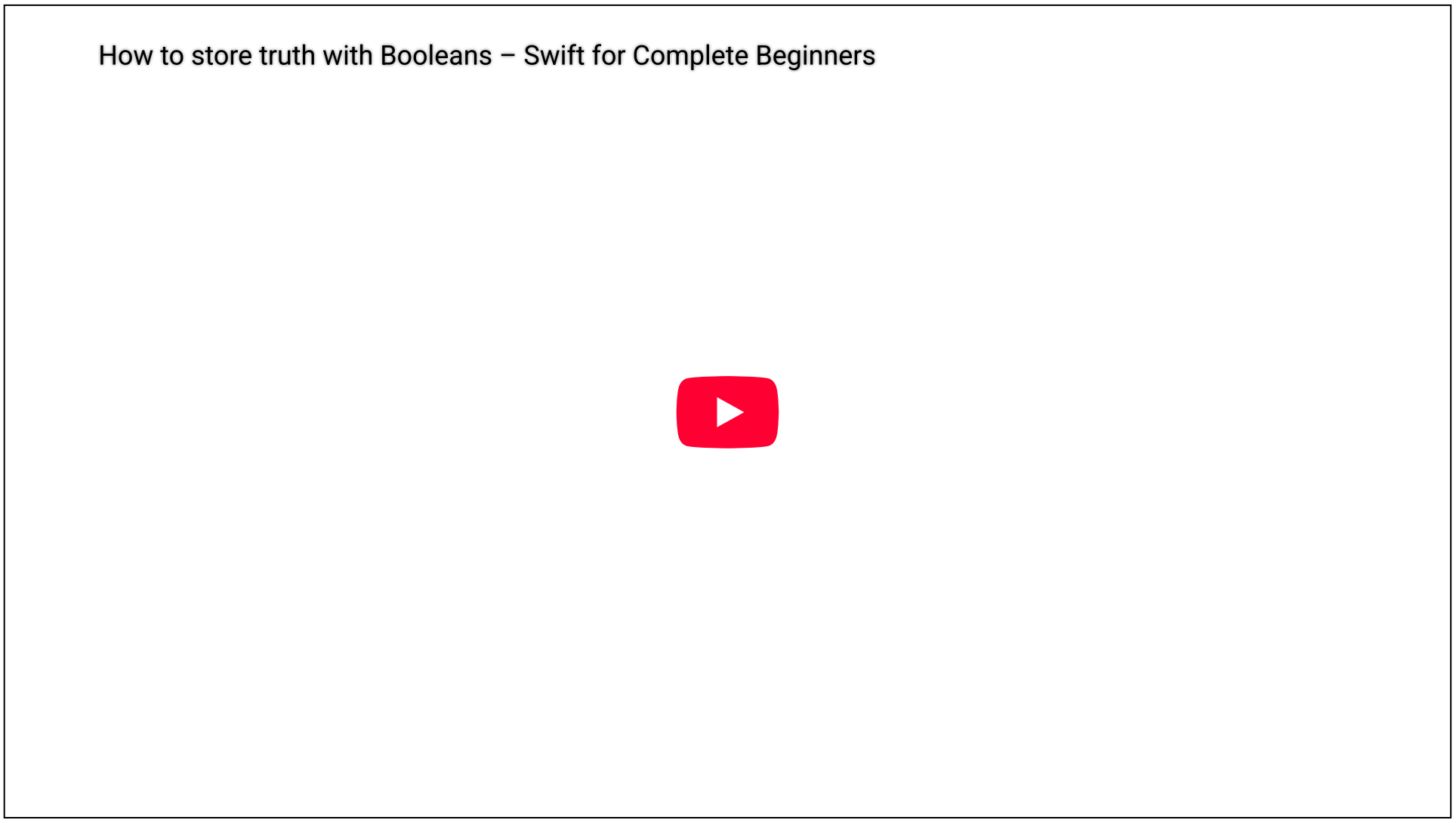
< How to store decimal numbers

How to join strings together >

How to store truth with Booleans

Paul Hudson @twostraws October 25th 2021

Updated for Xcode 16.4



So far we've looked at strings, integers, and decimals, but there's a fourth type of data that snuck in at the same time: a very simple type called a Boolean, which stores either true or false. If you were curious, Booleans were named after George Boole, an English mathematician who spent a great deal of time researching and writing about logic.

I say that Booleans snuck in because you've seen them a couple of times already:

```
let filename = "paris.jpg"
print(filename.hasSuffix(".jpg"))

let number = 120
print(number.isMultiple(of: 3))
```

Both **`hasSuffix()`** and **`isMultiple(of:)`** return a new value based on their check: either the string has the suffix or it doesn't, and either 120 is a multiple of 3 or it isn't. In both places there's always a simple true or false answer, which is where Booleans come in – they store just that, and nothing else.

Making a Boolean is just like making the other data types, except you should assign an initial value of either true or false, like this:

```
let goodDogs = true
let gameOver = false
```

You can also assign a Boolean's initial value from some other code, as long as ultimately it's either true or false:

```
let isMultiple = 120.isMultiple(of: 3)
```

Unlike the other types of data, Booleans don't have arithmetic operators such as **+** and **–** – after all, what would true + true equal? However, Booleans do have one special operator, **!**, which means "not". This flips a Boolean's value from true to false, or false to true.

For example, we could flip a Boolean's value like this:

```
var isAuthenticated = false
isAuthenticated = !isAuthenticated
print(isAuthenticated)
isAuthenticated = !isAuthenticated
print(isAuthenticated)
```

That will print "true" then "false" when it runs, because **isAuthenticated** started as false, and we set it to *not* false, which is true, then flip it again so it's back to false.

Booleans *do* have a little extra functionality that can be useful. In particular, if you call **toggle()** on a Boolean it will flip a true value to false, and a false value to true. To try this out, try making **gameOver** a variable and modifying it like this:

```
var gameOver = false
print(gameOver)

gameOver.toggle()
print(gameOver)
```

That will print false first, then after calling **toggle()** will print true. Yes, that's the same as using **!** just in slightly less code, but it's surprisingly useful when you're dealing with complex code!

< How to store decimal numbers

How to join strings together >

Was this page useful? Let us know!

☆☆☆☆☆

Average rating: 4.8/5

You are not logged in

Log in or create account