Day 15

Exception handling in python

- Exception handling in python is a way to manage errors gracefully without stopping program execution. It helps prevent crashes by catching and handling runtime errors.

Syntax: try, except, else, and finally blocks to handle exceptions.

1. try: - code that may raise an exception.
   - contains the code that might raise an error.
2. except: catches and handle the error if it occurs.
3. else block: runs only if no exception occurs inside the try block.
4. finally block:
   - always executes, whether an exceptional occurs (or) not : (useful for cleanup).

Common Exceptions in Python:

1. Zero Division error : Division by zero is ~~not~~ allowed

2. Value Error : invalid data type.
   eg. entering text instead of a number.

3. Type error : Mismatch of data types in operations.

4. Index Error : Accessing an index that doesn't exist in a list.

5. key Error : Accessing a non-existent key in a dictionary.

6. File not Found Error : Trying to open a file that doesn't exist.

Raising custom exceptions:

you can create your own exceptions using raise.

```
def check_age (age):
    if age < 18:
        raise Value Error ("you must be at atleast 18 years old.")
    else:
        print ("Access granted!")
try:
    check_age (16) except
    value_error as e:
    print ("Error :", e)
```

Summaary !

- Exception Handling Prevents program crashes.
- use try, except, else and finally blocks.
- Handle specific errors like zero Division Error, ValueError, etc.
- custom Exceptions allow better error messages.