

# Why Dependent Types?

```
map : forall {a : Type, b : Type}  
      . (a -> b) []
```

```
-> List a
```

```
-> List b
```

```
map [f] Empty = Empty;
```

```
map [f] (Cons x xs) = Cons (f x) (map [f] xs)
```

# Why Dependent Types?

```
map : forall {a : Type, b : Type, n : Nat}
      . (a -> b) [n]
    -> Vec n a
    -> Vec n b
map [f] Empty = Empty;
map [f] (Cons x xs) = Cons (f x) (map [f] xs)
```