

Swift is a strongly typed language. This means that the compiler checks the types of expressions and ensures that they are compatible. This can help to prevent errors and make your code more reliable.

For example, the following code will not compile:

Swift

```
var myString = "Hello, world!"  
var myNumber = 10
```

```
myString = myNumber // This line will not compile because myString is a  
String and myNumber is an Int.
```

The compiler will give an error because the types of `myString` and `myNumber` are not compatible.

However, the following code will compile:

Swift

```
var myString = "Hello, world!"  
var myNumber: Int = 10
```

```
myString = String(myNumber) // This line will compile because the type  
of myNumber has been converted to String.
```

The compiler will allow this line to compile because the type of `myNumber` has been explicitly converted to `String`.

Strong typing can help to make your Swift code more reliable and prevent errors. However, it can also make your code more verbose and explicit.

Here are some additional benefits of using a strongly typed language like Swift:

- It can help to prevent errors at compile time, which can save you a lot of time and debugging effort.
- It can make your code more readable and maintainable, as it is clear what type of data each variable stores.
- It can help to improve the performance of your code, as the compiler can optimize the code for the specific types of data that are being used.

Overall, strong typing is a valuable feature of Swift that can help you to write

more reliable, readable, and efficient code.