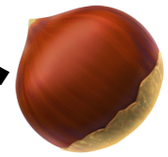
举个 

```
const canHalfNTimes = (num, n) => R.pipe(
  R.times(R.identity),
  R.reduce(
    (res) => res.chain(a => a % 2 === 1 ? Nothing() : Just(a / 2)),
    Just(num)
  )
)(n);
console.log(canHalfNTimes(663552, 5)); // Just {value: 20736}
console.log(canHalfNTimes(1988280, 5)); // Nothing {}
```

Functor
（函子）

应用一个函数到包裹的值

```
class Functor f where
  fmap:: (a -> b) -> f a -> f b
```

Applicative
(增强函子)

应用一个包裹的函数到包裹的值

```
class Functor f => Applicative f where
  pure:: a -> f a
  (<*>) :: f(a -> b) -> f a -> f b
```

Monad
(单子)

应用一个接收值返回包裹的值的函数到包裹的值

```
class Applicative m => Monad m where
  return :: a -> m a
  (>>=) :: m a -> (a -> m b) -> m b
```