May 9

Today’s topics:
continue with regular expressions (topic for lab 5)
examples
discussion -- html, cgi, regular expressions

Review

From Wednesday:
- A regular expression is a pattern
- Letters, digits, etc match themselves
- Some symbols (“operators”) match sets of characters or have other special meanings

<table>
<thead>
<tr>
<th>symbol</th>
<th>meaning</th>
<th>symbol</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>any character</td>
<td>^</td>
<td>beginning of line</td>
</tr>
<tr>
<td>[xyz]</td>
<td>any one of x, y, or z</td>
<td>$</td>
<td>end of line</td>
</tr>
<tr>
<td>\d</td>
<td>any decimal digit</td>
<td>\</td>
<td>quote character (ignore any special meaning of the next character)</td>
</tr>
<tr>
<td>\w</td>
<td>any “word character”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Review

grep and egrep are Unix commands that search files for lines containing patterns

```bash
[fintan:170] % egrep '[12345]' labs/docs/test.txt
line 1
line 2
[fintan:170] % egrep 'text' labs/docs/test.txt
text with special chars: * - \
[fintan:170] % egrep '\.$' labs/docs/test.txt
a sentence ends with a period.
some sentences, like this one, have commas.
```

Regular expressions in Ruby are strings surrounded by slashes

```ruby
>> s = "peter piper picked a peck of peppers"
>> s.index("pi")
=> 6
>> s.index(/p[aeiou]/)
=> 0
>> $["pi"]
=> "pi"
>> $[/pi/]
=> "pi"
>> $[/p[aeiou]/]
=> "pe"
>> s.scan(/p[aeiou]/)
=> ["pe", "pi", "pe", "pi", "pe", "pe", "pe"]
```
Hacked Election

- See:
  - election.rhtml (generic election outline)
  - election.final.rhtml (official results, with names filled in from database)
  - hacked.rhtml (Marvin's modified version of election.final.rhtml)

- To see what is sent back to user:
  - % erb election.final.rhtml
  - % erb hacked.rhtml