

Exception Handling

Guttag Chapter 9

Goals

Learn Terminology related to Exception Handling in Guttag Chapter 9

See Examples of Exceptions in Code

Practice Exception Handling

Exception Handling Terminology

Terms: Program Termination

Definition

- Fancy name for the moment a program stops running
- Could be caused by an unknown bug (CRASH)
- Could be caused by just being finished with the code

Terms: Exceptions

Definition

- Fancy name for the specific type of error that the python interpreter reports when a program terminates unexpectedly

Example

- `print(hello)`

-

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-1-1cd80308eb4c> in <cell line: 1>()  
----> 1 print(hello)  
  
NameError: name 'hello' is not defined
```

Terms: Exception Handling

Definition

- special python syntax that can prevent a crash when an exception occurs

Example

- `try:`
 - `print(hello)`
- `except NameError:`
 - `print("something above is not working correctly, but the code will keep running anyway!")`

```
something above is not working correctly, but the code will keep running anyway!
```

Terms: Try-Except Block

Definition

- special python syntax that can prevent a crash when an exception occurs

Example

- `try:`
 - `print(hello)`
- `except NameError:`
 - `print("something above is not working correctly, but the code will keep running anyway!")`

```
something above is not working correctly, but the code will keep running anyway!
```

Terms: Catching an Error

Definition

- preventing a crash

Example

- `try:`
 - `print(hello)`
- `except NameError:`
 - `print("something above is not working correctly, but the code will keep running anyway!")`

```
something above is not working correctly, but the code will keep running anyway!
```


Terms: Default Message

Definition

- catching a default error message that is provided by the python interpreter

Example

- try:
 - `print(hello)`
- except `NameError` as `msg`:
 - `print(msg)`

```
name 'hello' is not defined
```

Terms: Raising an Error

Definition

- Intentionally causing the python interpreter to throw a specific kind of error

Example

- `result = None`
- `if result is None:`
 - `raise ValueError`

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-12-7d8c89c534fb> in <cell line: 3>()  
      2  
      3 if result is None:  
----> 4     raise ValueError  
  
ValueError:
```

Terms: Assert

Definition

- Cause program termination if something is wrong
- Intentionally causing the python interpreter to throw an AssertionError

Example

- `result = None`
- `assert result is True`

```
-----  
AssertionError                                Traceback (most recent call last)  
<ipython-input-13-dd66c6c699ed> in <cell line: 2>()  
      1 result = None  
----> 2 assert result is True  
  
AssertionError:
```

See Exception Examples

Examples

- ZeroDivisionError
- TypeError
- ValueError
- IndexError
- NameError
- UnboundLocalError
- <https://docs.python.org/3/library/exceptions.html>

Practice Exception Handling

Summary

- Exceptions can sometimes be expected in code
- there are ways to keep the code running, even if there is an error

Announcement

- No class on Friday, March 19, 2025 or Monday, March 24, 2025