Statistical Programming with Python

Pepe García

2020-04-20

Statistical Programming with Python

Error handling

Error handling

Things can go wrong when programming. Today we'll learn how to deal with errors, and how to produce them ourselves.

Error handling

Can you name cases in which your program went wrong?

Trying to read a closed file

File not found

Type error

Key error

What are exceptions

Exceptions, or **errors**, are exceptional events that may happen in our program. They may happen because of different reasons

What are exceptions

```
def divide(a, b):
    return a / b
```

Can you spot the potential **exception** in the following function?

What are exceptions

By default, if we don't do anything, an exception terminates the program.

Recovering from exceptions

Python provides a way of handling exceptions, using the **try-except** block.

```python
try:
    do_something()
except:
    recover()
```

Recovering

Let's recover from a simple exception by printing a message

Matching exceptions

We can match particular exceptions in the except clause, this way, only the kind of exception we matched will be handled

```
try:
    do_something()
except ValueError:
    recover()
```

Recovering

Let’s modify the previous example

Matching exceptions

We can also match more than one exception, or different ones:

```python
try:
    do_something()
except (ValueError, ZeroDivisionError):
    pass
except TypeError:
    pass
```

Raising exceptions

Exceptions are a mechanism that we can use ourselves too.

```python
if len(phone_number) < 9:
    raise ValueError("Phone number is too short")
```

Raising exceptions

Exceptions are a mechanism that we can use ourselves too.

Exercises

~~~
1. Modify the mini-etl exercise and try to control all
   errors that might occur in there

2. You're given the following function.  control all
   errors that it may raise when being executed.

```
def write_division_to_file(filename, dividend, divisor):
    with open(filename, "r") as file:
        file.write(dividend / divisor)

    print("it didn't explode :D")
```
~~~