

Learn you a Hask/Elm
for Great JavaScript

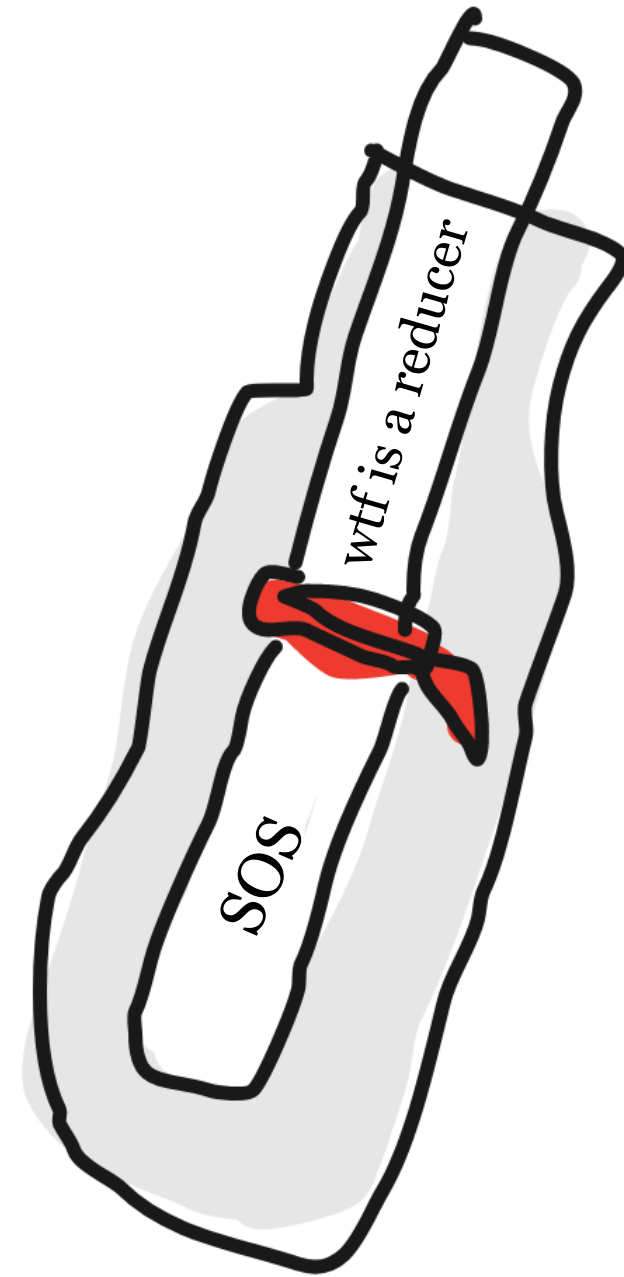
Message and a Model

```
-- MODEL
type alias Model = { ... }

-- UPDATE
type Msg = Reset | ...

update : Msg → Model → Model
update msg model =
  case msg of
    Reset → ...
    ...

-- VIEW
view : Model → Html Msg
view model =
  ...
```



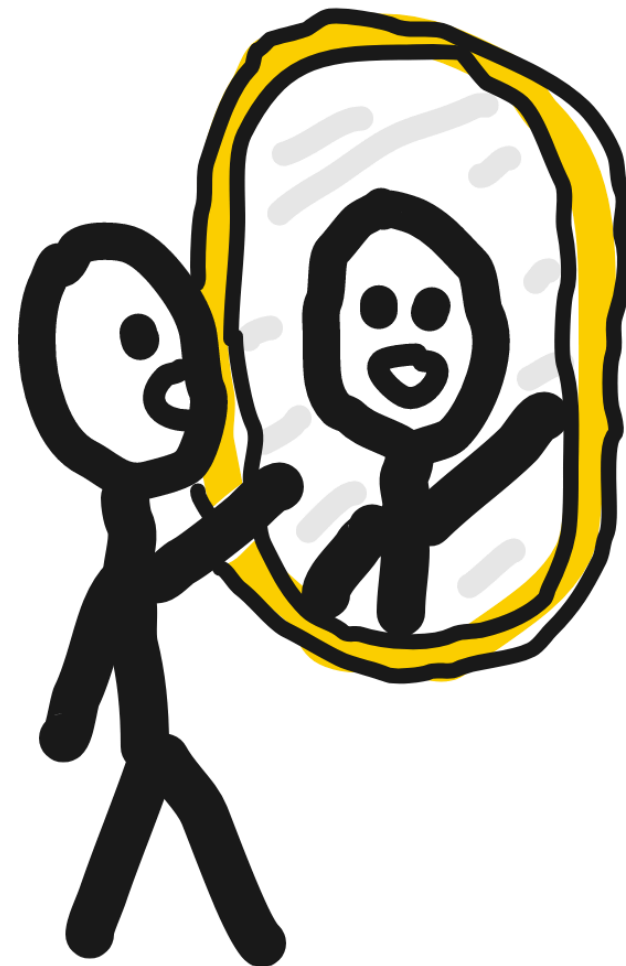
I'm looking at the man in the mirror

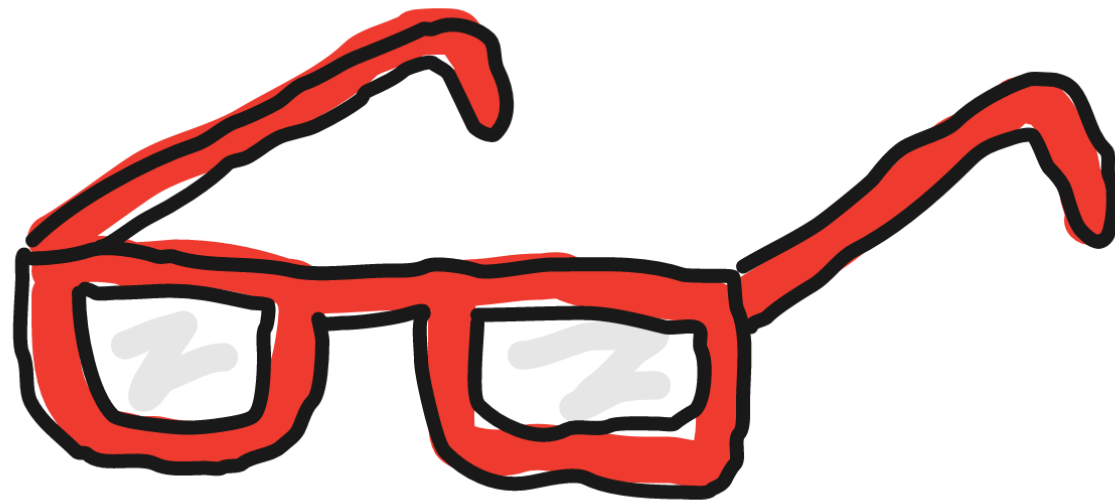
Haskell

```
elem' :: (Eq a) => a -> [a] -> Bool
elem' y ys = foldl (\acc x -> if x == y then True else acc) False ys
```

Pseudo JavaScript

```
elem(y: Eq, ys: Eq[]) : Bool {
  return _.reduce(
    ys,
    (acc, x) => {
      if (x == y) return true;
      return acc
    },
    false
  );
}
```





readability



side effects



type systems



thinking functionally





everything is connected

surprise: you learned something