(fromstr,tostr) : Returns copy of s with each occurrence of fromstr replaced by tostr

SPLIT : s.split()

JOIN : string = “ “.join(list)

s.capitalize() = first letter capitalize,

s.title() ,s.upper() = uppercase all

s.lower() – lowercase all

s.swapcase() – invert cases of string

RESIZING :

s.center(n) : returns string of length n with s centered,

rest blank

s.center(n,”\*”) : returns string of length n with s centered, rest \*

s = "hello"  
  
print(s.index("hello"))  
  
print(s.replace("hello","world"))  
  
print(s.center(100,"\*"))  
print(s.ljust(25,"%"))  
print(s.rjust(25,"$"))

o/p :

0

world

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*hello\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

hello%%%%%%%%%%%%%%%%%%%%

$$$$$$$$$$$$$$$$$$$$hello

OTHER FUNCTIONS :

s.isalpha() = alphabetic

s.isnumeric() = numeric

s.isaplhanum = aplhanum

* LECTURE 5 : FORMATTING PRINTED OUTPUT :

1. Format():

Print(“{}”.format(47))

Print(“{f} {s}”.format(f=9,s=10))

Print(“{0:3d}”.format(4))

D = integer

f = float

s = string

o = octal

x = hexadecimal

{0:6.4f}

6 = total space

LECTURE 6 : pass, del() and None