Python notes

34. press windows + E keys to open file manager or file explorer or windows explorer window

35.entering into our path like

D:\cmr\python\apps\console\pop\shell.cmd

36. press alt + d keys to activate cursor on address bar of file manager window

37. type cmd content in address bar as well as press enter

* Opened command prompt window with D:\cmr\python\apps\console\pop\shell.cmd

38. D:\cmr\python\apps\console\pop\shell.cmd>python -c"2+7"

39. D:\cmr\python\apps\console\pop\shell.cmd>python -c"print(2+5)"

Output:

7

40. D:\cmr\python\apps\console\pop\shell.cmd>python -c"print(type('5'))"

Output:

<class 'str'>

41. D:\cmr\python\apps\console\pop\shell.cmd>python -c"print(type(5))"

Output:

<class 'int'>

42. D:\cmr\python\apps\console\pop\shell.cmd>python -c"print(type(5.0))"

Output:

<class 'float'>

43. D:\cmr\python\apps\console\pop\shell.cmd>python.exe

Output:

Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32

43. D:\cmr\python\apps\console\pop\shell.cmd>python.exe

Output:

Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32

44. >>>import os

45. >>>os.system(‘cls’)

46. >>>os.system(‘cd’)

Output:

D:\cmr\python\apps\console\pop\shell.cmd

0

47. >>> from datetime import datetime

* ‘import’ loads python module file like datetime.py
* Before datetime.py module launch from datetime package or folder
* Package alias name is folder
* Therefore ‘from’ clause develop by directory or folder feature
* ‘import’ clause develop by file feature
* ‘now()’ function process from datetime.py module file

48. >>>datetime.now()

Output:

datetime.datetime(2025, 5, 21, 11, 59, 47, 782022)

about indentation:

* Indentation sign develop by blank space(s) from 0th column
* Normally, indentation develop by tab non printable character
* Indentation alias name is scope or block
* Indentation develop definition part of class, function , conditional statement ,loop and etc..
* Most important:- indentation before part always called as declaration part of class,function,conditional statement, loop and etc..

49. note: python allocate four blank space for tab non-printable charecters

50. note: declaration path always ends with colon (:).

51. note: declaration part after part is called as definition part

52. note: definition part always starts with blank space or tab

Exception handling feature

* Try,except,finally
* Exception handling purpose:
* Exception handling feature continuously execute a program even a run time error occurred
* Exception handling feature if missing then program will be terminated due to runtime error
* ‘try’ indentation process until error occurs
* ‘except’ indentation process when error occurs
* ‘finally(optional) indentation part erases unnecessary objects from memory even error occurred or not
* Therefore finally indentation process always exist .

53. >>>exit

Above exit command terminates python shell prompt and return back to command prompt window

54.type the following coding in any one non-ide like notepad

Filename: “eh01.py”

File path: D:\cmr\python\apps\console\pop\shell.cmd

Save as type: all files

Following program coding:

‘’'

title:

without exception handling feature

'''

print('welcome')

print(10/0)

print('thanks')

55. pres ctrl + s for save updated content of eh01.py file

56. type following command for execute python console app program via command prompt window

57. D:\cmr\python\apps\console\pop\shell.cmd>cd..

Above cd.. command exit from current folder like shell.cmd

58. D:\cmr\python\apps\console\pop>dir/ad/b

Above command list name of folders

Output:

non ide

shell.cmd

59. D:\cmr\python\apps\console\pop>cd non ide

Abvove cd <folder name> command entering into mentioned folder name

60. D:\cmr\python\apps\console\pop\non ide>dir/ad/b

Above command list name of folders

Output:

eho1.py

61. D:\cmr\python\apps\console\pop\non ide> python.exe eh01.py

Output:

welcome

Traceback (most recent call last):

File "D:\cmr\python\apps\console\pop\non ide\eh01.py", line 8, in <module>

print(10/0)

ZeroDivisionError: division by zero

62. type the following coding in any one editor like notepad

File name: eh02.py

File path: D:\cmr\python\apps\console\pop\non ide

Save as type: all files

Coding:

‘’’

Title:

With exceptional handling

‘’’

Print(‘welcome’)

try:

print(10/0)

except exception as e:

print(“error= “,e)

print(‘thanks’)

63. D:\cmr\python\apps\console\pop\non ide>dir/a-d/b

Output:

eh01.py

eh02.py

64. . D:\cmr\python\apps\console\pop\non ide>python.exe eh02.py

Output:

Welcome

error=division by zero

thanks

65. . D:\cmr\python\apps\console\pop\non ide>exit

Above exit dos command quit command prompt window

66.following steps describe how to develop python using internal ide like idle

(integrated development learning language)

67. press windows key for activate start button pull down box

68. type IDLE

69.select ( IDLE python 3.13 64 bit)

70.press run as administrator

71. 'sunday'.upper()

Output:

'SUNDAY'

72.'SUNDAY'.lower()

Output:

'sunday'

73.'sunday'.title()

Output:

'Sunday'

74.'SuNdAy'.title()

Output:

'Sunday'

75.'SUnDay MoNdAy'.title()

Output:

'Sunday Monday'