

# Type Classes



# Takeaways

Type class

```
trait MyTypeClassTemplate[T] {  
  def action(value: T): String  
}
```

Type class *instances* (often implicit)

```
implicit object MyTypeClassInstance extends MyTypeClassTemplate[Int] { ... }
```

Invoking type class instances

```
object MyTypeClassTemplate {  
  def apply[T](implicit instance: MyTypeClassTemplate[T]) = instance  
}
```

```
MyTypeClassTemplate[Int].action(2)
```

# Takeaways

Type class

```
trait MyTypeClassTemplate[T] {  
  def action(value: T): String  
}
```

Type class *instances* (often implicit)

```
implicit object MyTypeClassInstance extends MyTypeClassTemplate[Int] { ... }
```

Enriching types with type classes

```
implicit class ConversionClass[T](value: T) {  
  def action(implicit instance: MyTypeClassTemplate[T]) = instance.action(value)  
}
```

2.action

# Scala rocks

