ENGL 516X: Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

6 March 2018

Class outline

- Web applications with Bottle - continuation
- Small exercises and practice
- Midterm feedback

Web application development with Bottle

Web-applications with Bottle

- `@route`
- `@get`, `@post`
- Python functions that implement the logic (same as in regular programs)
- request - to read input data from browser into Python code
- html (to render stuff on the browser to the user)
- run()

Exercise-1 from previous class

- write a Python program using Bottle, that shows two text fields and a button in the initial screen: First Name, Last Name:
- In the program, have a function, that gets called when the user submits this information, which takes these two values, and returns - "Hello" + FirstName + LastName on the screen

Emily’s Solution

```
from bottle import get, post, request, route, run

@route("/")
@get("/name") # or @route('/name')
def get_name():
    return '''
            <form action="/name" method="post">
                First Name: <input name="firstname" type="text" />
                Last Name: <input name="lastname" type="text" />
                <input value="Name" type="submit" />
            </form>
            '''
@post("/name") # or @route('/name', method='POST')
def hello_user():
    firstname = request.forms.get("firstname")
    lastname = request.forms.get("lastname")
    return "Hello", " ", firstname, " ", lastname

run()
```

Exercise-2 from previous class

- Write a Python program using Bottle, that shows two textfields and a button in the initial screen to take two strings as input
- In the program, have a function, that gets called when the user submits this information, which takes these two values, and does the following:
  1. checks if the two words are permutations of each other (i.e.,god-dog; spam-maps, program-magropr etc.)
  2. Shows a message: "The words are permutations" or "The words are not permutations" depending on the result of the check.
  3. Note: Inputs don't have to be valid words. Any word stringsare okay, and you can ignore punctuation, spaces, numbers etc.

Tim’s Solution

```
from bottle import get, post, request, route, template, run

def permutation(first, second):
    if len(first) != len(second):
        return False
    else:
        return ' '.join(sorted(first)) == ' '.join(sorted(second))

@route('/')
@route('/hello')
def hello():
    name = 'Guest'
    return template('Hello {{name}}', name=name)

@get('/login') # or @route('/login')
def login():
    return '''
        <form action="/login" method="post">
            First string: <input name="first" type="text" />
            Second string: <input name="second" type="text" />
            <input value="Submit" type="submit" />
        </form>
    '''

@post('/login') # or @route('/login', method='POST')
def do_login():
    first = request.forms.get('first')
    second = request.forms.get('second')
    if permutation(first, second) == True:
        return "The strings are permutations."
    else:
        return "The strings are not permutations."

run()
```

"Template" files in bottle

- files with .tpl extension (you can type in pycharm or notepad and save as somename.tpl)
- Why? Avoiding typing of all html in the program itself, and storing it separately.
- Good thing: While most of the html is static, we can actually modify it based on program output
- Let me modify Tim's solution and show this.

```
from bottle import get, post, request, route, template, run

def permutation(first, second):
    if len(first) != len(second):
        return False
    else:
        return ' '.join(sorted(first)) == ' '.join(sorted(second))

@get('/')
def enter():
    return template('homepage.tpl')

@post('/check')
def checkpermutations():
    first = request.forms.get('first')
    second = request.forms.get('second')
    areornot = ""
    if permutation(first, second) == True:
areornot = "are"
    else:
areornot = "arenot"
    return template('resultpage.tpl', string1=first, string2=second, areornot=a

run()
```

Few exercises and discussion

What will the following things print?

-
  ```
  lst1 = ["this", "is", "a", "list"]
  print(lst1[10])
  print(lst1[10:])
  ```

What will the following things print?

-
  ```
  lst1 = ["this", "is", "a", "list"]
  print(lst1[10])
  print(lst1[10:])
  ```

-
  ```
  lst1 = [1,"a","b","cc",11]
  lst1.sort()
  print(lst1)
  ```

What will the following things print?

-
  ```
  lst1 = ["this", "is", "a", "list"]
  print(lst1[10])
  print(lst1[10:])
  ```

-
  ```
  lst1 = [1,"a","b","cc",11]
  lst1.sort()
  print(lst1)
  ```
- Will this work?

  ```
  a = 1
  b = 2
  a, b = a+1, b+1
  print(a)
  ```

Errors and types

- What errors do you see here?

```
dict1 = {"a":1, "b":2, "c":3}
print(dict1[a])
print(dict1[3])
```

Errors and types

- What errors do you see here?

  ```
  dict1 = {"a":1, "b":2, "c":3}
  print(dict1[a])
  print(dict1[3])
  ```
- What is a "Syntax Error"?
- What is a "Type Error"?
- What is a "Zero Division Error"?
- What is a "Index Error"?

Error Hierarchy in Python:
`https://docs.python.org/3/library/exceptions.html`

What is happening with this loop?

```
list1 = [1,2,3,4,5,6,7]
for item in list1:
  del[list1[item]]
  print(list1)
```

How many times will it run and what will be printed?

RegEx exercise: pluralize nouns

Write a small program now using regex, which takes a word as input and returns the plural word as output. Here are the rules to code:

- If a word ends in s, x, or z, add es to the end of the word.
- If a word ends in a consonant +y, add ies to the word. If a word ends in a vowel+y, add s in the end (vacancy is vacancies but day is days)
- If none of the above cases are valid for a word, just add s at the end of the word and be happy with that.
- Post your solution on the forum for today

(Your input should not be a number or a string with numbers, punctuation etc. It has to be an alphabetic string.)

Mid-term feedback - please fill it up

Next Class

- Topic: Final projects discussion
- Todo for you: Think about final projects and get back with your ideas - I will ask all groups to talk about their ideas.
- Look at the descriptions uploaded on Canvas
- Assignment 5 description