Primitive types in memory

Selecting transcript lines in this section will navigate to timestamp in the video

- If we need to store more than one character, then we would have to store our data in something called a string, which is a data type composed of a sequence of characters. Think of those happy birthday banners that are strung together with a string. That's what we're talking about here. Each letter flag is a character and the string literally ties them together and groups them in a specific order. Now strings are stored a little bit differently than everything we've worked with so far, so we'll come back to them. So how exactly does an int, or a double, or a boolean, or character relate to ones and zeros in bits and bytes? Well, each of these data types are a certain size, or in other words, they each contain a specific number of ones and zeros. We call a single one or zero a bit. In fact, a boolean is represented with a single bit, a single one or zero. This is because a boolean is either a true or false value. In this case, zero stands for false and one stands for true. This is the only piece of information we need, and thus the only piece of data we need to store. So a boolean is one bit. In Swift, an int or int32 is 32 bits. Here's some examples. Every int consists of 32 ones and zeros. This means that every combination of 32 zeros and ones represents a particular integer. Since we count every combination of ones and zeros as a possible value, if a data type takes up more bits, it will have more possible values. No matter what int or int32 value we store, the data will always be 32 bits in size. In other words, the specific value stored has no effect on how much memory is used. An int or int32 will always take up 32 bits. So ultimately everything boils down to ones and zeros, but they are abstracted to us for the most part through programming languages. This lets us focus on building bigger programs and not worry about binary for the most part. Now, most of everything we talked about so far is a primitive type. In other words, ints, doubles, longs, floats, shorts, booleans, and chars. These can also be called basic or value types because these are types that have a fixed size that do not depend on the data inside of them. Like we said before, a double in C#, or a double in Java is always 64 bits no matter what value it holds. However, there are data types that can be built out of other data types.