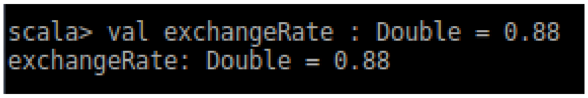
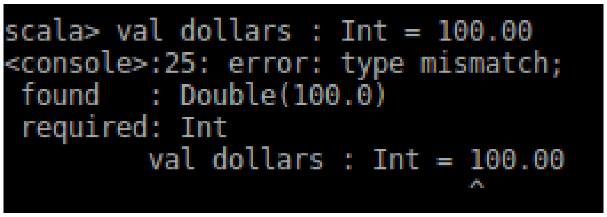
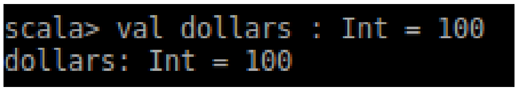
B. **Explore Scala-Spark Variables**



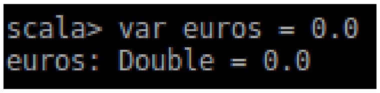
```
scala> val exchangeRate : Double = 0.88
exchangeRate: Double = 0.88
```



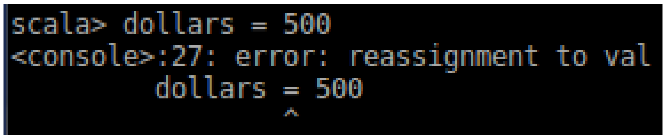
```
scala> val dollars : Int = 100.00
<console>:25: error: type mismatch;
 found   : Double(100.0)
 required: Int
         val dollars : Int = 100.00
                             ^
```



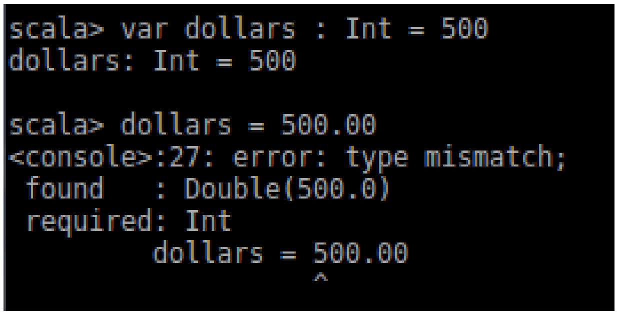
```
scala> val dollars : Int = 100
dollars: Int = 100
```



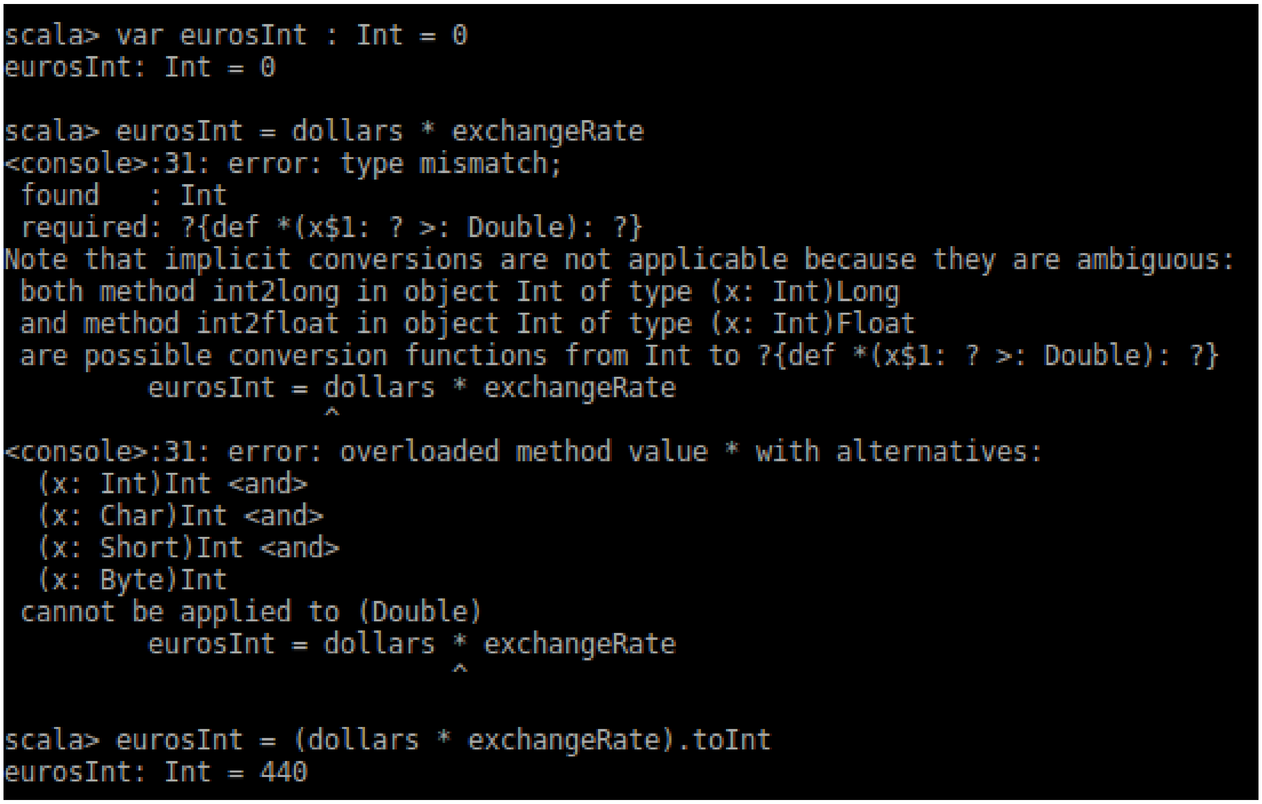
```
scala> var euros = 0.0
euros: Double = 0.0
```



```
scala> dollars = 500
<console>:27: error: reassignment to val
         dollars = 500
                 ^
```

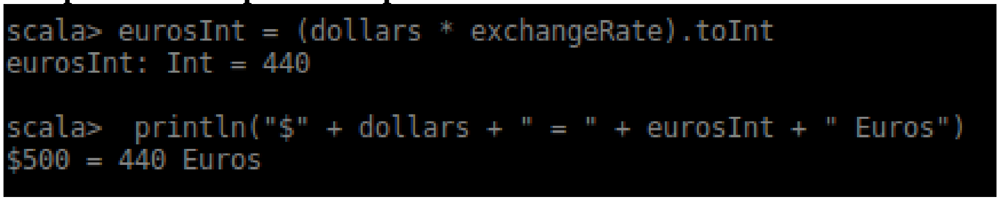


```
scala> var dollars : Int = 500
dollars: Int = 500

scala> dollars = 500.00
<console>:27: error: type mismatch;
 found   : Double(500.0)
 required: Int
         dollars = 500.00
                   ^
```

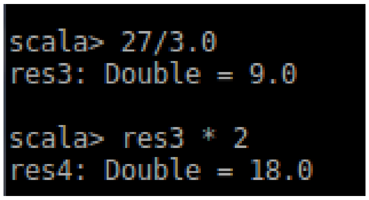


```
scala> var eurosInt : Int = 0
eurosInt: Int = 0

scala> eurosInt = dollars * exchangeRate
<console>:31: error: type mismatch;
 found   : Int
 required: ?{def *(x$1: ? >: Double): ?}
Note that implicit conversions are not applicable because they are ambiguous:
 both method int2long in object Int of type (x: Int)Long
 and method int2float in object Int of type (x: Int)Float
 are possible conversion functions from Int to ?{def *(x$1: ? >: Double): ?}
         eurosInt = dollars * exchangeRate
                    ^
<console>:31: error: overloaded method value * with alternatives:
  (x: Int)Int <and>
  (x: Char)Int <and>
  (x: Short)Int <and>
  (x: Byte)Int
 cannot be applied to (Double)
         eurosInt = dollars * exchangeRate
                            ^

scala> eurosInt = (dollars * exchangeRate).toInt
eurosInt: Int = 440
```

C. **Explore Scala-Spark Computation**

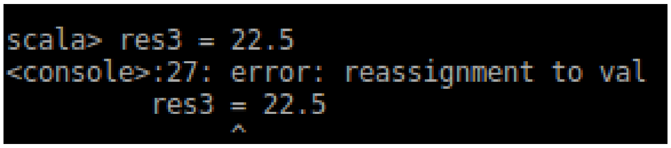


```
scala> eurosInt = (dollars * exchangeRate).toInt
eurosInt: Int = 440

scala>  println("$" + dollars + " = " + eurosInt + " Euros")
$500 = 440 Euros
```

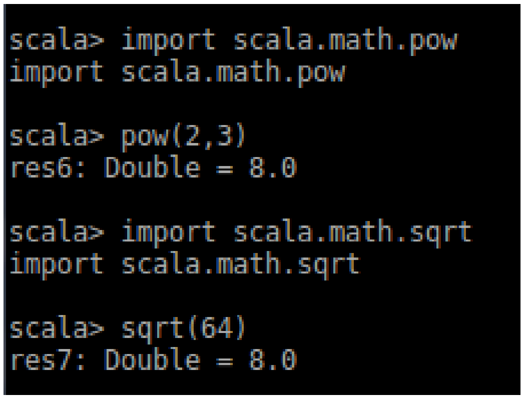


```
scala> 27/3.0
res3: Double = 9.0

scala> res3 * 2
res4: Double = 18.0
```

res3 is immutable variable

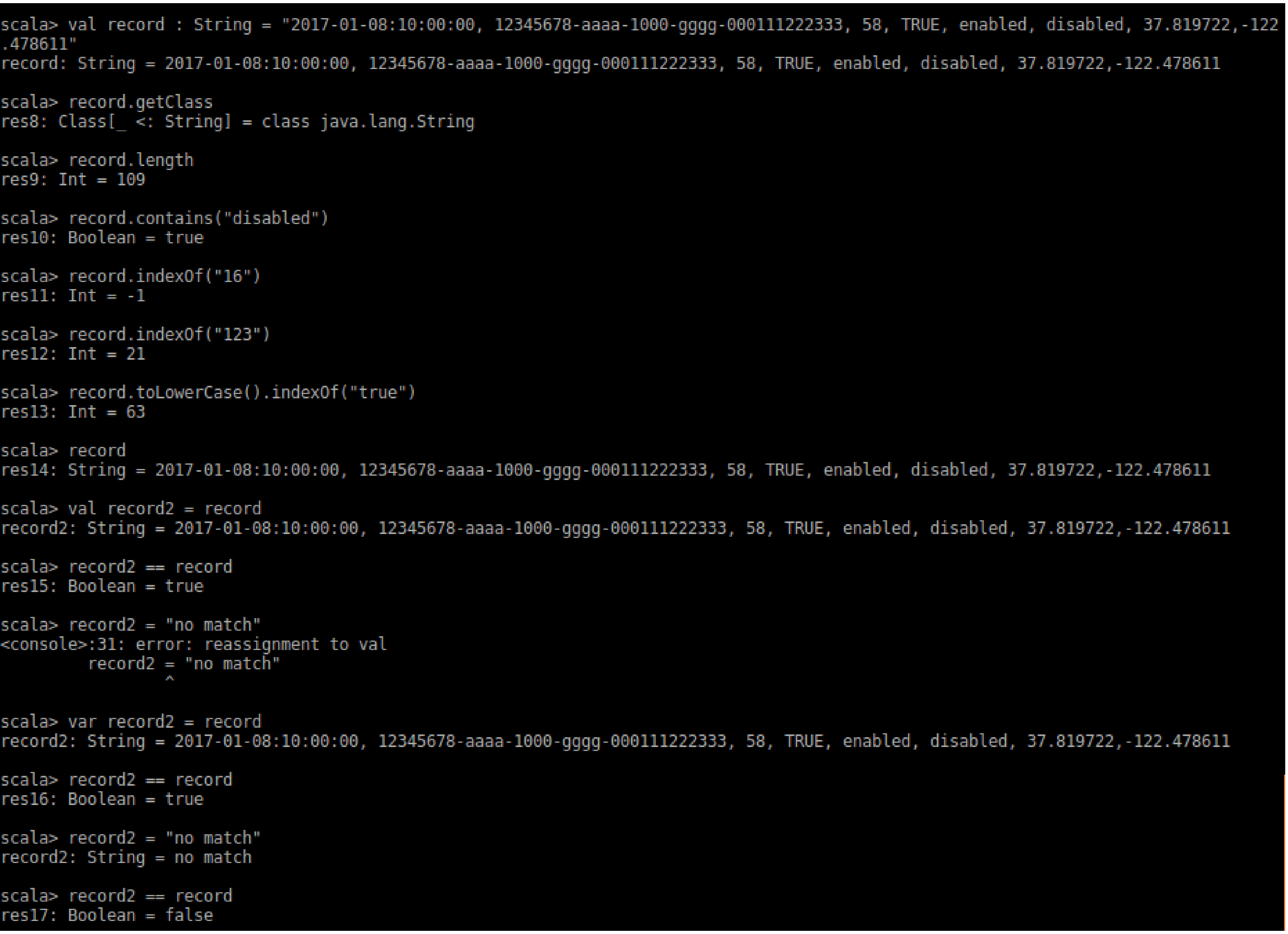


```
scala> res3 = 22.5
<console>:27: error: reassignment to val
         res3 = 22.5
              ^
```



```
scala> import scala.math.pow
import scala.math.pow

scala> pow(2,3)
res6: Double = 8.0

scala> import scala.math.sqrt
import scala.math.sqrt

scala> sqrt(64)
res7: Double = 8.0
```

D. **Explore Scala-Spark Strings**



```
scala> val record : String = "2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122
.478611"
record: String = 2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122.478611

scala> record.getClass
res8: Class[_ <: String] = class java.lang.String

scala> record.length
res9: Int = 109

scala> record.contains("disabled")
res10: Boolean = true

scala> record.indexOf("16")
res11: Int = -1

scala> record.indexOf("123")
res12: Int = 21

scala> record.toLowerCase().indexOf("true")
res13: Int = 63

scala> record
res14: String = 2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122.478611

scala> val record2 = record
record2: String = 2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122.478611

scala> record2 == record
res15: Boolean = true

scala> record2 = "no match"
<console>:31: error: reassignment to val
         record2 = "no match"
                 ^

scala> var record2 = record
record2: String = 2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122.478611

scala> record2 == record
res16: Boolean = true

scala> record2 = "no match"
record2: String = no match

scala> record2 == record
res17: Boolean = false
```