Regular expressions

@jessicamckellar
Regular expression

A sequence of characters used to find patterns in text
Regular expression
A sequence of characters used to find patterns in text

Great for

• Finding things
• Replacing things
• Cheating at crosswords
• Lots more!
Our framework for tonight

# The official SOWPODS Scrabble
# dictionary; 267751 words.

import scrabble

for word in scrabble.wordlist:
    # Print the words we care about
Warmup

What are all of the words that contain “UU”?
import scrabble

for word in scrabble.wordlist:
    if "uu" in word:
        print word
import scrabble
import re

pattern = re.compile("uu")

for word in scrabble.wordlist:
    if pattern.search(word):
        print(word)
Boundaries

What are all of the words that start with “AA”?
| ^ | the beginning |
Boundaries

What are all of the words that end with “OO”?
## Special characters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>^</code></td>
<td>the beginning</td>
</tr>
<tr>
<td><code>$</code></td>
<td>the end</td>
</tr>
</tbody>
</table>
Wildcards

31 across:

a break or pause (usually for sense) in the middle of a verse
Wildcards

• What are all of the valid 2-letter Scrabble words?

• Are there any words that start with “A” and end with “Z”?
# Special characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>^</code></td>
<td>the beginning</td>
</tr>
<tr>
<td><code>$</code></td>
<td>the end</td>
</tr>
<tr>
<td><code>.</code></td>
<td>a single character</td>
</tr>
<tr>
<td><code>.*</code></td>
<td>zero to many characters</td>
</tr>
</tbody>
</table>
Character classes

• What are all of the words that contain only vowels and Y?

• What are all of the words that contain NO vowels or Y?
Character classes

What is the longest word that can be typed with only your left hand on a QWERTY keyboard?
Character classes

What is the longest word that can be typed with only your left hand on a QWERTY keyboard?

```python
pattern = re.compile("^[qwertasdfgzxcvb]*$")
longest = ""
for word in scrabble.wordlist:
    if pattern.search(word) and \
        len(word) > len(longest):
        longest = word
print longest
```
sweaterdresses
Character classes

What is the longest word that can be typed with only your right hand on a QWERTY keyboard?

hypolymnion

“the dense, bottom layer of water in a thermally-stratified lake”
Character classes

ranges

• $[a-c]$
• $[a-zA-Z]$
• $[0-9]$

shorthands

• $\texttt{\d} = \text{digit} = [0-9]$
• $\texttt{\s} = \text{whitespace} = [\texttt{\t}\texttt{\n}\texttt{\r}\texttt{\f}\texttt{\v}]$
• $\texttt{\w} = \text{alphanumeric} = [a-zA-Z0-9_]$
Character classes

ranges

- \([a\text{-}c]\)
- \([a\text{-}zA\text{-}Z]\)
- \([0\text{-}9]\)

shorthands

- \(\d = \text{digit} = [0\text{-}9]\)
- \(\s = \text{whitespace} = [\ \t\n\r\f\v]\)
- \(\w = \text{alphanumeric} = [a\text{-}zA\text{-}Z0\text{-}9_]\)

Tip: the upper-case shorthand is the reverse of the lowercase, e.g.

\(\W = \text{non-alphanumeric} = [^{a\text{-}zA\text{-}Z0\text{-}9}_]\)
## Special characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>^</code></td>
<td>the beginning</td>
</tr>
<tr>
<td><code>$</code></td>
<td>the end</td>
</tr>
<tr>
<td><code>.</code></td>
<td>a single character</td>
</tr>
<tr>
<td><code>.*</code></td>
<td>zero to many characters</td>
</tr>
<tr>
<td><code>[]</code></td>
<td>character classes</td>
</tr>
<tr>
<td><code>[^]</code></td>
<td>negation</td>
</tr>
</tbody>
</table>
Repetition

\{m\}: repeat m times

Matching phone numbers:

617-[\d][\d][\d][\d]

or

617-[0-9]\{4\}
Repetition

\{m,n\}: repeat m through n times

Matching usernames:

[ \w\] \{3,18\}

+: repeat at least once

Matching URL slugs:

[ a-z0-9-\]+
<table>
<thead>
<tr>
<th>Special characters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{m}</code></td>
<td>repeat m times</td>
</tr>
<tr>
<td><code>{m, n}</code></td>
<td>repeat m through n times</td>
</tr>
<tr>
<td><code>+</code></td>
<td>repeat one or more times</td>
</tr>
<tr>
<td><code>'</code></td>
<td>repeat zero or one times</td>
</tr>
</tbody>
</table>
References

What actually matched my regex?
References

```python
>>> text = """Jessica 617-123-4567
...  Adam 617-987-6543
...  Olivia 617-222-2222"""

>>> pattern = re.compile('(\d{3}-\d{3}-\d{4}))
```
References

```python
>>> text = """Jessica 617-123-4567
... Adam 617-987-6543
... Olivia 617-222-2222"

>>> pattern = re.compile('(\d{3}-\d{3}-\d{4})')

>>> pattern.match(text)
```
References

>>> text = """Jessica 617-123-4567
... Adam 617-987-6543
... Olivia 617-222-2222"

>>> pattern = re.compile('(^\d{3}-\d{3}-\d{4})')

>>> pattern.match(text)
<_sre.SRE_Match object at 0x10051a558>

>>> pattern.search(text)
<_sre.SRE_Match object at 0x10051a558>
References

```python
>>> text = """Jessica 617-123-4567
... Adam 617-987-6543
... Olivia 617-222-2222""
>>> pattern = re.compile('(\d{3}-\d{3}-\d{4})')
>>> pattern.match(text)
<_sre.SRE_Match object at 0x10051a558>
>>> pattern.search(text).group()
'617-123-4567'
```
References

```python
>>> text = """Jessica 617-123-4567
... Adam 617-987-6543
... Olivia 617-222-2222""
>>> pattern = re.compile('(^\d{3}-\d{3}-\d{4})')
>>> pattern.match(text)
>>> pattern.search(text)
<_sre.SRE_Match object at 0x10051a558>
>>> pattern.search(text).group()
'617-123-4567'
>>> pattern.findall(text)
['617-123-4567', '617-987-6543', '617-222-2222']
```
>>> regex = "(?P<username>\[\w\]{6,18})/(?P<slug>[a-zA-Z-]+)"
>>> p = re.compile(regex)
>>> res = p.search("http://blog.com/jesstess/python-notes")
References
Named references!

```python
>>> regex = "(?P<username>\w{6,18})/(?P<slug>[a-zA-Z-]+)"
>>> p = re.compile(regex)
>>> res = p.search("http://blog.com/jessstess/python-notes")
```

Give this match the name “username”
An alphanumeric string between 6 and 18 characters

Give this match the name “slug”
At least 1 letter or hyphen
References
Named references!

```python
>>> regex = "(?P<username>\[\w\]{6,18})/(?P<slug>[a-zA-Z-]+)"
>>> p = re.compile(regex)
>>> res = p.search("http://blog.com/jesstess/python-notes")
>>> res.groups()
('jesstess', 'python-notes')
>>> res.group("username")
'jesstess'
>>> res.group("slug")
'python-notes'
```
## Special characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{m}</code></td>
<td>repeat m times</td>
</tr>
<tr>
<td><code>{m, n}</code></td>
<td>repeat m through n times</td>
</tr>
<tr>
<td><code>+</code></td>
<td>repeat one or more times</td>
</tr>
<tr>
<td><code>?</code></td>
<td>repeat zero or one times</td>
</tr>
<tr>
<td><code>()</code></td>
<td>a reference group</td>
</tr>
<tr>
<td><code>(?P&lt;name&gt;)</code></td>
<td>named reference</td>
</tr>
</tbody>
</table>
Backreferences

Matching what we’ve matched before
Backreferences
Matching what we’ve matched before

```python
>>> text = "The lazy brown dog went zzzzzzz."
>>> pattern = re.compile(r"(.)(.)\1\{5,\}")
>>> pattern.search(text).group()
'zzzzzz'
```
Backreferences
Matching what we’ve matched before

```python
>>> text = "The lazy brown dog went zzzzzzz."
>>> pattern = re.compile(r"(.\1){5,}" )
>>> pattern.search(text).group()
'zzzzzz'
```

We need a special string type, `r`, to correctly interpret the backslash.
Backreferences

Do any words have the same letter 7 times?

Mississi...ss...i...ss..ippi
Review

- Boundaries
- Wildcards
- Character classes
- Repetition
- References
- Backreferences
RegExes in the wild

**python-markdown:**

```python
header_rgx = re.compile("[Hh][123456]"")
```

**Jinja:**

```python
number_re = re.compile("-\d+(\.\d+)"")
```

**SQLAlchemy:**

```python
DATE_RE = re.compile("(\d+)-(\d+)-(\d+)"")
```
Visualizers

\(\text{(d\{3\}-d\{3\}-d\{4\})}\)

regrexper.com
What next?

• Level up your regex-friendly command line utilities

• Code audit: what would regexes make clearer or more robust?

• Cheat at Words with Friends
Thanks! Questions?

Regular expressions

@jessicamckellar