Regular Expressions

CS 339R (Python) — Chapter 16

Spring 2011
Regular Expressions

✦ The `re` Module

✦ A powerful pattern-matching capability

✦ Patterns are often written as *raw strings*:
  
  ✦ to avoid hassling with escape characters:
  
  ✦ `>>> print '\n'` # len == 1

  ✦ `>>> print r'\n'` # len == 2
**re Functions**

- `search(pattern, string)`  
  # Looks for match *anywhere* in string

- `match(pattern, string)`  
  # Looks for match at *start* of string

- `findall(pattern, string)`  
  # Extracts *strings* of interest as a *list*

  ```python
  words = (w for w in re.findall(r'\[a-zA-Z\']+', text))
  ```

- `finditer(pattern, string)`  
  # Like `findall`, but returns an *iterator* to *MatchObjects*

- `split(pattern, string)`  
  # Splits on matches for string

- `sub(pattern, repl, string)`  
  # Replaces matches with *repl*
**Special Character Sequences**

- .  
  any char but ‘\n’

- ^  
  start of string

- $  
  end of string

- *  
  0 or more reps

- +  
  1 or more reps

- ?  
  0 or 1 rep

- *?  
  0 or more (min)

- +?  
  1 or more (min)

- {m}  
  m reps

- {m,n}  
  m-to-n reps (max)

- [...]  
  match from set

- [^...]  
  match from complement

- A | B  
  match A or B

See re1.py, re2.py
Idioms

- .* Matches anything, including the empty string
- .+ Matches any non-empty string
- See resample1.py, resample2.py
Special Characters

- \b  match a word boundary
- \d  match a decimal digit (same as [0-9])
- \D  match non-digit (same as [^0-9])
- \s  match any whitespace character
- \S  match any nonwhite character
- \w  match alphanumeric and underscores
- \W  match non-alphanumeric