Final Exam

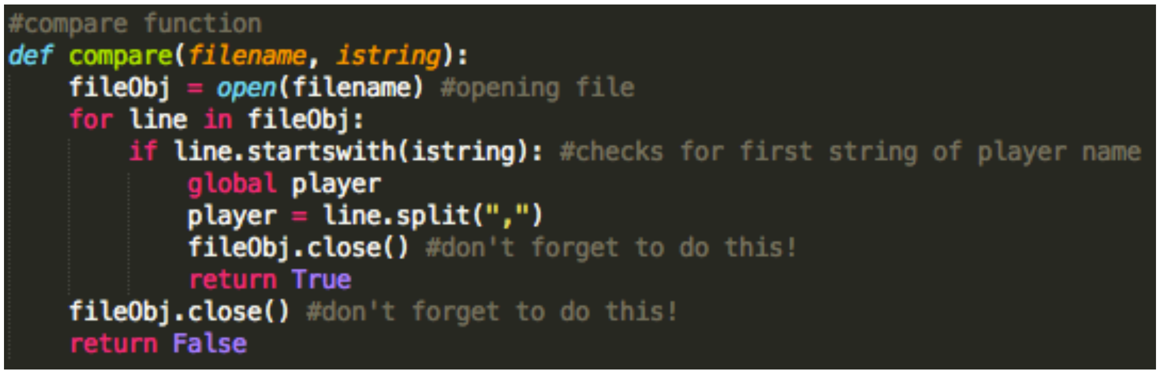
Matthew Wang

Y2 Compsci Exam

After re-imaging our computer, we can start solving our problem. To do some problem solving, we can try to segment the problem!

Compare

The first thing we need to do is make a compare() command. Let's get started on that.



```
#compare function
def compare(filename, istring):
    fileObj = open(filename) #opening file
    for line in fileObj:
        if line.startswith(istring): #checks for first string of player name
            global player
            player = line.split(",")
            fileObj.close() #don't forget to do this!
            return True
    fileObj.close() #don't forget to do this!
    return False
```

We've made two parameters for compare, the filename and the input string (istring). We'll pass this on to the compare function every time we run it!

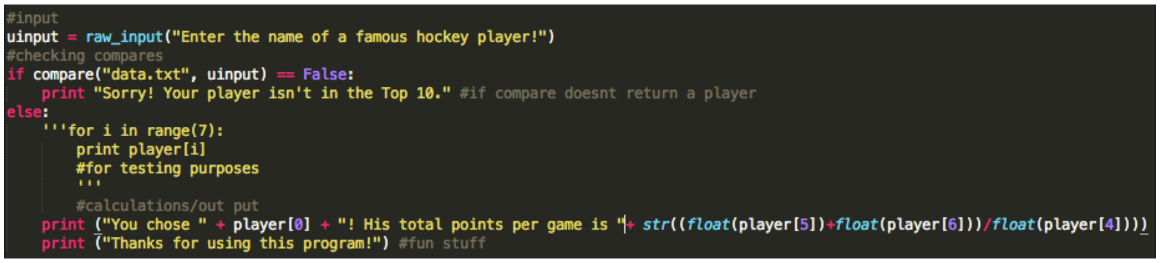
After opening the file, we'll check every line in the file with a for loop! In every line, we check if that line starts with our input string, or istring, using the startswith() function.

If it does, we use the split function, removing all those pesky commas and splitting each type of data (name, position, etc.) into a different element of an array. Then, it closes the file (always do that), and finally returns True, breaking the program.

If it doesn't, it continues going through all the lines. If it never gets anything, it'll close the file, and return false.

Main

Next, we have to make the actual meat of the program, that takes the compare input and does stuff with it!



```
#input
uinput = raw_input("Enter the name of a famous hockey player!")
#checking compares
if compare("data.txt", uinput) == False:
    print "Sorry! Your player isn't in the Top 10." #if compare doesnt return a player
else:
    '''for i in range(7):
        print player[i]
        #for testing purposes
        '''
        #calculations/out put
    print ("You chose " + player[0] + "! His total points per game is "+ str((float(player[5])+float(player[6]))/float(player[4])))
    print ("Thanks for using this program!") #fun stuff
```

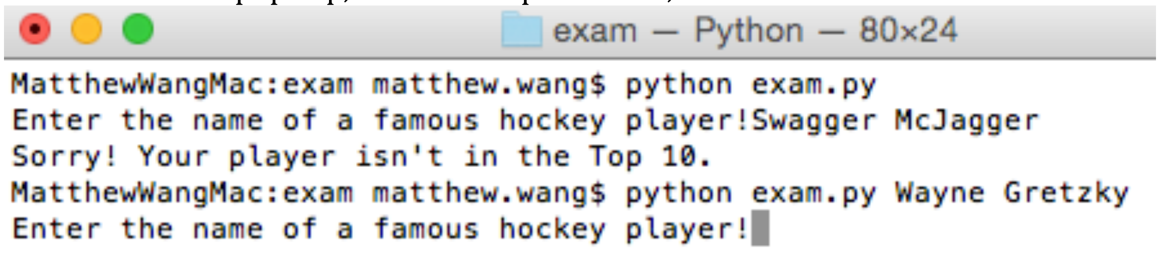
First, we'll use the raw_input command to get a user input. After that, we'll check whether or not compare returns False; if you remember, we programmed it to return False if a player isn't found. Then, we apologize (like a true Canadian), and move on.

If compare doesn't return False, then it means that we've found something! Instead, we'll print a nice String, with some numbers sprinkled in. Because we split the player's info into different parts of an array, we just call the part of the array we need for each part of the final output. Once that's done, we thank the user, and we're done!

We've done some commenting, to make sure that other people reading our code have some idea of what it does. It's good coding practice to do so.

Debugging

Awesome! Let's check if the code works! Debugging is an important exercise in coding; it's essential to making your code work, and also to better understand your code. There are three steps to debugging:
1. Find the problem. We can figure out the problem using error messages.
2. Try to fix the problem! Let's take a look at what we get in these error messages, and try to use them.
3. If another one pops up, rinse and repeat. If not, we're done!



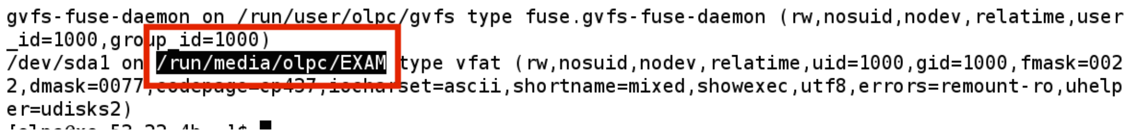
```
MatthewWangMac:exam matthew.wang$ python exam.py
Enter the name of a famous hockey player!Swagger McJagger
Sorry! Your player isn't in the Top 10.
MatthewWangMac:exam matthew.wang$ python exam.py Wayne Gretzky
Enter the name of a famous hockey player!
```

Oh wow, lucky us! No errors! If there are ever problems though, we'll know how to deal with them. Now, we can work on moving our code to the OLPC!

OLPC

Now that we've booted up our little green cute computer, let's try to get our code onto the computer. To do this we'll need to use a bit of our Linux knowledge. Linux is an operating system, which includes a command line interface, so we can tell the computer what to do. Let's go to the terminal to start giving some instructions!

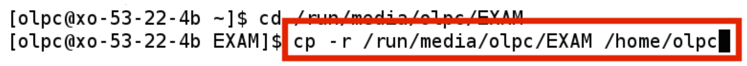
The first thing we're going to do is use the mount command. Mount tells you, among other things, what devices are connected to the OLPC, including our USB! Let's use the command.



```
gvfs-fuse-daemon on /run/user/olpc/gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,relatime,user
_id=1000,group_id=1000)
/dev/sda1 on /run/media/olpc/EXAM type vfat (rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=002
2,dmask=0077,codepage=cp437,iocharset=ascii,shortname=mixed,showexec,utf8,errors=remount-ro,uhelp
er=udisks2)
[illegible]
```

Awesome! We've found the directory that our USB is on, /run/media/olpc/EXAM

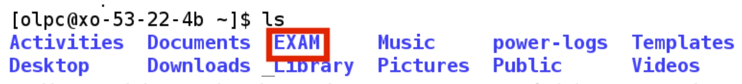
Let's put that to use. We want to copy our files from that directory to a workspace.



```
[olpc@xo-53-22-4b ~]$ cd /run/media/olpc/EXAM
[olpc@xo-53-22-4b EXAM]$ cp -r /run/media/olpc/EXAM /home/olpc
```

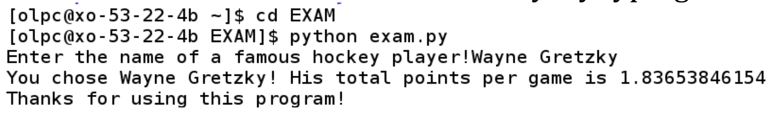
We'll use the cp command, which copies files from one directory to another. We'll also add the –r to the cp, which copies the entire directory! Convenient!

Now, let's check to make sure that our EXAM folder is actually in /home/olpc. We can, using the ls command. ls tells you every file that's in your current directory.



```
[olpc@xo-53-22-4b ~]$ ls
Activities  Documents  EXAM     Music     power-logs  Templates
Desktop     Downloads  Library  Pictures  Public      Videos
```

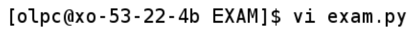
Well would you look at that! Our Exam folder is in there. Let's take a look, using the cd command. We can "cd" to a directory, by typing it in.



```
[olpc@xo-53-22-4b ~]$ cd EXAM
[olpc@xo-53-22-4b EXAM]$ python exam.py
Enter the name of a famous hockey player!Wayne Gretzky
You chose Wayne Gretzky! His total points per game is 1.83653846154
Thanks for using this program!
```

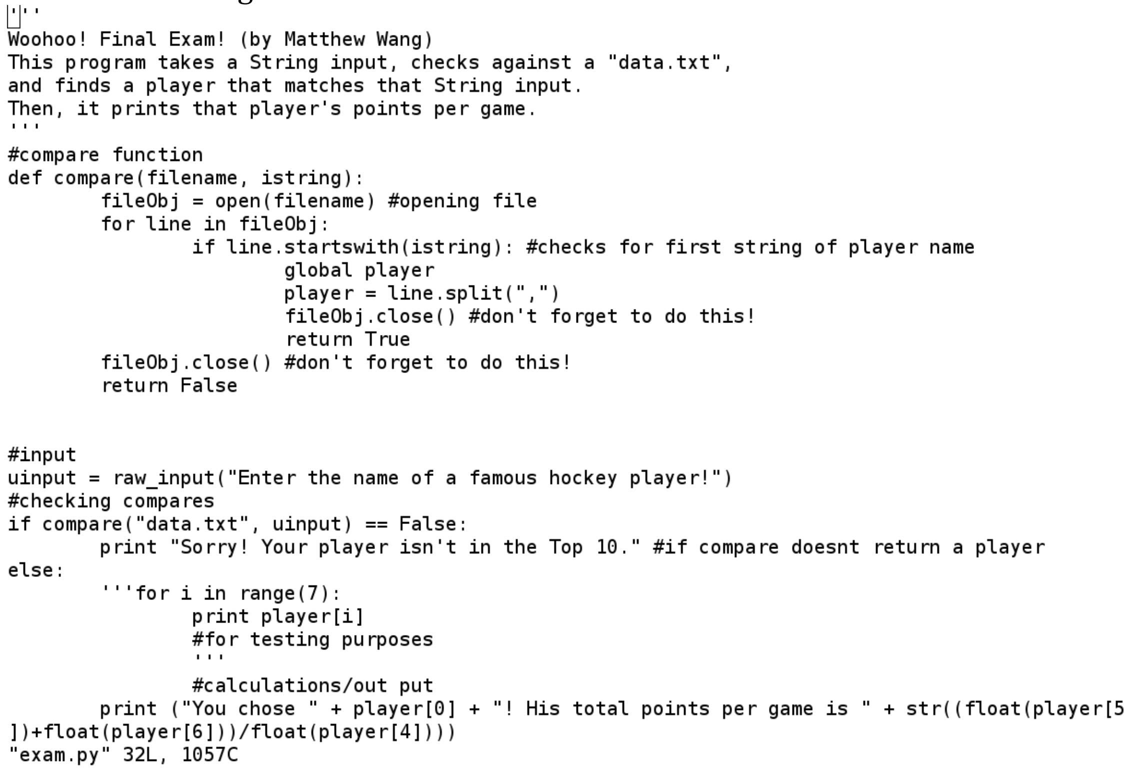
After that, we can run our exam.py python file, with the python command.
It works! Awesome!

To quickly prove that it's functional code, we can check it using the vi command. The vi command opens vi editor, a really lightweight text editing tool that doesn't need a mouse.



```
[olpc@xo-53-22-4b EXAM]$ vi exam.py
```

The vi command gives us this!



```
'''
Woohoo! Final Exam! (by Matthew Wang)
This program takes a String input, checks against a "data.txt",
and finds a player that matches that String input.
Then, it prints that player's points per game.
'''
#compare function
def compare(filename, istring):
        fileObj = open(filename) #opening file
        for line in fileObj:
                if line.startswith(istring): #checks for first string of player name
                        global player
                        player = line.split(",")
                        fileObj.close() #don't forget to do this!
                        return True
        fileObj.close() #don't forget to do this!
        return False


#input
uinput = raw_input("Enter the name of a famous hockey player!")
#checking compares
if compare("data.txt", uinput) == False:
        print "Sorry! Your player isn't in the Top 10." #if compare doesnt return a player
else:
        '''for i in range(7):
                print player[i]
                #for testing purposes
                '''
                #calculations/out put
        print ("You chose " + player[0] + "! His total points per game is " + str((float(player[5
])+float(player[6]))/float(player[4])))
"exam.py" 32L, 1057C
```

Well, looks like we're done here! We've created functional code, put it onto our target platform, and made sure it runs smoothly. Until next time!