Ternary Operator

In this lesson, we'll understand how the ternary operator serves as a substitute for the if-else expression.

Introduction

Dart has an operator that lets you concisely evaluate expressions that might otherwise require `if-else` statements.

The operator that we will be discussing is the *ternary operator* which is represented by `?:` .

Syntax

Let's take a look at how this operator is used.

```
condition ? expression1 : expression2
```

If **condition** is `true` , **expression1** is evaluated and its value is returned. Otherwise, **expression2** is evaluated and its value is returned.

Example

Let's look at an example where we have two integers, `a` and `b` . If `a` is greater than `b` , we want `b` to be subtracted from `a` . Otherwise, we want `a` to be subtracted from `b` .

```
main() {
  var a = 5;
```

```
  var b = 2;

  var result = a > b ? a - b : b - a;

  print(result);
}
```

On **line 5** we are using the `?:` operator. `a > b` is representing the condition while `a - b` is representing the first expression and `b - a` is representing the second expression.

The value assigned to `a` is **5** and the value assigned to `b` is **2**. Hence, `a > b` is `true`. As the condition is true, the first expression (`a - b`) will be evaluated and its result (**3**) will be stored in `result`.

> **Try it Yourself:** Modify the values of `a` and `b` in the code above so that the second expression is evaluated.

This example could also have been written using an `if-else` statement.

```
main() {
  var a = 5;
  var b = 2;
  var result;

  if(a > b){
    result = a - b;
  } else {
    result = b - a;
  }

  print(result);
}
```

If `a` and `b` are equal, the `else` statement will execute.

With this, our discussion on conditionals comes to an end. In the next lesson, we have a challenge for you to solve for yourself.