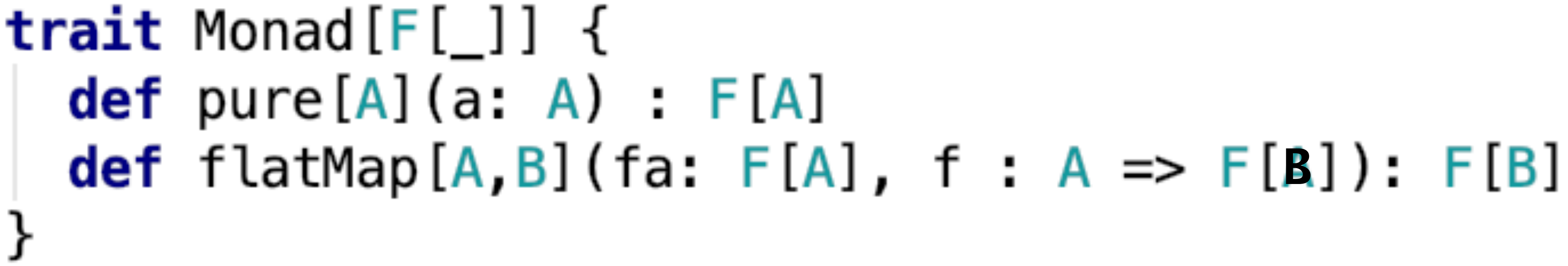
Monads in Scala



```
trait Monad[F[_]] {
  def pure[A](a: A) : F[A]
  def flatMap[A,B](fa: F[A], f : A => F[B]): F[B]
}
```

Implemented as a type class

Achieves the same as in Haskell

pure = put an A in a Monad

flatMap = sequence two monads

IO Monad - Future

Futures allow asynchronous effects

They have pure and bind (Future.success and Future.flatMap)

Future is not a Monad

Future execution is eager

http://justinhj.github.io/2018/05/05/hacker-news-api-4.html