Monoids

Monoids

Natural extension of Semigroups that can offer a "zero" value

```
import cats.Monoid
import cats.instances.int._

val naturalIntMonoid = Monoid[Int] // fetches the implicit instance
val intCombination = naturalIntMonoid.combine(2, 45) // same combine from Semigroup
val zero = naturalIntMonoid.empty // fundamental to Monoids
```

```
import cats.syntax.monoid._ // includes everything in the Semigroup syntax
val anIntCombination = 2 |+| 3 // dictated by the Monoid[Int]
```

Useful for general APIs

```
def combineFold[T](list: List[T])(implicit monoid: Monoid[T]): T = list.fold(monoid.empty)(_ |+| _)
```

Use cases: data structures meant to be combined, with a starting value

- data integration & big data processing
- eventual consistency & distributed computing

Cats rock