Exceptions in Python

CIT-129: Python 2

by Coral Sheldon-Hess

Exceptions are like a fancy `goto`.

People will tell you they're slow, but **not true in Python.**

Old languages:
Look before you Leap

Python:
Easier to Ask Forgiveness than Permission

Types of exceptions:

- try/except/else/finally
- raise
- assert
- with/as

Exception formatting:

- `raise Exception("text description of the exception")`
- `assert(`*`Boolean`*`), "text description of what may've failed"`

-
  ```
  try:
    # code goes here, probably something w/ an assert
    # or something that might throw an exception otherwise
  except ExceptionType as e: # technically optional
      # what we do if an exception happens
      # can use e as variable, e.g. print(e)
  else: # optional
      # what we do if no exceptions happen
  finally: #optional
      # we do this whether or not there's an exception
      # often it's cleanup
  ```

-
  ```
  with open('filename') as variable_name:
    # do things with the file
  ```

Times to use exceptions:

- file not found
- DB won't connect
- error that should be logged but program shouldn't stop
- a function won't be able to perform as expected
- checking for weirdness (an assert to make sure all is as expected)
- to make termination easy (`with`/`as`, or catching errors w/ `finally`)

If you don't catch (`except:`) an exception, it'll bubble to the top (the "default exception handler") and crash your program, printing to console.