DAY 30

ERRORS AND EXCEPTION

SYNTAX:

try:

something that might cause an exception

except:

do this if there was an exception.

else:

Do this if there were no exceptions

finally:

Do this no matter what happens.

EXAMPLE:

try:  
 a\_dictionary = {"name":"pavithra"}  
 print(a\_dictionary["student"])  
except KeyError as error\_message:  
 print(f"the key {error\_message} does not exist.")

OUTPUT:

the key 'student' does not exist.

RAISE OWN ERROR:

EXAMPLE:

height = float(input("enter your height(in m):"))  
weight = int(input("enter your weight:"))  
  
if height >3:  
 raise ValueError("human height can't be over 3 meters.")  
  
bmi = weight/ height \*\* 2  
print(bmi)

OUTPUT:

enter your height(in m):6

enter your weight:89

Traceback (most recent call last):

File "C:\Users\VICTUS\PycharmProjects\day-30\pythonProject\.venv\main.py", line 12, in <module>

raise ValueError("human height can't be over 3 meters.")

ValueError: human height can't be over 3 meters.

EXAMPLE:

fruits = ["Apple", "Pear", "Orange"]

 def make\_pie(index):

try:

fruit = fruits[index]

except IndexError:

print("Fruit pie")

else:

print(fruit + " pie")

make\_pie(4)

EXAMPLE:

facebook\_posts = [

{'Likes': 21, 'Comments': 2},

{'Likes': 13, 'Comments': 2, 'Shares': 1},

{'Likes': 33, 'Comments': 8, 'Shares': 3},

{'Comments': 4, 'Shares': 2},

{'Comments': 1, 'Shares': 1},

{'Likes': 19, 'Comments': 3}

]

 def count\_likes(posts):

  total\_likes = 0

for post in posts:

try:

total\_likes = total\_likes + post['Likes']

except KeyError:

pass

return total\_likes

 count\_likes(facebook\_posts

JSON- JAVASCRIPT OBJECT NOTATION:

FORMAT:

{

“amazon”:{

“email”: “[pavi@gamil.com](mailto:pavi@gamil.com)”,

“name”:”David”

},

“twitter”:{

“email”:” something”,

“name”:”else”

}

}