**Challenge 16**

[www.pythonchallenge.com/pc/return/romance.html](http://www.pythonchallenge.com/pc/return/romance.html)



This has to be about web cookies – which I know nothing much about. Title of the page is ‘eat?’ and picture is ‘cookies.jpg’

Also, those little guys at the bottom are from an early challenge (the linked list one I think.) Yes # 4 -- **. . . pc/def/linedlist.php**

The request module can show info about cookies for a page:

*<RequestsCookieJar[<Cookie info=you+should+have+followed+busynothing... for .pythonchallenge.com/>]>*

Remembering how we did challenge 4, it had something like *.php?nothing=12345* and then you made a chain. So this must be *?busynothing=12345*.

Trying that gets:

*If you came here from level 4 - go back!  
You should follow the obvious chain...  
and the next busynothing is 44827*

OK, this seems to be on the right track. Running basically the same code as challenge 4, there is a chain at it ends with 83051 and message ‘that’s it.’ Since this challenge has a picture of cookies, there must be cookie information on these links in the chain. Yes, there is **requests.get(url).cookies[‘info’]** gives a list of a single character for each. Stringing them together is a long thing that looks like this:

**BZ**h91AY%26SY%94%3A%E2I%00%00%21%19%80P%81%11%00%AFg%9E%A0+%00hE%3DM%B5%23%D0%D4%D1%E2%8D%06%A9%FA%26S%D4%D3%21%A1%EAi7h%9B%9A%2B%BF%60%22%C5WX%E1%ADL%80%E8V%3C%C6%A8%DBH%2632%18%A8x%01%08%21%8DS%0B%C8%AF%96KO%CA2%B0%F1%BD%1Du%A0%86%05%92s%B0%92%C4Bc%F1w%24S%85%09%09C%AE%24%90

The starting BZ is the clue here. Python has a module bz2. It compresses and decompresses data. I tried playing bz2 compressing and decompressing and got similar looking string except everything for bz2 \xx while challenge shows %xx. *Frustrated. . . so left it for a while.*

Finally, watched Hammond’s youtube and found out I needed to take the string from the program and use urllib.parse on it. In fact the doc for urllib.parse says

‘urllib.parse.**unquote**(*string*, *encoding='utf-8'*, *errors='replace'*)

Replace %xx escapes with their single-character equivalent. The optional *encoding* and *errors* parameters specify how to decode percent-encoded sequences into Unicode characters, as accepted by the [bytes.decode()](https://docs.python.org/3/library/stdtypes.html" \l "bytes.decode" \o "bytes.decode) method.

Not sure I ever would have found that. (plus I thought I was trying to avoid urllib … oh well) Anyway, that and bz2 worked an show the message:

*is it the 26th already?*

*call his father and inform him that  
"the flowers are on their way".  
he'll understand.*

So, this must be harking back to the Mozart challenge. Mozart’s father is a famous violinist, *Johann Georg Leopold Mozart*

Calling him (that was challenge 13) – Johann and Georg didn’t work. But Leopold did and printed 555-VIOLIN

Putting violin in the .html doesn’t quite do it:

*no! i mean yes! but ../stuff/violin.php.*

That web page shows a picture of Mozart’s father with title ‘it’s me. What do you want?

So we want to post a request telling him ‘the flowers are on their way’ and find out what he said. With lots of trial and error (still learning request module) printing this

requests.post(url, headers= {**'Cookie'**: **'info='** + message}).content

seemed to do the trick. In the contents is the clue ‘don’t you dare forget the balloons.

FINALLY, putting balloons in the url goes to the next challenge.