CoGrammar

Error Handling with Try-Except Blocks

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Progression Criteria

✅ Criterion 1: Initial Requirements

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✅ Criterion 2: Mid-Course Progress

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✅ Criterion 3: Post-Course Progress

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✅ Criterion 4: Employability

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

Lecture Objectives

1. Discuss the purpose of try-except blocks and how they are used to handle errors.

2. Identify common errors that occur in programmes and how try-except blocks handle them.

3. Implement try-except blocks.

CoGrammar

Recap on Error Types

Poll:

Assessment

Defensive Programming

- Programmers anticipate errors.
  - User errors
  - Environment errors
  - Logical errors
- Code is written to ensure that these errors don't crash the code base.
- Two ways - if statements and try-except blocks.

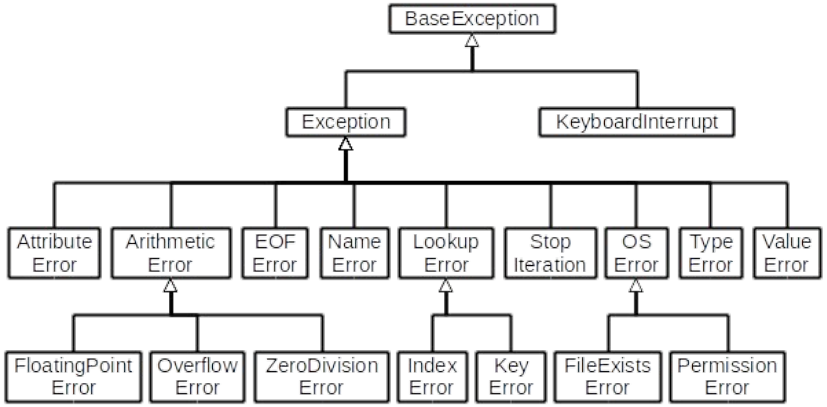
What are exceptions ?

An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the its initial instructions.

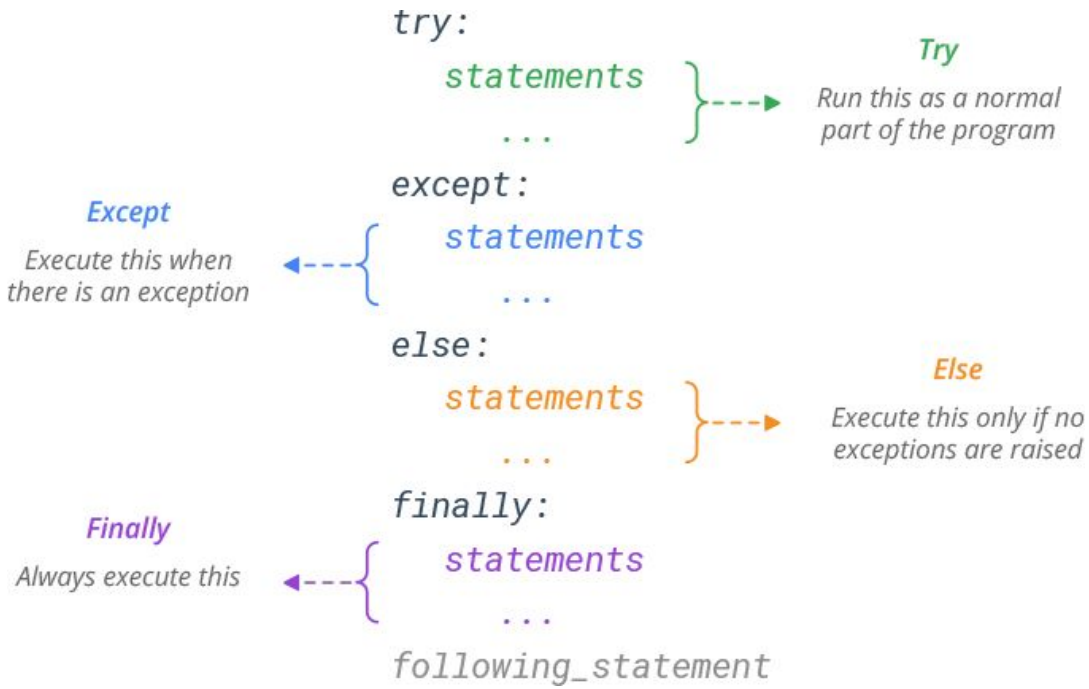
Question:

What are some scenarios where you may want to use a try-except block?

Basic types of exceptions

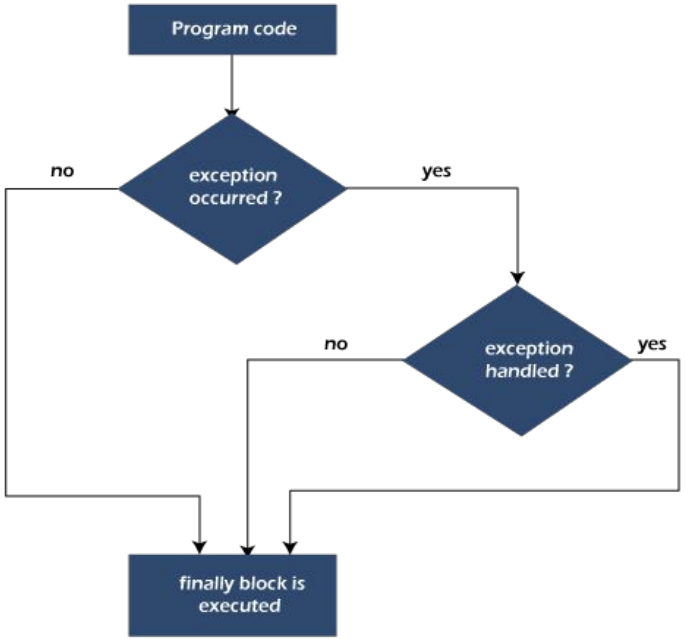


try-except block structure



https://www.learnbyexample.org/python-exceptions-try-except/

finally block



Raising Exceptions

- There will be occasions when you want your program to raise a custom exception whenever a certain condition is met.

- In Python we can do this by using the “raise” keyword and adding a custom message to the exception: In the next example we’re prompting the user to enter a value> 10. If the user enters a number that does not meet that condition, an exception is raised with a custom error message.

Raising Exceptions

```
num = int(input("Please enter a value greater than 10 : "))

if num < 10:

    raise Exception(f"Your value was less than 10. The value of num was : {num}")
```

Terminology

KEYWORD	DESCRIPTION
try	The keyword used to start a try block.
except	The keyword used to catch an exception.
else	An optional clause that is executed if no exception is raised in the try block.
finally	An optional clause that is always executed, regardless of whether an exception is raised or not.
raise	The keyword used to manually raise an exception.
as	A keyword used to assign the exception object to a variable for further analysis.

A Note on try-except

- ★ It may be tempting to wrap all code in a try-except block. However, you want to handle different errors differently.
- ★ Don't try to use try-except blocks to avoid writing code that properly validates inputs.
- ★ The correct usage for try except should only be for "exceptional" cases. Eg: The potential of Division by 0.

Challenge:

What is the term for using Try-Catch blocks in our code?

Poll:

Assessment

Wrapping Up

Exceptions

Events that occur during programme execution that can disrupt the normal flow of the program.

Try-except block

Allows you to handle errors gracefully and prevent your program from crashing).

Finally block

A part of a try-except block that is always executed regardless of whether an exception occurs or not.

CoGrammar

Questions around Exception Handling

CoGrammar

Thank you for joining