JAVASCRIPT - LESSON 1, TOPIC 3: JOINING STRINGS

INTRODUCTION
UNDERSTANDING CONCATENATION
DEMONSTRATION

INTRODUCTION

00:00 Welcome back. This is Lesson 1, Topic 3, Joining Strings, and specifically the concepts of concatenation and interpolation. You know the drill.

00:12 Let's look a bit at the structure of the coming video. So first, we're going to look at the structure of the coming video. Look at the concept of concatenation.

00:18 What is it? How do we use it? And secondly, we're going to do a little deep dive into interpolation. But before we jump into either of those, let's just have a quick refresher of what I mean when I say strings.

00:33 Can you remember from the last session? Strings are a type of data in programming that contains text. And we know when we see one because we look for those quotation marks surrounding it.

00:45 So a string is a data type that holds a text object. But let's look now at the idea of concatenation.

UNDERSTANDING CONCATENATION

00:53 So as a general whole, concatenation is the joining of some strings together. More specifically, concatenation is used to join together two or more strings of text to create a longer string of text.

01:06 So, consider the idea that sometimes we want to print more than just hello and an emoji to our console, right?

01:14 Wouldn't it be handy if we could add various strings together and get a fuller output in our console? This is the concept of concatenation.

01:24 And before we jump into a demonstration, I just want to flag that concatenation isn't just used for strings. So, that's one of its predominant uses.

01:33 You might have a string of hello and a string of Emily, and I'm going to use concatenation to add them together and create hello Emily.

01:41 But we can also use concatenation to combine strings with variables. So, if you remember from our last session, we looked at how we create constants.

01:51 For example, a constant of message equals we got this. So we can perform concatenation in the same way. We can have a string that says hello and we can add our variable to the end of it and it will print out hello, we got this.

02:06 But let's look at that in action. Okay. So in our screenshot here, we've got a bit of a demonstration of concatenation in action.

DEMONSTRATION

02:16 Let's look at it, do an overview and then we're going to jump into an actual walkthrough in CodePen in just a moment.

02:22 So as you can see, we're first starting by setting up two constants. We have a constant greeting which holds the value hello and a constant emoji which holds our little hand emoji.

02:36 Then the thing we're really interested in here is line six. So this is where we're performing the concatenation. First of all, we're creating a new constant and giving it a name message and then we're setting its value to equal emoji.

02:53 So the wavy hand plus a space plus hello plus another space and plus our emoji again. So what's happening here is we're taking those individual data that we've created

higher up these variables and we're adding them together to create a new variable that's longer.

03:16 So in this case we're console logging it and it's returning wavy hand hello wavy hand.

03:27 Let's look at this more in-depth. So I've hopped over to CodePen and let's break this down line by line. So we have our comment declaring our constants and we know it's a comment because of those two forward slashes at the start.

03:43 And then on lines three and four, we're creating not one, but two separate constants. Variables. So we've got greetings and emojis.

03:54 Line six, we have another comment, concatenate to create a full message. Now, this is the line that we're particularly interested in.

04:04 So let's look at line seven in depth. So first we're creating a constant and we're giving it the name message.

04:13 That's right. Then we have our equals sign to show us what value we're going to, we're about to give that message.

04:20 And as you can see here, we're reusing constant names from the lines above and adding them in to create a bigger, longer string.

04:33 So we're saying, okay, so the variable message, I want its value to be emoji plus, and then here you might be thinking, hmm, why have I got two quotation marks and nothing within them?

04:47 Well, that's because we need to add a space so that our sentence makes grammatical sense. So we're saying the one emoji and then a space and then the word hello and then another space and then an emoji.

05:02 And as you can see, it's added in the spaces nicely down here for us. So our string makes grammatical sense.

05:09 But let's have a look. What happens if I take these spaces out? Have you got any guesses before I hit run?

05:17 Again, that's another really fun part of coding is kind of testing your hypothesis. Let's see what happens. Off it goes and aha!

05:29 As we can see, if we don't add those spaces in, we just add the new string or variable that we're adding on directly to the end of the previous.

05:40 String or variable. So we need to make sure we add these empty spaces in and plus signs. So we've got emoji space, hello plus, and then I'm going to put another empty string.

05:58 And don't forget, I need another plus sign for our final emoji and another space there. There we go. So that we can make sure that it makes grammatical sense.

06:10 Now that we've looked a bit at our first method of joining strings together in concatenation, let's move to the second way we can do that.

06:19 This is called interpolation. So let's start with a definition. Interpolation helps you combine text and variable variables more smoothly. So let's look at a bit of a context for why interpolation exists.

06:33 So concatenation, the example we looked at before with the plus signs and adding an empty string to represent a space, this can get really messy.

06:43 Sometimes you want to combine lots of different text and a few different variables and it's really easy to forget a space between sentences.

06:53 Or maybe you've got some tricky punctuation you want to include as well. Imagine having lots of strings and variables and plus signs in between and always adding an extra space, it can quickly get a bit long and unwieldy.

07:10 So sometimes when you want to create a more complicated string or a multi-line string we use what's known as interpolation.

07:19 Let's see what this looks like. So using the same example from before where we have our two constants up the top here we have our greeting and our emoji.

07:34 This time on line six, instead of using concatenation, we're using interpolation. So before the equals, it looks pretty similar, right?

07:45 We're declaring a constant. We're giving it the name of message, but after you can see we've got some very different syntax here.

07:53 So just like with concatenation where we look out for those plus signs as an indication that something's being concatenated, you can spot interpolation because of its use of these backticks.

08:06 Now, you might be thinking, what's a backtick? Because I'd never heard of it before I learned JavaScript. But it's these little, they look a bit like speech marks, don't they?

08:16 Single speech marks. These little symbols here are called a backtick. And we want to put our whole string inside of those.

08:23 So, instead of our double quotation marks that we use commonly in regular strings, interpolation requires these backticks. And then, instead of just declaring our constant name, we have to wrap it inside this slightly odd-looking syntax.

08:42 We have a dollar sign and two sets of what are called curly braces. So, we have regular brackets, you know, our parentheses.

08:52 These kind of fancier-looking ones are called curly braces. And this is how we perform the interpolation. So, in this case, as you can see, we're getting the same output, but through a very different method.

09:06 So, let's put this into practice in CodePen now.

09:14 In Code Pen, so we have our example open here, and as you can see, the first few lines look exactly the same as they did in our concatenation example.

09:26 We're creating variables and we're giving them values. However, on line 6 here, we start off the same. So we declare a constant, we give it the name of the message, but then things start to look quite different after the equals sign.

09:42 So in this case, we're using interpolation. Instead of concatenation. And as I said before, one really obvious thing to look out for, to give you a hint that we're using interpolation, is that everything goes inside of these backticks.

09:57 And our objects are declared using a dollar sign. And a set of curly brackets. So first, let's just run this and show you that the output is the same as with concatenation.

10:10 So, yep, as you can see, we have exactly the same output below. But if you're cool when we were using concatenation, we had const message equals, and then we used our constant name, so emoji plus, and then we had to add in our spaces because it wouldn't automatically add them in for us.

10:34 Well, if you look at the interpolation example, you can see that we have our spaces added here, and that's actually because we've put the spaces between our constants.

10:46 Above. So let's have a little look at what happens if we take those out. So if we just get rid of them and run the code again, what do you suspect will happen to our output?

10:58 Let's see. You can see we don't get any. Any spaces in our output. So when we create a new string using interpolation, we have to add in the kind of grammar within these backticks and it will take it into account.

11:19 So if I put a random comma in here as well and run that again, we should see it adds it to our output.

11:28 So that's how it varies from concatenation. Well done. You've reached the end of this video and now it's time for another quiz on how we join strings together.