SAN DIEGO FUNCTIONAL PROGRAMMERS

Schedule

- 10 min - Intro & Welcome
- 30 min - Point-Free Style
- 60 min - Kata time
- 20 min - Demo

Point-Free

What Is Point-Free

Point-Free Code Is Meant To Aide Communication

For Example

```
sum (x:xs) = x + sum1 xs
sum []     = 0
sum xs = foldr (+) 0 xs
```

```haskell
sum = foldr (+) 0
```

Eta Reduction

For Example

```
sumPlus xs = sum (map (+1) xs)
sumPlus xs = sum $ map (+1) xs
sumPlus xs = sum . map (+1) $ xs
sumPlus = sum . map (+1)
```

Eta Abstraction

Currying

For Example

```haskell
add :: Num a => a -> a -> a
add :: Num a => a -> (a -> a)
```

```
> :t add 10
(add 10) :: Num a ⇒ a → a
```

```
> :t add 10 10
(add 10 10) :: Num a ⇒ a
```

Summary

Point-Free Is A Coding Style
That Doesn’t Use Arguments
When Defining A Function

Pointful Code Can Be Transformed Into Point-Free Using Eta Reduction

Point-Free Code Can Be Made More Pointful Using Eta Reduction

Most Importantly

Point-Free Code Is Meant To Aide Communication

References

https://en.wikipedia.org/wiki/Tacit_programming

https://www.youtube.com/watch?v=seVSlKazsNk

The End

https://bit.ly/sdfp-june-2019

Some Guidelines

- Try to work in groups!
- Prefer recursion to loops
- No mutation
- If all else fails, forget the guidelines