Ling 555 — Programming for Linguists

Python — Handling exceptions

Robert Albert Felty

Speech Research Laboratory
Indiana University

Oct. 29, 2008

Prereqs

homework
exceptions
Resources

While we are waiting

Please download the following files from:
http://robfelty.com/teaching/L555Fall2008/resources/py/
sortDictExcept.py, raising.py, nestedExcept.py

```
for file in
{sortDictExcept.py,raising.py,nestedExcept.py};
do curl -o $file
http://robfelty.com/teaching/L555Fall2008/
resources/py/$file; done
```

For Wednesday:

Read Chapter 8 on More Handling Exceptions

Outline

homework
exceptions
Resources

1. Homework questions and comments
   - comments
2. Exceptions
   - definition
   - basic
   - type-checking
   - catching specific error types
   - multiple except clauses
   - Improving sortDict
   - Raising exceptions
   - catching objects
   - Nesting exceptions
   - Practice
3. Resources

homework
comments
exceptions
Resources

Doc-strings

- Docstrings should be long strings with """
- They should start with a one sentence description in the imperative, like *Return a list of sorted values*
- By using long strings, they can be formatted however you want
- More info available online - look at the links at: delicious.com/robfelty/L555

Doc-strings

homework
comments
exceptions
Resources

Example

```
def foo():
    """ Print hello

    Here I can add some more information
     about the function (or method)
    I can do lists like so:

    * an item

      * a sub item
    """
    print "hello"
```

What is an exception?

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Definition

An exception is basically an error. However, instead of just having our program crash, we can anticipate some exceptions, and catch them.

mantra

It's easier to ask forgiveness than permission.
This is referred to as the *Look before you leap* programming paradigm.

Basic exception handling

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Example

```
foo='bar'
try:
    foo=float(foo)
except:
    pass
```

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Type checking IS GENERALLY BAD

Instead of using exception handling we could use type checking (input validation)

Example

```
foo='bar'
if type(foo)=='int'
    foo=float(foo)
```

Extra code execution!

With type checking, we always end up executing more code. Also, there still might be exceptions we hadn't thought of.

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Catching specific errors

Python defines many types of errors. Generally we should specify which type of error we want to catch.

Example

```
try:
    foo=float(foo)
except ValueError:
    pass
# to see all possible built-in
  exception types:
import exceptions
dir(exceptions)
```

multiple except clauses

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

We can handle different types of exceptions separately

Example

```
try:
    bar=20/float(foo)
except ValueError:
    print "foo must be a number"
except ZeroDivisionError:
    print "foo must be non-zero"
except:
    print "something else went wrong"
```

Improving sortDict

homework

exceptions

definition

basic

type-checking

error types

multiple except

sortDict

raising

catching objects

nesting

Practice

Resources

Let's take a look at sortDictExcept.py

Raising exceptions

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Definition

Exceptions need not be handled immediately. They can be raised, and handled later

Let's look at raising.py

Catching objects

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

You can catch more than one exception at a time:

```
try:
    bar=20/float(foo)
except (ValueError, ZeroDivisionError):
    print "bad input"
```

You can also catch the Exception object:

```
try:
    bar=20/float(foo)
except (ValueError,ZeroDivisionError),e:
    print "bad input: ", repr(e)
```

Nesting exceptions

homework
exceptions
definition
basic
type-checking
error types
multiple except
sortDict
raising
catching objects
nesting
Practice
Resources

Definition

Like pretty much anything else (loops, conditionals, lists, dictionaries), we can have nested try, except blocks

Let's look at nestedExcept.py

homework

exceptions

definition

basic

type-checking

error types

multiple except

sortDict

raising

catching objects

nesting

Practice

Resources

Practice

Look at homework 7, and see how you could improve your code to read in the celex into a dictionary of dictionaries, converting strings to ints where appropriate

homework
exceptions
Resources

Resources for you:

Some of the examples we have covered in class today can be found on the website at:
http://robfelty.com/teaching/L555Fall2008/resources/py
The files from today are:

1. sortDictExcept.py
2. raising.py