Discussion 6

DSC 80

2024-05-10

1. FA23 Final Exam Problem 7
2. SP23 Final Problem 1
3. WI23 Final Exam Problem 4
4. WI23 Final Exam Problem 5
5. Attendance

Section 1

FA23 Final Exam Problem 7

Problem

Alan set up a web page for his DSC 80 notes with the following HTML:

```
<html><body>
    <div id = "hero">DSC 80 NOTES</div>
    <div class="notes">
        <div class="notes">
            <p>Lecture 1: 5/5 stars!</p>
        </div>
        <div class="lecture notes">
            <p>Lecture 2: 6/5 stars!!</p>
        </div>
    </div>
    <div class="lecture">
        <p>Lecture 3: 10/5 stars!!!!</p>
    </div>
</body></html>
```

Parsing HTML

- What does `soup.find_all()` do? How do we use it to find all elements with a certain tag, like `div`, `p`, or `h1`?

Parsing HTML

- What does `soup.find_all()` do? How do we use it to find all elements with a certain tag, like `div`, `p`, or `h1`?
- What about finding specific instances of those tags?

Solutions

- `.find_all()` provides multiple keywords for specifying elements in the HTML structure: `id`, `class_`, `href`, etc.

Solutions

- `.find_all()` provides multiple keywords for specifying elements in the HTML structure: `id`, `class_`, `href`, etc.
- So, the blanks in the first two are just `p` and `div`, and the last one is: `'div', class_='lecture'`

Section 2

SP23 Final Problem 1

SP23 Final Problem 1

Consider the following line. Choose the regex pattern that, when filled in the blank, will return the desired matches.

```
re.findall(r'__(a)__', 'my cat is hungry, concatenate!,
                        catastrophe! What a cat!')
```

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does \b match?

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does `\b` match?
- What does `[a-z]*` match?

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does `\b` match?
- What does `[a-z]*` match?
- What does `\s` match?

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does `\b` match?
- What does `[a-z]*` match?
- What does `\s` match?
- What do the parentheses do?

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does `\b` match?
- What does `[a-z]*` match?
- What does `\s` match?
- What do the parentheses do?
- Solution: `([a-z]*)\scat\b`

Part 1

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['my', 'a']`.

- What does `\b` match?
- What does `[a-z]*` match?
- What does `\s` match?
- What do the parentheses do?
- Solution: `([a-z]*)\scat\b`
- Interpret as "0 or more lowercase characters, followed by a space, followed by the string `cat`, followed by a word boundary"

Other Options

- Option 1 would select `['', '']`
- Option 2 would select `['my cat', 'a cat']`
- Option 4 would select `['my cat', 'a cat']`

Part 2

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['concatenate']`:

- What does `.*` match?

Part 2

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['concatenate']`:

- What does `.*` match?
- What does `[a-z]+` match?

Part 2

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['concatenate']`:

- What does `.*` match?
- What does `[a-z]+` match?
- Solution: `[a-z]+cat[a-z]+`

Part 2

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['concatenate']`:

- What does `.*` match?
- What does `[a-z]+` match?
- Solution: `[a-z]+cat[a-z]+`
- Intepret as "1 or more lowercase characters, followed by the string `cat`, followed by 1 or more lowercase letters"

Other Options

- Option 1 would select `['my cat is hungry, concatenate!, catastrophe! What a cat']`
- Option 2 would select `['cat', 'concatenate', 'catastrophe', 'cat']`
- Option 4 would select `['cat', 'concatenate', 'catastrophe', 'cat']`

Part 3

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['cat', 'concatenate', 'catastrophe', 'cat']`:

- What does each option match?

Part 3

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['cat', 'concatenate', 'catastrophe', 'cat']`:

- What does each option match?
- What do each of these options have in common?

Part 3

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['cat', 'concatenate', 'catastrophe', 'cat']`:

- What does each option match?
- What do each of these options have in common?
- Solution: `\b[a-z]*cat[a-z]*\b`

Part 3

`my cat is hungry, concatenate!, catastrophe! What a cat!`

We want the output `['cat', 'concatenate', 'catastrophe', 'cat']`:

- What does each option match?
- What do each of these options have in common?
- Solution: `\b[a-z]*cat[a-z]*\b`
- Interpret as "word boundary, followed by 0 or more lowercase letters, the string `cat`, 0 or more lowercase letters, and a word boundary"

Other Options

- Option 1 would select `['my cat is hungry, concatenate!, catastrophe! What a cat!']`
- Option 2 would select `['my cat is hungry, concatenate!, catastrophe! What a cat!']`
- Option 4 would select `['concatenate']`

Section 3

WI23 Final Exam Problem 4

WI23 Final Exam Problem 4

```
s = '''
In DSC 10 [3], you learned about babypandas, a strict subset
of pandas [15][4]. It was designed [5] to provide programming
beginners [3][91] just enough syntax to be able to perform
meaningful tabular data analysis [8] without getting lost in
100s of details.
'''
```

Options

```
list1 = ['10', '100']
list2 = ['3', '15', '4', '5', '3', '91', '8']
list3 = ['10', '3', '15', '4', '5', '3', '91', '8', '100']
list4 = ['[3]', '[15]', '[4]', '[5]', '[3]', '[91]', '[8]']
list5 = ['1', '0', '3', '1', '5', '4', '5', '3',
         '9', '1', '8', '1', '0', '0']
```

Part 1

```
re.findall(r'\d+', s)
```

- What does `\d` refer to?

Part 1

```
re.findall(r'\d+', s)
```

- What does `\d` refer to?
- Which solution fits?

Part 1

```
re.findall(r'\d+', s)
```

- What does `\d` refer to?
- Which solution fits?
- Solution: `list3` – this looks for one or more digits anywhere in the string.

Part 2

```
re.findall(r'[\d+]', s)
```

- What do the brackets `[]` define?

Part 2

```
re.findall(r'[\d+]', s)
```

- What do the brackets `[]` define?
- Does that change the result?

Part 2

```
re.findall(r'[\d+]', s)
```

- What do the brackets `[]` define?
- Does that change the result?
- Solution: `list5`

Part 2

```
re.findall(r'[\d+]', s)
```

- What do the brackets `[]` define?
- Does that change the result?
- Solution: `list5`
- This is sort of a trick question: since `[]` defines a character class, the `+` sign is treated as a literal character, and therefore doesn't affect the match! And without a quantifier after the character class, this defaults to mean "one character of this class," so any single digit.

Part 3

```
re.findall(r'\[(\d+)\]', s)
```

- There's a lot of backslashes here – what are they doing?

Part 3

```
re.findall(r'\[(\d+)\]', s)
```

- There's a lot of backslashes here – what are they doing?
- What are the parentheses doing?

Part 3

```
re.findall(r'\[(\d+)\]', s)
```

- There's a lot of backslashes here – what are they doing?
- What are the parentheses doing?
- Solution: `list2` – we're matching one or more digits that are between brackets, or all the citation numbers.

Section 4

WI23 Final Exam Problem 5

WI23 Final Exam Problem 5

We're given an HTML document, and we want to find an expression that evaluates to `"verbal"`.

- `soup.find("scorerow").get("kind")`
- `soup.find("sat").get("ready")`
- `soup.find("scorerow").text.split(":")[0].lower()`
- `[s.get("kind") for s in soup.find_all("scorerow")][-2]`
-
```
soup.find("scorelist", attrs={"listtype":"scores"}
      ).get("kind")
```

Solution

- First: what does `.find()` do? `.find_all()`?
- How about `.get()`?
- How about `.text`?

Section 5

Attendance

Attendance

Once I give you a number, fill out the following Google form: https://forms.gle/JiNR7LsgK65ur99K6