CSE 114A - lecture 7!

today:

- Tail recursion
- case expressions
- if time: come back to our simple AST & interpreter and try to do more.

```
fact :: Int -> Int
fact n | n <= 1 = 1
fact n | otherwise = n * fact (n - 1)
```

```
> fact 5
5 * fact 4
5 * (4 * fact 3)
        :
5 * (4 * (3 * (2 * fact 1)))
5 * 4 * 3 * 2 * 1
120
```

lots of "work" to do still!
Not tail-recursive.

"A recursive function is tail-recursive if the recursive call is the final result of the function itself."

↑ Quoting my friend Brent Yorgey

If the result of the function call needs some further processing (e.g. multiplying by another number, consing something onto it, etc.), it's not tail-recursive.

A tail-recursive version of factorial would behave like this:

```
> factTR 5 1
factTR 4 5
factTR 3 20
factTR 2 60
factTR 1 120
120
```

Tail-recursive because there's no more work to do after the recursive call returns.