Variables and literals

January 2, 2025

[1]:

```kotlin
// Mutable variables using `var`
var greeting: String = "Welcome"
println(greeting)
greeting = "Goodbye"
println(greeting)
```

```
Welcome
Goodbye
```

```
greeting = Goodbye
greeting = Goodbye
```

[1]:

```
Goodbye
```

[2]:

```kotlin
// Immutable variables using `val`
val number: Int = 10
println(number)
number = 20 // This would cause a compilation error
```

[3]:

```kotlin
// Examples of integer literals
val decimal: Int = 42          // Decimal literal
val hex: Int = 0x2A            // Hexadecimal literal
val binary: Int = Integer.parseInt("101010", 2) // Binary literal (converted from binary string)
val longNumber: Long = 123456789L // Long literal
println(s"Decimal: $decimal, Hex: $hex, Binary: $binary, Long: $longNumber")
```

```
Decimal: 42, Hex: 42, Binary: 42, Long: 123456789
```

```
decimal = 42
hex = 42
binary = 42
longNumber = 123456789
```

[3]:

```
123456789
```

[4]:

```
// Examples of floating-point literals
val pi: Float = 3.14f        // Float literal
val scientific: Double = 1.2e-5  // Scientific notation (Double)
val shortForm: Float = .5f  // Short form
val doubleVal: Double = 10.0
println(s"Pi: $pi, Scientific: $scientific, Short Form: $shortForm, Double: $doubleVal")
```

```
Pi: 3.14, Scientific: 1.2E-5, Short Form: 0.5, Double: 10.0
```

```
pi = 3.14
scientific = 1.2E-5
shortForm = 0.5
doubleVal = 10.0
```

[ ]:

```
// Examples of Boolean literals
val isActive: Boolean = true
val isComplete: Boolean = false
println(s"Is Active: $isActive, Is Complete: $isComplete")
```

[ ]:

```
// Examples of Character literals
val letter: Char = 'X'
val newline: Char = '\n'
println(s"Letter: $letter, Newline: ${newline}Here")
```

[ ]:

```
// Examples of String literals
val simpleString: String = "Scala is great!"
val escapedString: String = "This is a \"quoted\" string."
val multilineString: String = """This string spans
multiple lines."""
println(simpleString)
println(escapedString)
println(multilineString)
```

[ ]:

```
// Null value examples
val nullableString: String = null
val nullableInt: Integer = null
println(s"Nullable String: $nullableString, Nullable Int: $nullableInt")
// Output:
// Nullable String: null, Nullable Int: null
```

[ ]:

```
```