

finding_complicated_patterns_solution

March 18, 2023

1 TODO: Finding email Addresses Revisited

In the cell below, our `sample_text` consists of a multi-line string with four different email addresses. Write a regular expression that is able to find all these email addresses. As usual, save the regular expression object in a variable called `regex`. Then use the `.finditer()` method to search the `sample_text` string for the given regular expression. Finally, write a loop to print all the matches found by the `.finditer()` method.

HINTS: Notice that all the characters before the `@` symbol only contain lowercase letters, underscores, and numbers. To match this part of the email address we can use the character set `[a-z_0-9]` followed by the `+` metacharacter, to account for the fact that all email addresses must have at least one character or more before the `@` symbol. The `+` metacharacter matches 1 or more repetitions of the preceding regular expression. For example, `ab+` will match `a` followed by any non-zero number of `b`'s, such as `ab` or `abb`, etc., but it will not match just `a`.

The `@` symbol is not a metacharacter so we can match it directly without the need of escaping it. Also, notice that the domain names contain lowercase letters, uppercase letters, underscores, and dashes. Again we can use the characters set `[a-zA-Z_-]` followed by the `+` metacharacter, to account for the fact that all domains must have at least one character or more after the `@` symbol. To match any dot (`.`), we need to use the backslash (`\.`) because the dot is a metacharacter. You can use the character set `[a-z]+` to match either `com`, `edu`, or `gov`.

To match the last email address you need to add an optional dot followed by another character set of only lowercase letters.

```
In [1]: # Import re module
import re

# Sample text
sample_text = '''
fake_email@fake-email.edu
fakeemail43@fake_email.com
fake891_email@fakemail.gov
52fake_email@FAKE_email.com.nl
'''

# Create a regular expression object with a regular expression that can match all
# the email addresses
regex = re.compile(r'[a-z_0-9]+@[a-zA-Z_-]+\.[a-z]+\.[a-z]+')
```

```
# Search the sample_text for the regular expression
matches = regex.finditer(sample_text)

# Print all the matches
for match in matches:
    print(match)

<_sre.SRE_Match object; span=(1, 26), match='fake_email@fake-email.edu'>
<_sre.SRE_Match object; span=(27, 53), match='fakeemail43@fake_email.com'>
<_sre.SRE_Match object; span=(54, 80), match='fake891_email@fakemail.gov'>
<_sre.SRE_Match object; span=(81, 111), match='52fake_email@FAKE_email.com.nl'>
```

```
In [ ]:
```