Part 2: Quote Picker

Walkthrough of the pseudo-random quotation picker.

Quote Picker Setup

First, within your `network_project` directory, create the following directory, and file within that directory

```
(NetworkProj) $ mkdir talkback
(NetworkProj) $ touch talkback/quote_picker.py
```

Go ahead and open up `quote_picker.py` within your text editor.

We want to select from our list of quotes at random, so we use the `random` library from Python's standard library and import the choice function:

```
from random import choice
```

This module does not generate or select quotes at "true" randomness. The `random` (http://docs.python.org/2/library/random.html) module in Python generates pseudo-random number based off of the Mersenne twister (http://en.wikipedia.org/wiki/Mersenne_twister). The `choice` method uses `random()` to generate a pseudo-random number between 0 and 1, multiply it by the number of items in the list to choose from, and index into the list.

QuotePicker class

Next, we need to define the behavior of how we want to grab a quote that our bot uses. We'll create a class, `class QuotePicker(object):` with two functions, `__init__` and `pick`:

```
# <--snip-->

class QuotePicker(object):

    def __init__(self, quotes_filename):
        """Initialize our QuotePicker class"""
        # <--snip-->

    def pick(self):
        """Return a random quote."""
        # <--snip-->
```

We want to initialize our class with the file that we plan to pull a quote from. If you remember from earlier tutorials, Python has a few keywords and built-in functions for file input/output. We open our quote file using the `with` keyword, which will implicitly take care of closing our quote file when we are done with the `with` loop:

```
# <--snip-->

def __init__(self, quotes_filename):
    """Initialize our QuotePicker class"""
    with open(quotes_filename) as quotes_file:
        # <--snip-->

# <--snip-->
```

Within the `with` keyword, we label the opened file, `open(quotes_filename)` as `quotes_file`. We then read each line item in the `quotes_file` (each quote is on its own line) with another built-in function, `readlines`, and assign it to `self.quotes`. This variable, `self.quotes`, will be accessible to the rest of the `QuotationPicker` class, including our next function, `pick`.

```
# <--snip-->

def __init__(self, quotes_filename):
    """Initialize our QuotePicker class"""
    with open(quotes_filename) as quotes_file:
        self.quotes = quotes_file.readlines()

# <--snip-->
```

For our `pick` function, we use the `choice` function we imported from `random` to randomly pick a quote from `self.quotes` variable we initialized in `__init__`. To be safe, we use a built-in string method, `strip`, to clean up any extraneous blank spaces and whitespace characters that may be before or after the quote itself.

Rather than creating a variable and assigning it `choice(self.quotes).strip()` and then returning the variable, we simplify it by just returning `choice(self.quotes).strip()`.

```
# <--snip-->

def pick(self):
    """Return a random quote."""
    return choice(self.quotes).strip()

# <--snip-->
```

The whole `quote_picker.py` module put together:

```
from random import choice


class QuotePicker(object):

    def __init__(self, quotes_filename):
        """Initialize our QuotePicker class"""
        with open(quotes_filename) as quotes_file:
            self.quotes = quotes_file.readlines()

    def pick(self):
        """Return a random quote."""
        return choice(self.quotes).strip()
```

That was pretty simple, right? I hope! Now let's move on!

← Part 1: Settings (/networks/part-1/) Part 3: Bot Module → (/networks/part-3/)