



**КОРТЕЖИ**

**Кортежи** — predetermined types, representing sequences of values of fixed length

# ОБЪЯВЛЕНИЕ КОРТЕЖА

```
val tuple = (value1, value2, ..., valueN)
```

```
type TupleType = (Type1, Type2, ..., TypeN)
```

```
def divMod(x: Int, y: Int): (Int, Int) =  
    (x / y, x % y)
```

```
def firstLastAndCount(line: String): (Char, Char, Int)  
    (line(0), line.last, line.length)
```

# РАСПАКОВКА КОРТЕЖА

```
val (val1 : Type1, val2: Type2, ..., valN) = tupl
```

```
val (div, mod) = divMod(17, 5)
```

```
val (first, last, count) = firstLastAndCount("Scala")
```

```
def showDiv(x: Int, y: Int) =  
  divMod(x, y) match {  
    case (d, r) => s"$x is $d * $y + $r"  
  }
```

# ОБРАЩЕНИЕ К ЭЛЕМЕНТУ

```
tuple._1  
tuple._2  
...  
tuple._N
```

```
val dm = divMod(17, 5)  
  
val div = dm._1  
  
val mod = dm._2
```

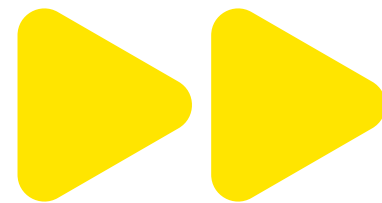
# ПАРЫ ЭЛЕМЕНТОВ

```
val nameAndPopulation = "Moscow" -> 12e6  
val pairs = List(1 -> "one", 2 -> "two", 3 -> "three")
```

# ПАРЫ ЭЛЕМЕНТОВ

```
val intAndString: (Int, String) = 1 -> "one"  
val stringAndInt: (String, Int) = intAndString.swap
```

**В этом разделе  
мы изучили кортежи**



**В следующем узнаем  
об опциональных значениях**