**ViewJS**  
  
The first coarse:  
<https://www.vuemastery.com/courses/intro-to-vue-3/intro-to-vue3/>  
  
**Overview of Vue:**  
**Another good IDE for Vue**  
https://www.dcloud.io/hbuilderx.html?hmsr=vue-en&hmpl=&hmcu=&hmkw=&hmci=  
VueMastery.com website  
https://www.vuemastery.com/courses-path/beginner/  
<https://vuejs.org/guide/introduction.html>  
  
Features/Differences/Project Structure  
Project Structure(s)  
**CDN Based Project Structure**

A picture containing text, screenshot, computer

Description automatically generated  
  
This is created using the CDN version of vue  
<https://vuejs.org/guide/quick-start.html#with-build-tools>  
The example allows you to just include the script link from the official docs in the link above  
The Basic project file structure:  
You have a main.js file and an Index.html file  
The main.js file is where you create your application using Vue.createApp  
The Index.html file makes a reference to the main.js file and mounts the app you create in main.js  
The index.html file is where you make a reference to each component in your application that you build via the import script tag imports  
In the <div = “app”>, this is where you make reference to our imported components.  
**Components:** (CDN Based Project)  
In a CDN based project, you create your components as regular javascript files with the .js extension.  
Withing the component, you a “**Template” section**. The template section contains the markup for it’s component that will be rendered (The HTML). You also have the data method that houses your properties for our component (data). You also have a methods section for your methods that you want to use in your application. You also can have “**computed**” properties. Computed properties are properties that can have more that just a value, they can have “computations” or “operations” associated with them.  
**Tools:**  
VSCode has a “live” server that you can use to view real-time changes to your components as you code. You can also make changes in the console’s browser tools to change properties to test out state changes in realtime.

**CLI Based project** (Minified)  
This project type is created with the vue cli  
<https://vuejs.org/guide/quick-start.html#with-build-tools>  
Graphical user interface, text, application, chat or text message

Description automatically generated  
  
Installing the cli on windows

<https://docs.microsoft.com/en-us/windows/dev-environment/javascript/vue-on-windows>

npm install -g @vue/cli

To create a project

vue create <projectname> A screenshot of a computer

Description automatically generated with medium confidence  
  
This type of project is designed based on SFC (Single File Components) <https://vuejs.org/guide/scaling-up/sfc.html>  
Graphical user interface, text, application, email

Description automatically generated  
  
The project is still mainly the same with some exceptions:  
You have a main.js file (the same as the CDN), this is where you create your app and mount it. (In CDN, you mount in the Index.html file). But this step is done in one file (main.js)  
In the CLI (SFC) approach, you create components much like the way you create components in frameworks like React.js  
The files use the .vue extension.  
The components have a Template section as well as a data() function as well.  
The components are imported as references the way other Javascript frameworks like React and Angular references them. (Standard es6 syntax) for importing  
The export default method is used to make the components “exportable” and “importable”  
**Tools:**  
There are some good Tools as well for the SFC way as well. The Vue Tools for Chrome are very good for analyzing the state and structure and performance of your application.

**Data-Binding**  
Vue uses Data-binding like angular does (Two way data-binding)  
<https://vueschool.io/lessons/user-inputs-vue-devtools-in-vue-3?friend=vuejs>  
This is different from React that uses a Virtual DOM

<https://stackoverflow.com/questions/32712605/is-reacts-data-binding-really-one-way-it-seems-like-it-is-two-way>  
  
**Angular:**

Controller <--> View

**React:**

Component State --> DOM --> New Component State --> New DOM  
  
Documentation:  
View has “EXCELLENT!!!!” documentation. They have a lot of good training videos (paid and free).

Tip: Extensions to Install  
Tip: Using the console to change reactive properties  
Tip: Adding Methods  
Tip: Adding a computed property  
Tip: Creating components  
Tip: Adding props to our re-usable component  
Tip: Emitting events to child components (i:e button events) – event binding  
Tip: Form and Field binding  
Tip: Alerts  
Tip: Importing components and the scope made available to all components  
Tip: The no-brainer deployment  
Tip: Spinning up a live server when coding with the CDN in vs code (as well as other extensions to install)  
**Tip**: VModel (Two way binding), modifiers etc.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* USING THE VIEW CLI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Tip: Creating a view project using the CLI (Seeing the VUE Chrome Extension work as well)  
Tip: CSS <script> tag in App.vue

**Tip**: Another tip on emitting methods  
**Tip**: Passing props (just a refresher)  
**Tip**: Bootstrap for Vue  
**Tip**: Routing with vue

Tip: Bootsrap for vue

<https://bootstrap-vue.org/docs/components/table>  
Everything you can do with bootstrap (tables, elements etc..) is documented in the URL above.

Tip: Extensions to Install  
Live Server  
es6-string-html  
Vetur

Tip: Using the console to change reactive properties  
When you browse your app, you can open developer tools and change a property reactively  
Text

Description automatically generated  
Graphical user interface, text, application

Description automatically generated  
This works with any property on our data object for setting properties using the chrome console  
A screenshot of a computer

Description automatically generated with medium confidence  
  
  
Tip: Adding Methods  
Text

Description automatically generated  
  
Text

Description automatically generated

Tip: Adding a computed property

A screenshot of a computer

Description automatically generated with medium confidence  
A screenshot of a computer

Description automatically generated with medium confidence

Tip: Creating components

A screenshot of a computer

Description automatically generated with medium confidence  
  
Text

Description automatically generated

One thing to note, you have to have the /\*html\*/ directive in order to get the markup not think it is a standard string. This was made possible by using the VS Code Extension:  
Text

Description automatically generated

Text

Description automatically generated  
Above we cut and pasted our markup and code from our main.js and pasted it into our component  
Also note:  
In the main.js file, we still need to have the code:

A screenshot of a computer

Description automatically generated with medium confidence  
This is the default method for vue  
  
And now in our Index.html file  
Text

Description automatically generated  
We import the component via script tag and reference the tag like in angular

Tip: Adding props to our re-usable component  
Text

Description automatically generated  
  
Graphical user interface, text

Description automatically generated  
  
Text

Description automatically generated  
  
Text

Description automatically generated  
  
To check

Graphical user interface, text, application, website

Description automatically generated

Tip: Emitting events to child components (i:e button events) – event binding  
In order to fix the button event, we have to create a event binder using emit  
Over in our component:  
Text

Description automatically generated  
  
In our index.html file:  
Text

Description automatically generated  
  
Then in our main.js file for our methods:  
Graphical user interface, text

Description automatically generated  
it's pretty straight forward

Tip: Alerts  
Text

Description automatically generated  
Graphical user interface, application

Description automatically generated

Tip: Form and field binding:  
Two way binding accomplished as shown below  
A screenshot of a computer screen

Description automatically generated with medium confidence

Tip: Importing components and the scope made available to all components

One interesting thing about imports in Vue:  
You import all of your components in the index.html file only

Text

Description automatically generated  
  
Then in any of your components, you can reference them  
Text

Description automatically generated  
  
Tip: The no-brainer deployment  
Shit is a no-brainer, just copy over the file structure:  
A screenshot of a computer

Description automatically generated with medium confidence  
  
Add application in IIS  
Graphical user interface, text, application

Description automatically generated  
  
And just browse:  
Graphical user interface, application

Description automatically generated

Tip: Spinning up a live server when coding with the CDN in vs code

In VS code add this extension  
A screenshot of a computer

Description automatically generated with medium confidence  
To use the extension, click on the icon at the bottom tool-bar  
Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated  
  
Then right-click on the index.html file:

Graphical user interface, text, application

Description automatically generated  
  
Graphical user interface

Description automatically generated  
  
And as you make changes, they will be reflected in your page immediately. SWEET!!!!  
**Another Extension to install as well**  
A screenshot of a computer

Description automatically generated with medium confidence  
And also don’t forget:  
Text

Description automatically generated

Tip: Creating a view project using the CLI  
This is from the video series  
<https://www.youtube.com/watch?v=GWRvrSqnFbM&list=PL4cUxeGkcC9hYYGbV60Vq3IXYNfDk8At1&index=4>  
To install vue (provided you have node installed)  
Using Powershell (it’s better)  
npm install -g @vue/cli  
Text

Description automatically generated  
  
To create a new project  
vue create modal-project  
Text

Description automatically generated  
A screenshot of a computer

Description automatically generated  
  
Text

Description automatically generated  
  
As shown above, you can run the project with the command:  
npm run serve  
And as you can see, the project is like the one on code-sandbox on my iPAD  
Text

Description automatically generated  
  
A picture containing text

Description automatically generated  
  
Also , the Vue Chrome extension is showing up as well  
Graphical user interface, text, application

Description automatically generated  
  
A screenshot of a computer

Description automatically generated  
  
Tip: CSS <style> tag in App.vue  
One thing to note, in the App.vue file there is a <style> tag, any .css you place in there will be global to all components (see my asana note on code sandbox to see what I am taking about)  
  
This is a good video that explains how to use global styles as well  
<https://www.youtube.com/watch?v=KM1U6DqZf8M&list=PL4cUxeGkcC9hYYGbV60Vq3IXYNfDk8At1&index=5>

**Tip**: Another tip on emitting methods  
If you look below, we are emitting a method that we will name ‘end’ in the parent component. Notice how we are passing data along with it.  
Text

Description automatically generated  
Now over in our parent component:  
Notice below how we are making reference to the emitter by its name (‘end’)  
We are pointing it to the method “endgame”  
In the method, we are specifying an argument as a parameter for the function. When created the emitter, we passed in some data, so what that means is that the method create automatically receives that data as it’s parameter.  
Text

Description automatically generated  
  
**Tip**: Passing props (just a refresher)  
We declare props

A screenshot of a computer

Description automatically generated with medium confidence  
  
Then to receive the prop from the parent:

Text

Description automatically generated  
**Tip**: VModel (Two way binding), modifiers etc.  
<https://vueschool.io/lessons/user-inputs-vue-devtools-in-vue-3?friend=vuejs>  
This is a “VERY” video on vmodel and other features working with model/view in your application  
Text

Description automatically generated  
  
  
**Tip**: Routing with vue  
<https://www.vuemastery.com/blog/vue-router-a-tutorial-for-vue-3/?gclid=Cj0KCQjw5ZSWBhCVARIsALERCvyeRNuUS6qrbYZ3sSelC4Pad_blHQWUgx9pb7i9TcAzH0vIo5aMOcUaArXpEALw_wcB>  
Routing is done the similar to “REACT” (SWEETTTT!!), see the link above for a sample creating of how to install and use routing in Vue.js  
Text

Description automatically generated