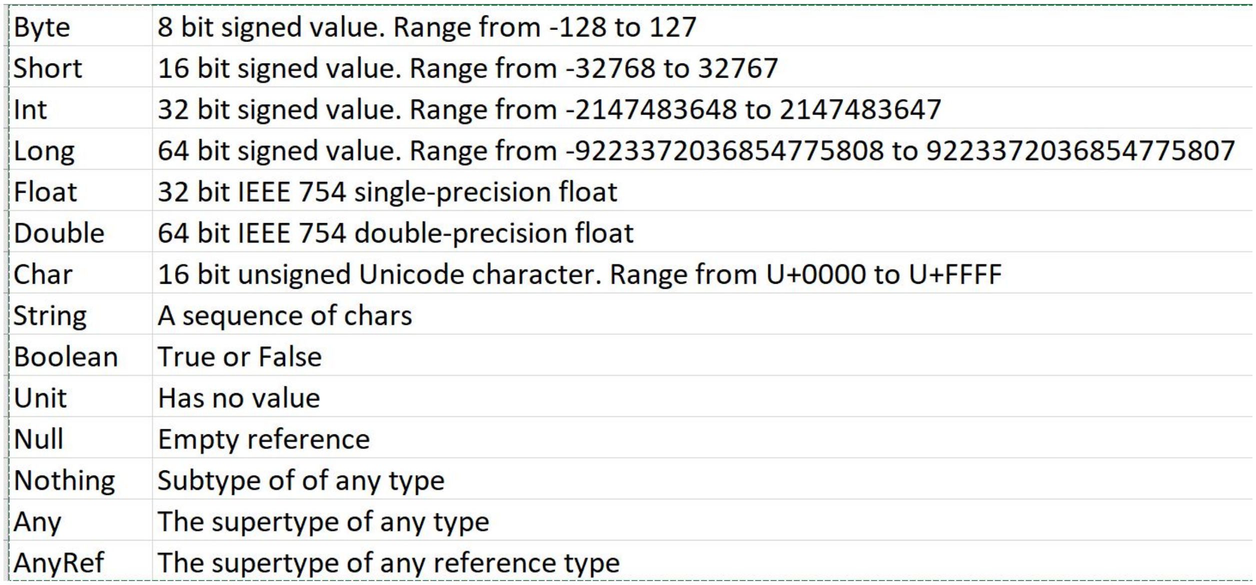
Scala has the same memory types as data types as Java, with the same memory footprint and precision. However, there are no primitive types as in Java, so you can call methods on Int, Long, etc.



Byte	8 bit signed value. Range from -128 to 127
Short	16 bit signed value. Range from -32768 to 32767
Int	32 bit signed value. Range from -2147483648 to 2147483647
Long	64 bit signed value. Range from -9223372036854775808 to 9223372036854775807
Float	32 bit IEEE 754 single-precision float
Double	64 bit IEEE 754 double-precision float
Char	16 bit unsigned Unicode character. Range from U+0000 to U+FFFF
String	A sequence of chars
Boolean	True or False
Unit	Has no value
Null	Empty reference
Nothing	Subtype of of any type
Any	The supertype of any type
AnyRef	The supertype of any reference type

Any is the supertype of all types and defines some universal methods, such as equals, hashCode, and toString. Any has two subclasses, AnyVal and AnyRef. AnyVal represents the nine value types: Double, Float, Long, Int, Short, Byte, Char, Unit, and Boolean. The unfamiliar Unit is a value type which has no meaning or value. There is one instance of Unit, (). As all functions in Scala must return a value, Unit can be used for what would otherwise be a function with a void return type. AnyRef represents reference types, including non-value types. If Scala is used in a Java runtime environment, AnyRef corresponds to java.lang.Object.

Types can be based unidirectionally within their AnyVal or AnyRef distinction. Nothing is a subtype of all types and is thus the “bottom type”. Nothing is commonly used to signal non-termination, such as a thrown exception or program exit. Null is a bottom subtype of any AnyRef subtype.