

L-5

Integer and Floating-Point Conversion

Conversions between integer and floating-point numeric types must be made explicit:

```
let three = 3
let pointOneFourOneFiveNine = 0.14159
let pi = Double (three) + pointOneFourOneFiveNine
// pi equals 3.14159, and is inferred to be of type Double
```

Here, the value of the constant `three` is used to create a new value of type `Double`, so that both sides of the addition are of the same type. Without this conversion in place, the addition would not be allowed.

Floating-point to integer conversion must also be made explicit. An integer type can be initialized with a `Double` or `Float` value:

```
let integerPi = Int (pi)
// integerPi equals 3, and is inferred to be of type Int
```

Floating-point values are always truncated when used to initialize a new integer value in this way. This means that 4.75 becomes 4, and -3.9 becomes -3.

Note

The rules for combining numeric constants and variables are different from the rules for numeric literals. The literal value 3 can be added directly to the literal value 0.14159, because number literals don't have an explicit type in and of themselves. Their type is inferred only at the point that they're evaluated by the compiler.