



CoGrammar

Types of Errors

**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. **Identify and categorise different types of errors in Python.**
2. **Explain the causes and effects of various error types.**
3. **Implement effective error handling techniques to manage errors in Python.**

CoGrammar

Recap on Iteration



Poll:

Assessment



We all make mistakes :)

- ★ No programmer is perfect, and we're going to make a lot of mistakes in our journey – and that is perfectly okay!
- ★ What separates the good programmers from the rest is the ability to find and debug errors that they encounter.

Syntax Errors

- ★ Some of the easiest errors to fix... usually
- ★ Mainly caused by typos in code or Python specific keywords that were misspelled or rules that were not followed.
- ★ When incorrect syntax is detected, Python will stop running and display an error message.

Syntax Errors

```
print("Hello world!")
```

```
print("Hello world!"  
      ^
```

```
SyntaxError: '(' was never closed
```

Logical Errors


$$1 + 1 = 3$$

Logical Errors


- ★ Logical errors occur when your program is running, but the output you are receiving is not what you are expecting.
- ★ The code could be typed incorrectly, or perhaps an important line has been omitted, or the instructions given to the program have been coded in the wrong order.



Question:



What kind of error occurs when you misspell a keyword, such as **for** or **if**?



Runtime Errors

```
print(100/0)
```

```
print(100/0)
```

~~~~^~

```
ZeroDivisionError: division by zero
```



## Challenge:



Correct the following syntax error:

```
prnt("Hello World")
```





Poll:

**Assessment**





# Wrapping Up

---

## Error Handling

A process of dealing with errors that may occur during the execution of a program.

## Types of Errors

Syntax, runtime and logical errors.

# CoGrammar

Questions around Error Types



# CoGrammar

**Thank you for joining**

