

Partial Functions



Takeaways

```
trait PartialFunction[-A, +B] extends (A => B) {  
  def apply(x: A): B  
  def isDefinedAt(x: A): Boolean  
}
```

How to use:

```
val simplePF: PartialFunction[Int, Int] = {  
  case 1 => 42  
  case 2 => 65  
  case 3 => 999  
}
```

done with pattern matching!

Utilities:

- `isDefinedAt`
- `lift`
- `orElse`

Used on other types:

- *map, collect* on collections
- *recover*

Scala rocks

