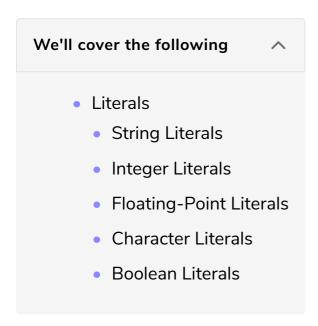
# Strings and Other Literals

In the following lesson, you will be introduced to literals.



By now, you must have noticed that we haven't discussed strings. Strings do not fall under anyVal in the data type hierarchy. This raises the question: what are strings?

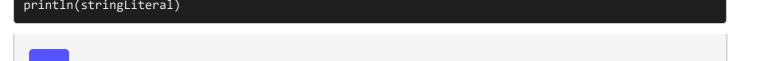
The single word answer is 'literals'.

## Literals #

A literal is defined as taking anything in its most usual and basic sense. Mapping this onto computer programming, literals are fixed values appearing directly as is in the source code. For example, "Hello World", 5, and 'A' are all literals.

## String Literals #

Strings literals are a combination of characters surrounded by double quotation marks. The syntax for declaring a variable of type String is the same as declaring a variable of any basic value type.



### Integer Literals #

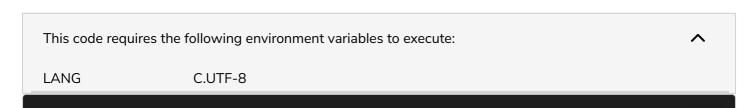
Integer Literals are used with the value types <code>Long</code>, <code>Int</code>, <code>Short</code>, and <code>Byte</code> and come in two forms: decimal and hexadecimal. Decimal numbers have a base 10 while hexadecimal numbers have a base 16. The way an integer literal begins indicates the base of the number. If the number begins with a 0x or 0X, it is hexadecimal and may contain the numbers from 0 through 9 as well as upper or lowercase letters from A to F. Numbers that do not start with 0x or 0X are decimal by default.



One thing to remember is that Scala will always print an integer literal as decimal regardless of how it is declared.

### Floating-Point Literals #

Floating-point literals are used with <code>Double</code> and <code>Float</code>. They are made up of decimal digits and have the option of adding exponents using either an <code>e</code> or <code>E</code> followed by the exponent.



```
val floatLiteral: Float = 1.2345F // The F at the end is for Float
val somethingBigger: Double = 1.2345e1
val evenBigger: Double = 1.2345e4

// Driver Code
println(floatLiteral)
println(somethingBigger)
println(evenBigger)
```

#### Character Literals #

Character literals are used with Char and are made up of a single Unicode character surrounded by single quotation marks.



Escape sequences, as shown in the table below, also represent character literals.

Literal	Meaning
\n	linefeed
\b	backspace
\t	tab
\f	form feed
\r	carriage return

\"	double quote
\'	single quote
\\	backslash

#### Let's look at some examples:



#### **Boolean Literals** #

Boolean literals are used with Boolean. They are of two types: true and false.



Let's complete our discussion on variables and types with type inference in the next lesson.