

Perl Lesson 7

L615

Spring 2009

1. RE Interpolation: we can use variables inside regular expressions

```
$name = "tom";

# reading through a piped-in file
while (<>) {
    if (/$name/) {
        print $_;
    }
}
```

2. Automatic Match Variables: gives us a way to refer to the actually matched item vs. the surrounding context

- `$&` = matched string
- `$'` = material preceding matched string
- `$'` = material following matched string

```
if ("larry, moe, and curly" =~ /\W+(moe)\W+/) {
    print "$'\n";
    print "$&\n"; # NB: not the same as $1
    print "$'\n";
}
```

3. Substitutions — `s///` — REs allow us to specify a pattern and then replace it

- Global replacements — `/g` — by default, `s///` replaces only the first match; `/g` allows you to replace all matches
- Delimiters — can change from `/` to something else (if, e.g., you want to refer to a string that uses `'`)
- Options (Case Shifting) — as with regular RE matching, we can use the `/i`, `/x`, and `/s` modifiers

```
$_ = "larry, moe, and curly";
s/curly/shemp/;
print "$_\n"; # prints "larry, moe, and shemp"

$_ = "larry, moe, and curly";
s/(\w+)\W+(\w+)/$2 $1/; # swaps words
print "$_\n";

$_ = "larry, moe, and curly had curly hair";
s/curly/shemp/g;
print "$_\n";

$_ = "Larry, Moe, and Curly";
s/(\w+)/\U$1/; # makes first word uppercase
print "$_\n";
s/(\w+)/\L$1/g; # makes all words lowercase
print "$_\n";
```