Homework: Melon Raffle

Emails

Mel is heading out of town for the rest of the week, and has a few tasks that need to get done tonight.

To: jrdev48@ubermelon.co <Junior Developer #48>
From: mmelitopolski@ubermelon.co <Mel Melitopolski>
Subject: On my way out
Date: Wed Sept. 30, 2015 5:50:59 PM PDT

Hello,

We've been writing a new application for managing customers
and running a weekly raffle of a free melon.

I'm not sure what a "virtual environment" is (sounds like
one of those things lefty nonprofits worry about), but,
according to CTO Today! Magazine, we need to make sure that
developers are setting up their environment correctly
to manage libraries and split up files as needed.

Get up to speed on implementing "virtual environments"
on this project so we don't have any conflicts or missing
requirements. Make sure that you have a ***.gitignore*** file so your
environment doesn't end up on GitHub.

We just poached Squysh's 16-year-old wonderkid engineering
manager, Janet, to optimize our process. She says everyone
should install "IPython" as well. You won't need it now,
but try it so you know how to use it later. Apparently, you'll use
something called "pip", inside of your "virtual environment".
Oh, and my golf buddy says you should "freeze" this, so others
can get the same setup. *Pip*? *Freeze*? *Virtual environment*?
This is starting to sound like a science fiction musical if
you ask me! But we're paying a lot for this advice so make
it happen.

Mel

P.S. I'm not clear what this "virtual environment" is, but
don't think we're going to buy you any of those fancy

```
goggles. All of our discretionary income has gone into
my new shiny titanium golf clubs.
```

And then this email showed up:

```
To: jrdev48@ubermelon.co <Junior Developer #48>
From: mmelitopolski@ubermelon.co <Mel Melitopolski>
Subject: One more thing
Date: Wed Sept. 30, 2015 5:55:59 PM PDT

Oh, and make sure you take a look at our app's code.
There is a bug keeping things from working just right.

It's all in this file, raffle.py. To use it, you'll need
to go into Python and "import" it, like so:

$ python3
>>> import raffle
>>> raffle.run_raffle()

That causes a problem, though --- some sort of
"import error", whatever that means.

While you're at it, it's annoying that we have to go into
a Python shell to run our raffle. Can't we just have this
happen when we run the script, like so?

$ python3 raffle.py
Tell Blah ... that they've won

- Mel
```

And this one…

```
To: jrdev48@ubermelon.co <Junior Developer #48>
From: mmelitopolski@ubermelon.co <Mel Melitopolski>
Subject: One more last thing
Date: Wed Sept. 30, 2015 5:59:59 PM PDT

Janet mentioned one more thing: it seems bad that
we have the code about customer stuff in the same file as
the code about the raffles. She says that was bad for
"modularity". (Boy, she uses weird words!)

She created a new version of our app, in version2, where
she split things into two files. She had to run out before
she could finish, though. Can you finish her work here,
so that things work like they did before?

Mel
```

Let’s hope Mel doesn’t have WiFi, wherever he’s golfing.

To Do

- Download all necessary files from the fellowship website
- *cd* into this homework assignment’s directory and set up a virtual environment. **Windows users: make sure you use the command:**

  ```
  $ virtualenv env --always-copy
  ```

- Activate your virtual environment. (Hover below for a reminder on how to do this)

  ▼ *Click to hide*

  ```
  $ source env/bin/activate
  ```

- Tell ***git*** to ignore your virtual environment

  ▼ *Click to hide*

  Your ***.gitignore*** file should look like this:

  .gitignore

  ```
  env
  ```

- Install IPython

  ▼ *Click to hide*

  ```
  $ pip3 install ipython
  ```

- Output your dependencies to a file called ***requirements.txt***

▼ *Click to hide*

```
$ pip3 freeze > requirements.txt
```

- Fix the import error in ***raffle.py***
- Finish the work on version 2 — you'll find the project files in ***version2/***

> **Note:** *customers.py*
>
> The code in ***customers.py*** defines a class called ***Customer***. If this is unfamiliar to you — don't worry! — **you only need to edit the code in *raffle.py***.

Solution

Click here to view the solution.