LOOPS!

Brandon Seah

Perl Course 2017

ITERATION ≠ RECURSION

- Iteration - Repeating an action
- Recursion - Including a function inside its own definition

RECURSION EXAMPLE

```perl
sub factorial_recurse {
    my ($N) = @_;
    if ( $N == 0 ) {
        return 1;
    } else {
        return $N * factorial_recurse($N-1);
    }
}
```

Can rapidly eat up a lot of memory!

HOW TO ITERATE?

- Use a counter
- Over an array

HOW DO I LOVE THEE?

> *How do I love thee? Let me count the ways.*
> *I love thee to the depth and breadth and height*
> *My soul can reach, when feeling out of sight*
> *For the ends of being and ideal grace. ...*
>
> *Elizabeth Barrett Browning*

HOW DO I LOVE THEE?

```
my @ways = ("to the depth ... my soul can reach",
           "to the level of every day's/ Most quiet need",
           "freely",
           "purely",
           "with the passion put to use/ In my old griefs",
           "with my childhood's faith, # et cetera
           );
```

File: `loops_template.pl`

Pop quiz: How do I count the ways?

HOW DO I LOVE THEE?

USING `WHILE` LOOP

```
my @ways = ...; # From above
my $i = 0; # Initialize counter
while ($i < scalar(@ways)) {
   print "I love thee ". $ways[$i]."\n";
   $i++;
}
```

Over the **index** of an array

THREE THINGS A COUNTER NEEDS

- Initialize (Where to start?)
- Stop condition (When to stop?)
- Increment (How much per step?)

HOW DO I LOVE THEE?

USING `FOR` LOOP

```
my @ways = ...; # From above
for (my $i=0; $i< scalar(@ways); $i++) {
    print "I love thee ". $ways[$i]."\n";
}
```

- All three things are in one line
- Does `$i` exist outside the loop?

HOW DO I LOVE THEE?

USING `FOREACH` LOOP

```
my @ways = ... ; # From above
foreach my $current_way (@ways) {
   print "I love thee ". $currentWay."\n";
}
```

- No counters or indices! (Implicit)
- Does `$current_way` exist outside the loop?

`FOR` VS `FOREACH`

No difference!

But `foreach` sounds like English

ALL TOGETHER

```
my @ways = ...; # From above
my $i = 0; # Initialize counter
while ($i < scalar(@ways)) {
   print "I love thee ". $ways[$i]."\n";
   $i++;
}

for (my $j=0; $j< scalar(@ways); $j++) {
    print "I love thee ". $ways[$j]."\n";
}

foreach my $current_way (@ways) {
   print "I love thee ". $currentWay."\n";
}
```

REVIEW

- Iteration with and without counter
- *Scope* of variable in loop
- `foreach` is easier for arrays

EXERCISES

First make array of integers from 1 to 10:

```
my @integers = 1..10;
```

Use `while`, `for`, and `foreach` to do the following:

- Return the square of each integer
- Return the running sum
- Return the running product (i.e. factorial series)