An important part of being

a developer is not just the tools you use,

such as your integrated development environment,

but also the repository of knowledge

and code templates you build up over time.

In this video,

I'm going to introduce you to one of these templates,

which will help you get set up

quickly for building webpages.

Let's open Visual Studio Code to

go through the templates step-by-step.

In the folder, we have index.html

along with the JS folder and a CSS folder.

Inside the JS folder,

there's a file named script.js.

There is no content in this file.

It is added to this folder so that you can

easily add JavaScript code to the webpage.

I recommend that you add

any other JavaScript files to this folder too.

Next, I'll open the CSS folder.

In the folder there's a file named styles.css.

Again, there is no content in this file,

but it is here so that you can easily

add CSS rules to the webpage.

Again, it's best if you add

any other CSS files also to this folder.

I go back to my template folder and

open it in Visual Studio Code.

I open the index.html file.

The file has a semantic HTML document structure set up.

At the top, there is the DOCTYPE declaration

followed by the HTML root element.

Inside the root element,

there are the head and body elements in the head element.

The title and meta tags are

already optimized for search engines.

I recommend that you update the title along

with the description and author meta tags.

Below, are the Meta tags for the Open Graph Protocol.

Don't worry about the details of these tags

just yet as you learn about them in a later video.

For now, the main thing to know is that

if you plan on sharing the webpage on social media,

you should remember to uncomment

this HTML section and update the content attributes.

The next block of commented out code is for icons.

If you want to show an icon in

the web browser tab for your website,

you can update the href

attributes here to point to your icon.

Remember it you always link elements for

the CSS style sheet to

the last section of the head element.

To speed things up for you,

there is a link already set up to reference the

styles.css file in the CSS folder.

By now you can probably see how useful this template is.

Let's continue to the body element.

The body element already contains

a basic semantic structure.

This includes the header element

for your website title or logo,

the nav element for your website navigation,

the main element for your main content,

and the footer element for any copyright notices

or links to secondary webpages.

Below the footer element,

you add script elements for JavaScript files.

To make your life easy,

a script element is already set up to reference the

script.js file in the JS folder.

With this template, you'll be able to

quickly start building out webpages.

It's worth keeping a copy of this template on

your local machine to use

and reference in future projects.

In this video, you've been introduced to

a semantic HTML template that you can

use for building webpages.

Good luck.



Add the CSS file to the css folder

Correct

That's right. You need to add the CSS file to the css folder, the JavaScript file to the jsfolder, a link element to reference the CSS file and a script element to reference the JS file.



Add a link element to reference the CSS file

Correct

That's right. You need to add the CSS file to the css folder, the JavaScript file to the jsfolder, a link element to reference the CSS file and a script element to reference the JS file.



Add the JavaScript file to the js folder

Correct

That's right. You need to add the CSS file to the css folder, the JavaScript file to the jsfolder, a link element to reference the CSS file and a script element to reference the JS file.



Add a link element to reference the JS file



Add a script element to reference the JS file

Correct

That's right. You need to add the CSS file to the css folder, the JavaScript file to the jsfolder, a link element to reference the CSS file and a script element to reference the JS file.