```
m * n = m/2 * 2n

  8 * 12
= 4 * 24
= 2 * 48
= 1 * 96

.
```

```
m * n = m//2 * 2n
             ...  w/ leftover

  9 *  12  ---> 12
= 4 *  24
= 2 *  48
= 1 *  96  ---> 96
= 0 * 192
.
```

Sequencing only matters when expressions have side effects.

Side effects only matters when expressions are in sequence.

.

Until today, every data value we have used in the course has been

**immutable**

Even when we created a local "variable", we assigned a value to it exactly once.

.

# imperative programming

.

**Defining a name**

and

**changing the value of a named object**

are different activities.

They should be different operations in the language.

.

In C++:

```
Foo a = new Foo();
```

versus:

```
Foo a;
a = new Foo();
```

.

# set!

# closure

a **function**

plus

the **environment** that existed
when the function was created

# state

name — value — location