**7-16-cli-colors-chalk**

**Ahmad Awais:** Okie dokie. So we have a CLI called in whenever we run the CLI, it prints this particular string, which is actually information about me, as you can see here. And I'm going to go ahead and link this package so I can run the CLI. Okay. Now, if I type in awais, you can see it prints out the information about me.

One thing I can see as an improvement here is what if we want to add colors to our name or, you know, to Twitter, GitHub, blog. So how about if we add colors and dim, a little bit of unimportant stuff like these URLs or whatnot. How would you go about and do that? For that I recommend this package called chalk.

This package is actually built by Sindre Sorhus. He's actually an excellent open sourcer. If you see a package built by him, you can basically trust that this package is going to be great. Okay. So terminal string done right is what chalk does. It helps you style your string inside the terminal in whatever way you think is possible.

For example, you can bold, dim, italicize, underline, inverse the text colors. And even, uh, you know, allows you to add background colors and whatnot. It has a highly expressive API. It's quite performant. Allows you to use all 256 true color support, and it auto detects if your terminal has color support or not.

So it can basically disable itself if there is no color support or whatnot. And as of now, it is being used by about, I think 57,000 packages. So, uh, you can assume with the weekly downloads of 14 million. You can see it is actually quite a popular package. So how about we go ahead and install chalk as a dependency. For that I'm going to type in npm install chalk.

And if you take a look at package.json file, chalk was added as a dependency. Okay, now let's go to our index.js file and require chalk. One more thing I'm going to do is to make things easier, I'm going to create a copy of console log call log, and it would just be called console log. So now, instead of writing console log, I will be able to write just log.

So I'm going to remove this thing from here. And if I still type in awais, you will see my CLI is still working. So this is not a compulsory thing to do. This is just so that I can fit all the text in this recording screen. You can easily keep using console log and not confuse it with log or whatnot.

Okay. How chalk works is that inside of console log, you can start coloring different colors. How about coloring your name. For that since this is a string literal, I am able to run JavaScript inside of it using these curly brackets and the dollar sign. And inside here, I'm going to run chalk.blue, and then here, I'm going to give it that string which was initially awais.

Now if I run my CLI again, and as you can see the name of my CLI is now turned into this blue color. How about we go ahead and dim a couple of things. I'm again going to put all of this text in curly brackets. A dollar sign here, and then let's cut this thing with chalk.dim. And then add a string again.

And there you go. Now, if I run my CLI again, you can see the bio text has been dimmed, so it doesn't pop out in your face, you know, with all this white color thingy or whatnot. So how about we keep going, add a couple more colors. So what I'm going to do is I'm going to add a couple of multi cursors and then do the same thing here.

Let me cut these strings. I'm going to say chalk.dim. And then I'm going to paste all of these strings. Now, if I run awais again, so you can see all of the important information has different colors and the information that I didn't really need highlighted is about dim down. And how about we add the Twitter blue color here, the GitHub green color hair, and maybe something of a purple color to this particular string blog.

So chalk actually gives you a particular way to use hex colors. For example, if you go down in its documentation, you can see that you're able to use a keyword. For example, chalk.keyword. And then there are name of that color, or you can use an RGB function to create that RGB color and use an underline to create an underlined text or whatnot. Or my favorite,

you can use the hex function and define the hex code. And then your texts would be of this particular hex color. So how about we go ahead and do that? Uh, but before that, let me find out what is the Twitter brand color. For that I have opened up brandcolors.net, and I'm going to search for Twitter. And here it is, this is the blue color that Twitter uses.

Now let's add that color using chalk.hex to this particular Twitter string. For that let's create this thing again. Put it in this chalk.hex. And the hex I'm going to use is this sign. And I think this should be about it. Let's see if this is working. Oh yeah, we have got that particular Twitter, you know, bluish color here.

Maybe we can make it a little bit bolder by using .bold. And there you go. Now it's bold and now you can actually see this color. How would I make this one bold as well? So you can see my name there. There you go. How about we create bgBlue? Let's see how that looks. Oh, there's an error. Let me see what the documentation says about it.

Okay. Turns out that bgBlue is not a color because as you can see, this is a sort of a camel case here where bg and then the capital case of that initial word of the color. So I think if I turn this into the capital B, it is going to just work and there you go. I think I'm going to add a space here and a space here.

And if I run it, now that looks good. Okay. So how about we turn this particular thing into background colors as well? So hex and how do we do that? So chalk actually has a property called .Inverse, where you can inverse the colors where the text color would now become the background color. And if we run this particular CLI again, you will see now Twitter is written this way.

How about we add that space thingy again? So it looks a little bit better. Now it is looking good. And how about we remove the colon now, since we don't really need it. And now that looks really good. How about we keep going on and let's add color from GitHub. Okay. For GitHub, I'm going to be using this particular green GitHub color.

And let's just go ahead and copy this thing and paste on top of GitHub. So the color I copied was this one, and this is called GitHub. Let's verfiy this thing is working. Right. And finally, let's add a color for my blog. And for that, I'm probably going to use this particular color. I think let's see if this works good, the purple color and running it again.

I think we can get by by using a little bit of a darker color, but this is not that readable. How about I turn it into something else? Uh, let's see this particular, uh, purple color looks good. I'm going to probably pick it up and use this one here and now it's readable. So let's remove these emojis. And there you go.

Now, our CLI is looking pretty damn good. Uh, there's my name. There's this bio I have. And all of the links. Maybe I can not dim my bio and make it italic or whatnot. Let's see if this works. Oh, nice looking bio there. So I'm using the Operator Mono font. Okay. I don't know about you or I'm pretty happy with the result.

I think we can make use of the same color I used here in the blog for this one as well. Let's see how we can do that. Okay. Okie dokie. So here's my name and what I'm going to do is I'm going to use this particular thing right here. And now let's see how it works. Yeah. I like this more. I like this more. I definitely like this more.

Uh, what else I'm going to do is since we are also using a background color here up top, I'm going to use the same purple color for the welcome text. I think it looks good, but what else we can do is we can totally remove this particular thing. I'm going to go ahead and remove this thing. The heading we have here, and I'm going to add a title which is going to actually act as my name.

And that will be my Ahmad Awais. Now, see, this is actually looking much, much better than I had hoped for. Yep. I'm going to move this back here. So there's not an extra line here. Now if I run the CLI again. Now I'm actually quite satisfied here. I think we have done, we have out done ourself. We have used colors here, and this is the chalk.italic effect.

This is the chalk.dim effect. And these are actual hex colors that we are using in our CLI, which is incredible. A couple more things we can do to improve this is we can actually define a colors inside of variables. For example, instead of defining all of this, how about we try defining a Twitter color, you know, const twitter.

Let's see if it works. I'm going to use Twitter and paste it here. And it still works. We still see that same color, but it is far far better to just use a Twitter constant right here, you know, a twitter variable call it twitterClr. So it's a bit better. You kind of know what is happening here. And then what we are going to do is we are going to add a GitHub color.

githubClr. There you go. And let's call it githubClr color. And finally, this is the purple thing I like. So I'm going to call it purple. const purple. Since purple is already a color, we don't need to call it color. Seems to be working quite nice here. What more can we do. Since we are using it chalk.dim a couple of times here,

how about we go ahead and define a dim color as well. For that I'm going to just say. Uh, const dim is equal to chalk.dim. And now, instead of saying chalk.dim, I'm going to press command + alt + G and replace all of that for dim. But I think it replaced this one. So let's put it back here. Uh, what else?

And then there's italic. How about we go ahead and do the same thing here. Italic and remove chalk. from here. This is looking quite nice. And it is still working the same way that we were hoping for it to work. So let's review what we have done here in a new console log here. You can either type console log, or since I have defined this console log as a log variable, I'm just going to define it as log. Let's go down here and close this thing.

So in log, what you can do is you can actually type chalk. And then the color name for example, green. And then give it the same string, you are going to give console log, for example, this is green. Now if I run awais, you can see that string here. This is green. What else can we do it? We can use .underline.

It will put an underline below it. Think we can add another log here just to create an extra line. Yeah, there you go. As you can see, it is actually underlining that particular string now with the same color we are using here. And now if you go ahead and use inverse, it is just going to inverse the, uh, you know, text colo with the background color.

There you go. You can see that underline as well and the text color and the background color has actually been inverse. Let's take a look at the documentation. So you can do things like these. You can do things like this. If you are building a CLI that reports your CPU RAM and disk space or whatnot. You can actually chalk.red,

green and yellow. For the time being, I'm going to clear everything above here. Clear so we can have a better image and there you go. CPU 90%. Since it is 90%, it is red. And you know, all these colors. We can have all kinds of fun with this kind of library called chalk. You can also go crazy and use RGB if you want to.

And finally, you can also go ahead and create your own themes. For example, an error console log would be a chalk with bold red color or with a warning would be a chalk with, you know, HTML color, uh, that we are used to. How about we go ahead and copy these and try them out. For that I'm going to remove all of this and paste it here.

And if I run the CLI again, you can see, you can see the error with bold red color, and the warning with this particular orange color. And finally, how about we add another one. A success variable. Success. And it is just going to be green because success is green and it is going to shout at us success. Okay.

I'm just going to remove this. And we're going to add another log here for an extra line. Ket;s print them out. There you go. Error, warning, success and whatnot. You can also add blue color. The info alert is kind of always blue, so you can do that by info. And then you can say info. There you go. Some info here, right?

So, this is how you can start improving the user experience of your CLI you know, by using all these different colors. Just don't go super crazy with these colors. And you can also always provide a way for your users to not use colors in your CLI. And that is something we are going to study later in this course.

So finally, I'm going to remove all of this and since I'm happy with it, going to run it and see if this is working all right. Now I'm going to add a new commit called colors. And finally, let's go right ahead and create a new npm version for this particular CLI and publish it. Since this is a minor change, though we have changed colors,

it's a new feature. I'm going to create a new minor release. For that I'm using for that I'm writing npm version minor, and then I'm using hyphen m to actually give it some sort of a message that is better than just the version number. For that I'm going to use release. And this is going to be 2.1.0 release. Now press enter and finally publish this package using npm publish.

There you go. We have published a minor version update of the CLI. If we npm unlink the CLI and try running it using npx, let's see if it is going to run the 2.1.0 of awais CLI. There you go. We are running the 2.1.0 version of awais CLI. There's my name. Howdy. Nice to meet ya! And then some bio here with different colors of whatnot.

And this is all looking really, really good. Go ahead and start adding colors to your terminals, to your CLIs, improve your user and developer experience.