



# ЧАСТИЧНАЯ ФУНКЦИЯ

# ЧАСТИЧНАЯ ФУНКЦИЯ

```
PartialFunction[-A, +B]
```

```
def isDefinedAt(input: A): Boolean
```

# ЧАСТИЧНАЯ ФУНКЦИЯ

```
val divide10: PartialFunction[Int, Int] = {  
  case 1 => 10  
  case 2 => 5  
  case 5 => 2  
  case 10 => 1  
}
```

```
divide10.isDefinedAt(2) == true  
divide10.isDefinedAt(3) == false
```

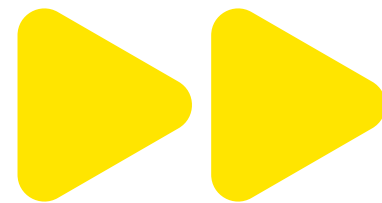
```
divide10(2) == 5  
divide10(3) == throw new MatchError !!!!!
```

# МЕТОД .COLLECT

```
List.range(1, 10) // List(1, 2, ..., 10)
```

```
List.range(1, 11).collect(divide10)  
//List(10, 5, 2, 1)
```

**Мы изучили  
частичные функции**



**В следующей части  
мы изучим полезные типы данных**