```
/**
 Assignment Operator
 We have already used the assignment operator for simple values.
 var value = 17
 value = 1337
 */

/**
 Exercise 2.1
 What would happen if you used the assignment operator in the following way? (This is mainly an
 exercise for Objective-C coders.)
 var value = 17
 if value = 19 {
 print("Whatever and stuff.")
 }
 */
var value = 17
//if value = 19 { //Error Use of '=' in a boolean context, use '==' instead
if value == 19 {
    print("Whatever and stuff.")
}

/**
 Arithmetic Operators
 Everyone knows how these work. They're basically the same as in other lanugages. +, -, *, /
 */

/**
 Exercise 2.2
 Use all the arithmetic operators in a single statement and assign the result to a constant.
 */
let a = 1 + 2       // equals 3
let b = 5 - 3       // equals 2
let c = 2 * 3       // equals 6
let d = 10.0 / 2.5  // equals 4

/**
 Exercise 2.3
 Append the string "Larionov" to the end of the string "Igor " and assign the resulting string to a
 constant.
 */
let context = "Igor " + "Larionov"

/**
 Exercise 2.3
 What happens if you try the unary increment operator (e.g. value++) from other C-like languages?
 var value = 0
```

```
 value++
 */
var value2 = 0
//value2++ // Error Cannot find operator '++' in scope; did you mean
 '+= 1'?

/**
 Exercise 2.4
 How do you check if two strings are equal in Swift?
 */
if "Hi" == "Hi" {
    print("This is the way to check if strings equal")
}

/**
 Exercise 2.5
 Let's compare some tuples. Guess if these are true or false:
 (1, "zebra") < (2, "apple")
 (2, "zebra") < (1, "apple")
 (3, "apple") < (3, "bird")
 (4, "dog") == (4, "dog")
 (4, "dog") == (4, "cat")
 */
(1, "zebra") < (2, "apple") //true
(2, "zebra") < (1, "apple") //false
(3, "apple") < (3, "bird")  //true
(4, "dog") == (4, "dog")    //true
(4, "dog") == (4, "cat")    //false

/**
 Ternary Conditional Operator
 Exercise 2.6
 Use the ternary conditional operator to assign the correct number of days in a year to the constant
 daysInYeardepending on the value of leapYear.
 var leapYear = true
 */
//leap year = a year will have 366 days
var leapYear = true
let daysInYeardepending = leapYear ? 366 : 365

/**
 Nil-Coalescing Operator
 Coalescing is not the easiest word to spell. You could think of it as the default operator, I guess.
 */

/**
 Exercise 2.7
```

```
 Use the nil-coalescing operator ?? to provide a fallback value if a value for the optional variable
 name has not been provided.
 // The first name is optional in this example.
 var firstName: String? = nil

 // The last name is however not optional.
 var lastName: String = "Jones"

 // Use ?? operator here to provide a fallback value,
 // if no first name has been provided.
 // For example, the default value could be "Dr.",
 // because this code is to be used at a medical conference. var name: String = firstName

 name += " " + lastName

 print(name)
 */
// The first name is optional in this example.
var firstName: String? = nil

// The last name is however not optional.
var lastName: String = "Jones"

// Use ?? operator here to provide a fallback value,
// if no first name has been provided.
// For example, the default value could be "Dr.",
// because this code is to be used at a medical conference.
var name: String = firstName ?? "Dr."
name += " " + lastName
print(name)


/**
 Range Operators
 Exercise 2.8: Closed Range Operator
 Define a closed range, e.g. for an amplifier volume knob that goes from 0 to 11. The range should
 include both 0 and 11, because this amplifier really does go to 11.
 */
for i in 0...11 {
    print(i)
}

/**
 Exercise 2.9: Open Range Operator
 Define an open range, e.g. for an amplifier volume knob that goes from 0 to 10. The range should
 include both 0 and 10, but not 11.
 */
for i in 0..<11 {
```

```
    print(i)
}
```