Regular Expressions

Computational Linguistics
Emory University
Jinho D. Choi
Regular Expressions

Special Characters

.  any character except a newline

^  the start of the string

$  the end of the string

*  ?  match 0 or more repetitions

+  ?  match 1 or more repetitions

?  ?  match 0 or 1 repetitions

{m}  ?  match exactly m repetitions

{m,n}?  match from m to n repetitions

|  or

minimal match
# Regular Expressions

## Special Sequences

| \num | match the contents of the group of the same number |
| \d   | match any decimal digit                           |
| \D   | match any non-digit character                    |
| \s   | match any whitespace character                    |
| \S   | match any non-whitespace character                |
| \w   | match any alphanumeric character and the underscore |
| \W   | match any non-alphanumeric character              |

## Functions

- compile, match, search, findall, finditer

[https://docs.python.org/2/library/re.html](https://docs.python.org/2/library/re.html)
String Formatting

Go to http://mathcs.emory.edu

Click “Undergraduate study” → “CS Courses”

year = 2015
term = 2
graduate = 0
url = 'http://www.mathcs.emory.edu/classes-semester.php?
subject=CS&year=%d&term=%d&graduate=%d' % (year, term, graduate)

String Operators

%d signed integer decimal
%f floating point decimal format
%s string

https://docs.python.org/2/library/stdtypes.html#string-formatting
HTML Request & Response

```python
import urllib2

request = urllib2.Request(url)
response = urllib2.urlopen(request)
page = response.read()  # html stream
kb = dict()

dictionary to store course information

for line in response:
    print line

https://docs.python.org/2/howto/urllib2.html
```
import re

RE_CLASSES_MAIN = re.compile('<div class="classes-main-toggle-buttons">(.+)</div>', re.DOTALL)

any sequence whose length > 0

m = RE_CLASSES_MAIN.search(page) vs. match?
main = m.group(1) vs. group(0)?

Find Iterator

list of tuples

RE_CLASS_TITLE = re.compile('<table .*?class="class-title">(.+?)</table>')

titles = [(m.group(1), m.start(), m.end()) for m in RE_CLASS_TITLE.finditer(main)]

create a tuple

find iteratively
Split & Join

\[
\text{RE_CLASS_NAME} = \text{re.compile('\langle td class="class-name">([^\s]+)\langle/td\rangle', re.DOTALL)}
\]

\[
\text{def splitTitle(title):}
\]
\[
\text{name = RE_CLASS_NAME.search(title).group(1)}
\]
\[
\text{t = name.split('::')}
\]
\[
\text{cnum = ' '.join(t[0].split())}
\]
\[
\text{cdes = t[1].strip()}
\]
\[
\text{return (cnum, cdes)}
\]

return multiple values
Grouping

```python
RE_CLASS_INFO = re.compile('(<td class="class-number">(.*?)(?:</td><td class="class-location">(.*?)(?:</td><td class="class-schedule">(.*?)(?:</td>)', re.DOTALL)

for i, title in enumerate(titles):
    (course_number, course_title) = splitTitle(title[0])
    start = title[2]
    if i+1 < len(titles): end = titles[i+1][1]
    else: end = -1

    for m1 in RE_CLASS_INFO.finditer(main, start, end):
        section = m1.group(1).strip()
        location = m1.group(2).strip()
        schedule = m1.group(3).strip()

        k = (name+'-'+section).upper()
        kb[k] = (location, schedule)
```
import sys

course = sys.argv[1]  # First parameter
print kb[course]

$ python parse_html.py CS329-001