Code

Here are a couple of types.

```haskell
data Couldbe a = Sure a
  | Nope

data Weird a = Foo a
  | Bar a a
  | Baz
```

None of this “deriving Show” stuff! We’re going to do this manually.

Questions

Do these questions first for Couldbe, and then for Weird.

1. Use the Show typeclass to implement a show function.

2. Use the Eq typeclass to implement equality.

3. Use the Functor typeclass to implement `fmap`.

4. Use the Applicative typeclass to implement `pure` and `<*>`. 