Recipes for PDSC #2: Hashes of Lists

by Tom Christiansen
< tchrist@perl.com >

release 0.1 (untested, may have typos)
Sunday, 1 October 1995

Declaration of a HASH OF LISTS:

```
%HoL = (
       "flintstones"        => [ "fred", "barney" ],
       "jetsons"            => [ "george", "jane", "elroy" ],
       "simpsons"           => [ "homer", "marge", "bart" ],
     );
```

Generation of a HASH OF LISTS:

```
# reading from file
# flintstones: fred barney wilma dino
while ( <> ) {
    next unless s/^(.*?):\s*//;
    $HoL{$1} = [ split ];
}

# reading from file; more temps
# flintstones: fred barney wilma dino
while ( $line = <> ) {
    ($who, $rest) = split /:\s*/, $line, 2;
    @fields = split ' ', $rest;
    $HoL{$who} = [ @fields ];
}

# calling a function that returns a list
for $group ( "simpsons", "jetsons", "flintstones" ) {
    $HoL{$group} = [ get_family($group) ];
}

# likewise, but using temps
for $group ( "simpsons", "jetsons", "flintstones" ) {
    @members = get_family($group);
    $HoL{$group} = [ @members ];
}

# append new members to an existing family
push @{ $HoL{"flintstones"} }, "wilma", "betty";
```

Access and Printing of a HASH OF LISTS:

```
# one element
$HoL{flintstones}[0] = "Fred";
```

```
# another element
$HoL{simpsons}[1] =~ s/(\w)/\u$1/;

# print the whole thing
foreach $family ( keys %HoL ) {
    print "$family: @{ $HoL{$family} }\n"
}

# print the whole thing with indices
foreach $family ( keys %HoL ) {
    print "family: ";
    foreach $i ( 0 .. $#{ $HoL{$family} } ) {
        print " $i = $HoL{$family}[$i]";
    }
    print "\n";
}

# print the whole thing sorted by number of members
foreach $family ( sort { @{$HoL{$b}} <=> @{$HoL{$b}} } keys %HoL ) {
    print "$family: @{ $HoL{$family} }\n"
}
# print the whole thing sorted by number of members and name
foreach $family ( sort { @{$HoL{$b}} <=> @{$HoL{$a}} } keys %HoL ) {
    print "$family: ", join(", ", sort @{ $HoL{$family}}), "\n";
}
```