DON'T FEAR MAP & GREP

LIST PROCESSING FOR FUN AND PROFIT

WHY USE map AND grep?

Simplifies your code when processing lists and arrays

Replace or augment `for` and `foreach` loops to better describe what's going on

Stepping stone to other list processing functions from CPAN

From this:

```perl
my @greetings;
foreach my $name (@people) {
    push @greetings, "Hello $name";
}
```

To this:

```perl
my @greetings = map {"Hello $_"} @people;
```

MARK GARDNER 22 JUNE 2022

map IS THE LIST TRANSFORMER

Two inputs: an expression or block of code, and a list to run it on

Expression style:
```
my @foo = map bar($_), @list;
```

Block style:
```
my @foo = map { bar($_) } @list;
```

Block is better—easier to to read, more flexible when building complicated lists

Another example, building a hash:

```
my %IS_EXIT_WORD = map { $_ => 1 }
  qw(q quit bye exit stop done last finish);

...

die if $IS_EXIT_WORD{$command};
```

MARK GARDNER 22 JUNE 2022

grep IS THE LIST FILTER

Inspired by Unix grep command, but not just for regular expressions

Simple example:

```perl
my @months_with_a = grep { /[Aa]/ } qw(
  January  February March
  April    May      June
  July     August   September
  October  November December
);
```

Running functions that return true/false:

```perl
my @results =
  grep { large_database_query($_) } @foo;
```

Check for defined-ness:

```perl
say $_ > 5 ? "$_ is bigger"
           : "$_ is equal or smaller"
  for grep { defined } @numbers;
```

MARK GARDNER 22 JUNE 2022

BEWARE SIDE EFFECTS

THIS MACHINE KILLS ARRAY ELEMENTS

What does this do?

```
my @needs_docs =
  grep { s/\.pm$/.pod/ && !-e } @pm_files;
```

Should have used a variable or refactored:

```
my @needs_docs = grep { no_docs($_) }
                    @pm_files;
sub no_docs {
    my $file = shift;
    $file =~ s/\.pm$/.pod/;
    return !-e $file;
}
```

Or non-destructive substitution:

```
use v5.14;
my @needs_docs = grep { !-e s/\.pm$/.pod/r }
  @pm_files;
```

foreach if you really want side effects

MARK GARDNER 22 JUNE 2022

MORE ON LISTS:

phoenixtrap.com/tag/lists