Regular expression in Python

prepared by E.C.Darsow | Fall 2018

A regular expression--or regexp--is an odd-looking sequence of characters and symbols which specify a textual pattern. A tool, such as the python re module, can search a given string for the pattern specified by the regular expression. Instances of the pattern are referred to as matches, which are passed back to the programmer for further manipulation.

See *https://docs.python.org/2/library/re.html*

EXAMPLE 1:

```
import re

testtext = '471 WALNUT  ST<br>PITTSBURGH, PA 15238'
splitlinelist = testtext.split('>')
# find zero or more decimals followed by zero or more characters which
are not <
exp = re.compile(r'\d*[^<]*')
adr = exp.search(splitlinelist[0])
street = re.sub(r' ', ' ', adr.group())
street = re.sub('  ', ' ', street)
print(street)
```

output: `471 WALNUT ST`

EXAMPLE 2:

```
import bs4 import BeautifulSoup

scrapedhtml = r'''<span class="Data" id="BasicInfo1_lblAddress">618 \
AIRBRAKE  AVE<br/>WILMERDING,     PA 15148-23423</span>'''

soup = BeautifulSoup(scrapedhtml, 'lxml')
magiclist = soup.span.contents
exp=re.compile(',\s*(\w\w)')
m = re.search(exp,magiclist[2])
print(m.group(1))
```

output: PA

You try!

Given the string: parid='0450L00350000000'

extract the string: 45-L35

Regular expression in Python

prepared by E.C.Darsow | Fall 2018

A regular expression--or regexp--is an odd-looking sequence of characters and symbols which specify a textual pattern. A tool, such as the python re module, can search a given string for the pattern specified by the regular expression. Instances of the pattern are referred to as matches, which are passed back to the programmer for further manipulation.

See *https://docs.python.org/2/library/re.html*

EXAMPLE 1:

```
import re

testtext = '471 WALNUT  ST<br>PITTSBURGH, PA 15238'
splitlinelist = testtext.split('>')
# find zero or more decimals followed by zero or more characters which
are not <
exp = re.compile(r'\d*[^<]*')
adr = exp.search(splitlinelist[0])
street = re.sub(r' ', ' ', adr.group())
street = re.sub('  ', ' ', street)
print(street)
```

output: `471 WALNUT ST`

EXAMPLE 2:

```
import bs4 import BeautifulSoup

scrapedhtml = r'''<span class="Data" id="BasicInfo1_lblAddress">618 \
AIRBRAKE  AVE<br/>WILMERDING,     PA 15148-23423</span>'''

soup = BeautifulSoup(scrapedhtml, 'lxml')
magiclist = soup.span.contents
exp=re.compile(',\s*(\w\w)')
m = re.search(exp,magiclist[2])
print(m.group(1))
```

output: PA

You try!

Given the string: parid='0450L00350000000'

extract the string: 45-L35