Programming fundamentals with Python

Pepe García

2020-04-20

Programming fundamentals with Python

Plan for today

OOP: Inheritance

Creating our own exceptions

More advanced error handling

Inheritance

Inheritance

Inheritance is a mechanism by which classes *inherit* methods and attributes from other classes

Inheritance

classes and their subclasses have a **is a** relationship, and almost all things that have that relationship can be expressed using inheritance

Inheritance

```
class ClasName(ParentClass):
    pass
```

Inheritance

Example: Animals

question: Can you think of any other **is a** relationship that we can try to do together?

Method resolution

when calling a **method** in an object, python searches for the **method** in the object's class and, if not found, it goes up the class hierarchy

Method resolution

Vehicle

- run()
- stop()

Car

- open_trunk()

SportsCar

- run_a_lot()

Creating exceptions

Creating our own exceptions is really simple, we just need to create a new class that **inherits Exception**

Creating exceptions

Example: Creating our own exceptions

Creating exceptions

Exercise: imagine you're a programmer doing the validation of a form. Create all the exceptions that come to your mind related to the validation of the fields

Advanced error handling

in last session we saw how the **try-except** block helps us run code and **handle** the exceptions that may happen there.

Advanced error handling

```
try:
    file = open("data.txt")

    for line in file:
        print(line)

except Exception:
    print("file not found error")
```

Advanced error handling

one cool features of try-except blocks is that we can put more than one **except** part.

When using more than one **except**, we put one for each type of exception we want to handle:

Advanced error handling

```
try:

    file = open("data.txt")

except ArithmeticException:
    print("an arithmetic exception occurred")

except FileNotFoundError:
    print("no file named data.txt exists")
```

Advanced error handling

Create a program that reads the data from a file (data.txt) and prints each line multiplied by two.

Control all exceptions that come to your mind

Exercises

Exercise 1

Create a **Polyhedron** class, and two classes **Triangle** and **Circle** that inherit from it.

Make **Triangle** and **Circle** have an **area** method that return their respective areas

Exercise 1

Change your ecommerce example so:

You create a class **Product** from which the classes **Item** and **Service** will inherit

Exercise 3

Create a function to **copy files**.

The function must receive two names (origin and destination, for example), and copy the contents of origin into destination.

You'll need to **investigate** how to write files for this exercise