Doing things they don’t support.

- The overview:
  - Tools for Regular Expressions (Regex)
  - Mechanize Libraries
  - Beautiful Soup Scraping Libraries

First Up – Regular Expressions

- We know what these are, but forget subtleties.
- We need tools:
  - Regexr program :: tinyurl.com/regexr
    - Desktop version available for Adobe Air
  - *Mastering Regular Expressions*, O’Reilly Press :: tinyurl.com/masteringregex

Next Up – Mechanize == The Shiz

- What it is:
  - A way to script browsing the web
- How it rocks:
  - Cookies, session management, form completion, browsing
- Where it fails:
  - Overly AJAXy stuff
  - ASP pages, it seems are a pain

The Mechanize Workflow

- Workflow:
  1) Examine the form we’re working on:
     - Web Developer Toolbar :: tinyurl.com/web-devel
     - Live HTTP Headers :: tinyurl.com/livehttp
     - Mobile sites :: m.google, m.facebook, m.yelp, etc.
  2) Write the script
  3) Test & use the script
- Link for Python: tinyurl.com/mech-py
- Link for Perl: tinyurl.com/mech-perl

For Example (7 lines)

```
from mechanize import Browser

# Create a browser object & give it a URL
br = Browser()
br.open("http://twitter.com")

# Select the form you want, in this case, number 1
br.select_form(nr=1)

# Set the variables
br["session[username_or_email]"] = "snahblah"
br["session[password]"] = "snahblah"

# Submit the form!
response2 = br.submit()
```

Finally, Beautiful Soup

- Beautiful Soup :: tinyurl.com/beaut-soup
- It lets you handle HTML in better ways.
- Helpful for scripts that scrape to get what you want from the HTML ”Soup”
- Sorry, I’m no expert, but check out the documentation – It’s good.