

Apply



Weaker Applicatives

= Functor + Semigroupal + ap method

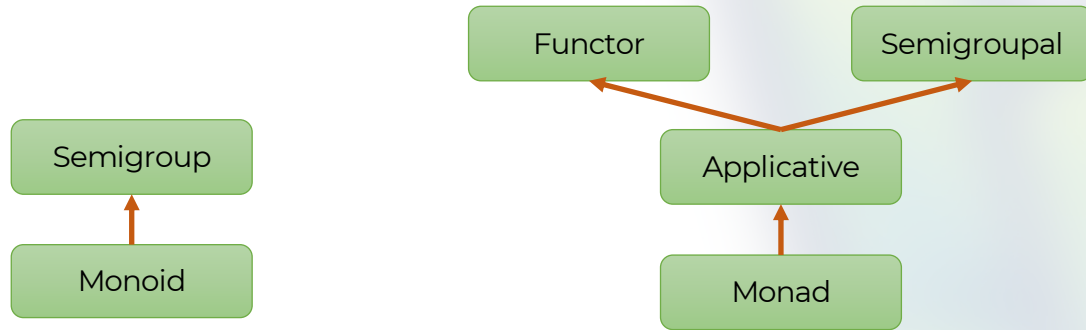
```
def ap[A, B](wf: W[A => B])(wa: W[A]): W[B]
```

```
import cats.Apply
import cats.instances.option._ // implicit Apply[Option]
val applyOption = Apply[Option]
val funcApp = applyOption.ap(Some((x: Int) => x + 1))(Some(2)) // Some(3)
```

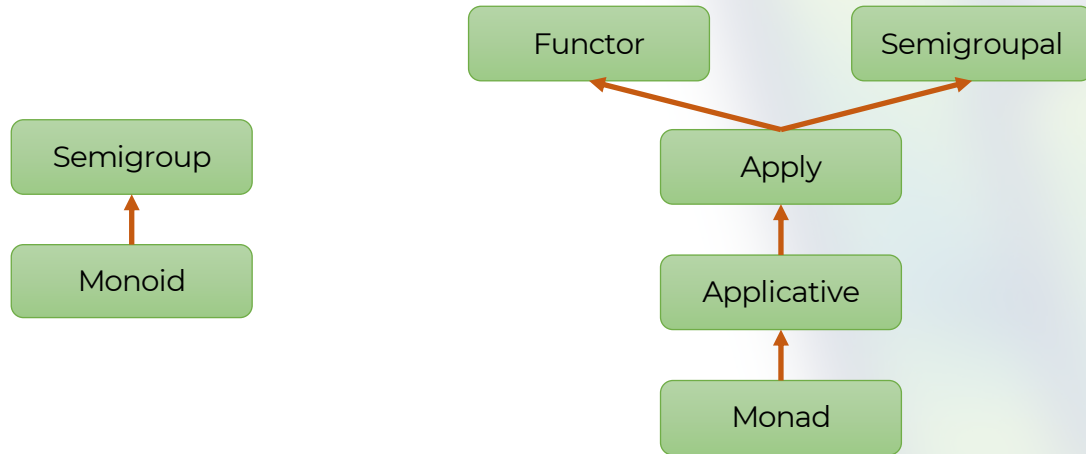
Convenient for extracting and combining tuples

```
import cats.syntax.apply._ // extension methods from Apply
val tupleOfOptions = (Option(1), Option(2), Option(3))
val optionOfTuple = tupleOfOptions.tupled // Some((1,2,3))
val sumOption = tupleOfOptions.mapN(_ + _ + _) // Some(6)
```

Cats TC Hierarchy



Cats TC Hierarchy



Cats rock

