codecademy

Data Types & Variables

Mutable Variables

A mutable variable is declared with the `var` keyword and represents a value that is expected to change throughout a program.

```
var age = 25
age = 26
```

Immutable Variables

An immutable variable is declared with the `val` keyword and represents a value that must remain constant throughout a program.

```
val goldenRatio = 1.618
```

Type Inference

When a data type is not specified in a variable declaration, the variable's data type can be inferred through type inference.

```
// The following variable is assigned a te
xt value within double quotes, thus the in
ferred type is String

var color = "Purple"
```

String Concatenation

String concatenation is the process of combining Strings using the `+` operator.

```
var streetAddress = "123 Main St."
var cityState = "Brooklyn, NY"

println(streetAddress + " " + cityState)
// Prints: 123 Main St. Brooklyn, NY
```

String Templates

String templates contain String values along with variables or expressions preceded by a `$` symbol.

```
var address = "123 Main St. Brooklyn, NY"
println("The address is $address")
// Prints: The address is 123 Main St. Bro
oklyn, NY
```

Built-in Properties and Functions

The Kotlin String and Character data types contain various built-in properties and functions. The `length` property returns the number of characters in a String, and the `capitalize()` function capitalizes the first letter of a String.

```
var monument = "the Statue of Liberty"

println(monument.capitalize()) // Prints: The Statue of Liberty
println(monument.length) // Prints: 21
```

Character Escape Sequences

Character escape sequences consist of a backslash and character and are used to format text.

- `\n` Inserts a new line
- `\t` Inserts a tab
- `\r` Inserts a carriage return
- `\'` Inserts a single quote
- `\"` Inserts a double quote
- `\\` Inserts a backslash
- `\$` Inserts the dollar symbol

```
print("\"Excellent!\" I cried. \"Elementary,\" said he.")

// Prints: "Excellent!" I cried. "Elementary," said he.
```

Arithmetic Operators

The arithmetic operators supported in Kotlin include `+` addition, `-` subtraction, `*` multiplication, `/` division, and `%` modulus.

```
5 + 7  // 12
9 - 2  // 7
8 * 4  // 32
25 / 5 // 5
31 % 2 // 1
```

Order of Operations

The order of operations for compound arithmetic expressions is as follows:

1. Parentheses
2. Multiplication
3. Division
4. Modulus
5. Addition
6. Subtraction

When an expression contains operations such as multiplication and division or addition and subtraction side by side, the compiler will evaluate the expression in a left to right order.

```
5 + 8 * 2 / 4 - 3 // 6
3 + (4 + 4) / 2   // 7
4 * 2 + 1 * 7     // 15
3 + 18 / 2 * 1    // 12
6 - 3 % 2 + 2     // 7
```

Augmented Assignment Operators

An augmented assignment operator includes a single arithmetic and assignment operator used to calculate and reassign a value in one step.

```
var batteryPercentage = 80

// Long Syntax
batteryPercentage = batteryPercantage + 10


// Short Syntax with an Augmented Assignme
nt Operator
batteryPercentage += 10
```

Increment and Decrement Operators

Increment and decrement operators provide a shorthand syntax for adding or subtracting `1` from a value. An increment operator consists of two consecutive plus symbols, `++`, meanwhile a decrement operator consists of two consecutive minus symbols, `--`.

```
var year = 2019
year++ // 2020
year-- // 2019
```

The Math Library

The `Math` library, inherited from Java, contains various mathematical functions that can be used within a Kotlin program.

```
Math.pow(2.0, 3.0)  // 8.0
Math.min(6, 9)      // 6
Math.max(10, 12)    // 12
Math.round(13.7)    // 14
```