CS173 Class Activity – Apr 5, 2019

Regular Expressions

Source: https://www.nltk.org/

Answer the following questions below. Please practice writing and use full sentences.

1. Learn Python.... go you have 10 minutes! (Yeah it should be that easy... just teasing. ;)

2. Open a terminal and get on bolt, and create a folder for CS173 to organize your stuff:

   ```
   > ssh USERNAME@bolt.cs.ucr.edu
   > mkdir ~/cs173
   ```

   Did it work?

   If not, ask your neighbor for help. Now did it work?

   If not, ask the professor for help. Now did it work?

   If not, ask Google for help offline and see the TA during discussion.

3. Read about NLTK (see link above). Spend 10-15 minutes learning what NLTK can do.

4. Install NLTK (don't use sudo on bolt, it won't let you since you are a mere peon, like me):

   ```
   pip install --user nltk
   ```

   ```
   pip install --user numpy
   ```

   The "--user" option installs it only for yourself, a peon user. The sudo option, on the other hand installs it system-wide, for all users. That is something which Victor would never allow you to do. His bolt is his precious and "You shall not pass!" Stay in your sandbox.

   That last one is pronounced "num pee" not "num pie" .... (LOL, I'm just kidding. I've heard people pronounce it that way and it makes me snort my milk. So if you want me to giggle... say, "num pee.")

5. Test that it worked:

   launch python shell

   ```
   > python
   .........
   >>> import nltk
   >>> import numpee
   ```

   LOL -- at least now you know what it looks like if it DIDN'T work...

6. Now use NLTK to download some data. In NLP, usually when we say "data" we mean text corpora.

   How do you download NLTK data -- write the commands below:

   Which ones did you choose to download?

   Where to did the file(s) download?

7. Let's look at the brown dataset, hopefully you at least downloaded that.

   How are the data organized?

   What do the files look like?

   What do you think it means?

8. Try the basic activity on https://www.nltk.org/data.html :

   ```
   >>> from nltk.corpus import brown
   >>> brown.words()
   ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
   ```

   Does yours look the same?

   What's different? Why?

9. Go back to homepage https://www.nltk.org/ :

   Practice the tokenizing example.

   Did you have to download something extra? What? Why?

   What is tagged doing? What does that mean?

   What are named entities?

   Did t.draw() fail for you? Why do you think that is? How could you make it work?

Now, let's practice regular expressions in Python.

10. Download mobydick.txt to your cs173 folder, HINT: wget is a nice little tool.

    ```
    > wget URL ~/cs173/
    ```

    Verify it's there:

    ```
    > less ~/cs173/mobydick.txt
    ```

11. Back to python, and play with regular expressions:

    ```
    > python
    ....
    >>> import re
    ```

    Load the data (the corpus, Moby Dick), all of it, why not, Victor will love us sucking up all his RAM:

    ```
    >>> f = open("mobydick.txt", "r")
    >>> data = f.readlines()
    ```

    What did that do exactly? What data type is "data"?

    What did you notice about newlines? Is that okay, do you want to get rid of those?

    Here's a nice trick about Python you don't easily get from C++ — C++ sucks rocks, btw, IMHO ... not that my opinion matters. You can have your own opinion, and that's okay.

    ```
    >>> data = map(lambda x: x.strip(), data) # lambda calculus rocks
    ```

    For those just learning python, here's a loop doing the same thing:

    ```
    newdata = list()
    for x in data:    # smart, collection and iterator savvy
      x = x.strip()   # no semicolons YAY! and MANDATORY clean indentation, double YAY!
      newdata.append(x)
    ```

What is a lambda? What does that do?

Here is a "pythonic" shorthand for the above lamda + map:

```
>>> [x.strip() for x in data]
```

Do you want to turn the corpus into a giant string? Here's how, more "pythonic" tricks:

```
>>> corpus = "\n".join(data)
```

We love a good corpus! Strings are cool.

12. Find all Ishmael's:

```
>>> re.findall(r"\bIshmael\b", corpus)
```

How many are there?

```
>>> len (re.findall(r"\bIshmael\b", corpus) )
```

13. Play, learn, read.... enjoy... Regular Expressions:

https://www.guru99.com/python-regular-expressions-complete-tutorial.html
https://docs.python.org/3/howto/regex.html

https://lmgtfy.com

Your life is that much better now... **you're welcome!**