🧠 Topic 12: try/except

🔹 Definition

try/except is used to handle errors in Python without crashing the program.

🔹 Terminologies

- **try block:** Code that may raise an error.
- **except block:** Code executed if an error occurs.
- **finally block:** Optional block executed always.
- **raise:** Manually throw an exception.

🔹 Example + Output

```
try:
    x = int(input("Enter a number: "))
    print(10 / x)
except ZeroDivisionError:
    print("Cannot divide by zero!")
except ValueError:
    print("Invalid number!")
```

Output if input=0:

```
Cannot divide by zero!
```

🔹 Challenges

Solved Challenge 1: Handle division by zero

```
try:
    print(5/0)
```

```
except ZeroDivisionError:
    print("Error: Division by zero")
```

Solved Challenge 2: Convert string to int safely

```
try:
    num = int("abc")
except ValueError:
    print("Invalid integer")
```

Your 18 Challenges:

1. Handle IndexError in a list.
2. Handle KeyError in a dictionary.
3. Handle FileNotFoundError.
4. Handle TypeError.
5. Use finally block to print “Done”.
6. Raise custom exception if age < 18.
7. Handle multiple exceptions in one except.
8. Use else block in try/except.
9. Open file safely with try/except.
10. Handle AttributeError.
11. Try converting float string to int.
12. Handle ValueError in input loop.
13. Prevent program crash on zero division.
14. Handle errors in a function.
15. Use nested try/except.
16. Catch all exceptions with except Exception.
17. Try casting list to int.
18. Handle error when importing a module.