

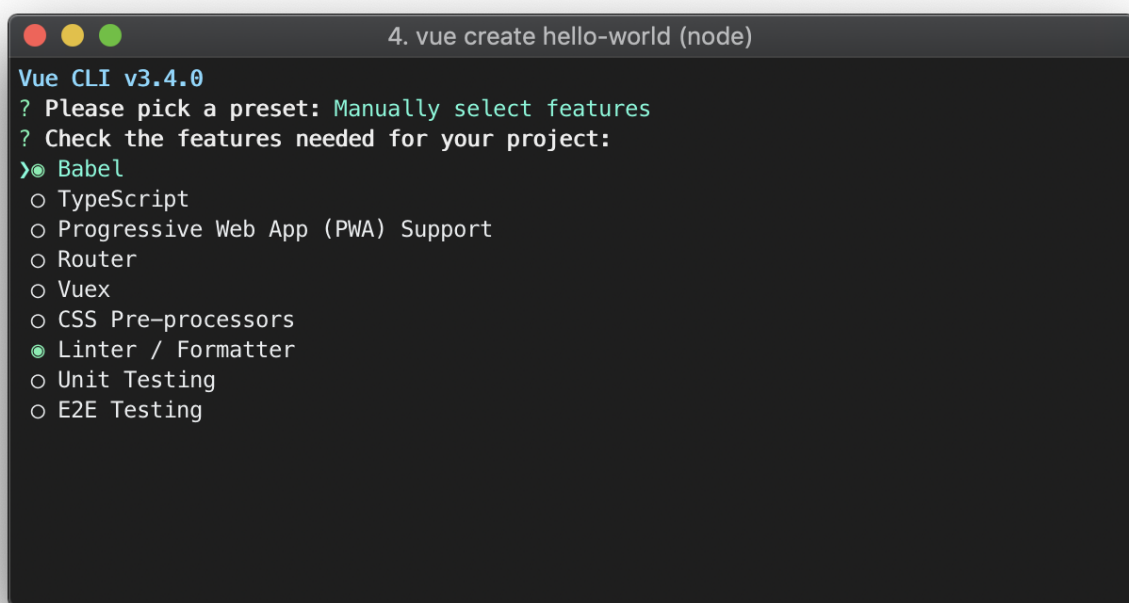
# Tutorial for Introduction to Vue.js

---

In the reading material you were introduced to Vue. In this tutorial you are going to make sure you have all the tools you need to get started. After that you will write your first Vue Component.

## Installing the Vue CLI

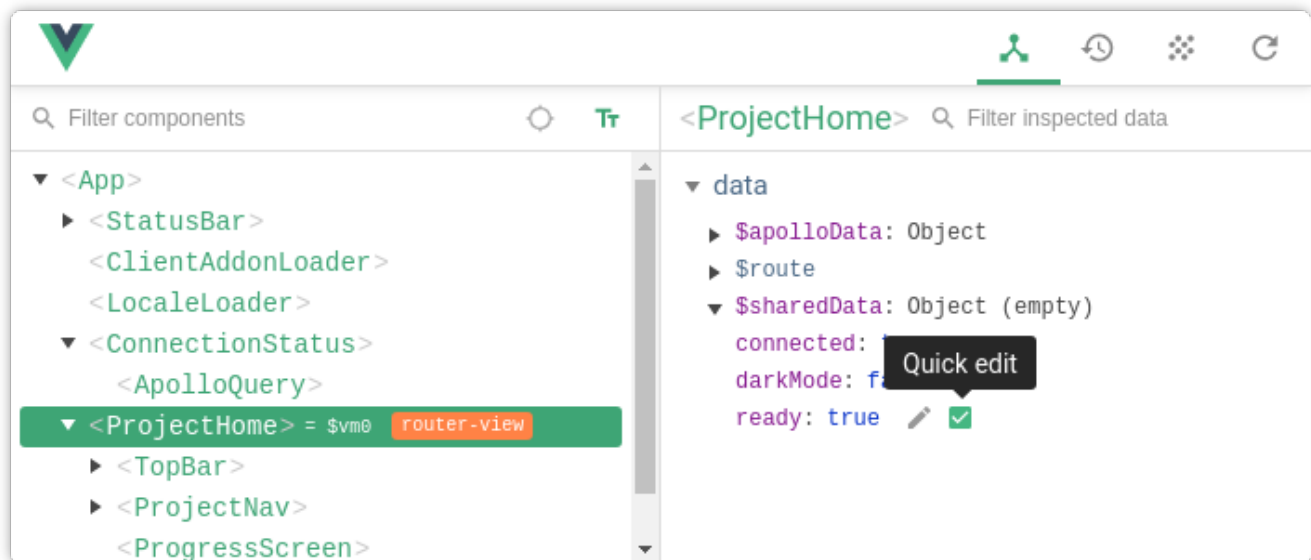
In the reading material you saw how to install the Vue CLI. If you have already done this you can skip this step and move on to Installing Vue DevTools in this tutorial. If you haven't already installed the Vue CLI please go back through the reading material and get this installed. You will want to make sure the Vue CLI is installed and working before lecture.



## Installing Vue DevTools

The Vue DevTools is a Browser extension for Chrome and Firefox that is used to debug Vue.js applications. Please follow the links below and install it for your browser of choice.

- [Get the Chrome Extension](#)
- [Get the Firefox Addon](#)



## Installing Vetur Visual Studio Code Extension

The Vetur Visual Studio Extension gives you a lot of [features](#) for developing Vue applications.

- Syntax-highlighting
- Snippet
- Emmet
- Linting / Error Checking
- Formatting
- IntelliSense
- Debugging

To install the extension go to the extensions marketplace in Visual Studio Code, search for Vetur and install it. With a newer version of VS Code you no longer need to restart VS Code to complete the installation of an extension.

## Building your first component

In this tutorial you are going to build out a simple todo list component. Once you have that component built you will import it into your main application and display it on screen.

You are going to start with a small project that has already been created for you. The project is in a folder called **todo** and it should be located in this directory.

### NPM Install

If you remember from the student book this is an existing project that has some dependencies that aren't stored in source control. To Install your projects dependencies cd into the **todo** folder and run the following command:

```
npm install
```

### TodoList Component

In your components folder create a new file called `TodoList.vue`. A new component is usually made up of 3 pieces;

- HTML `<template></template>`
- JavaScript `<script></script>`
- CSS `<style></style>`

You can copy and paste the following code into your new component or if you have the [Vetur Extension](#) installed you can type out `scaffold` and hit tab to end up with something like this.

```
<template>

</template>

<script>
export default {

}
</script>

<style>

</style>
```

You will start with building the markup for this component. You are going to be building out a simple todo list. You can copy and paste the following code inside of the `<template></template>` tag.

```
<div class="todo-list">
  <h1>My Daily Routine</h1>
  <ul>
    <li>Wake Up</li>
    <li>5 Minute Morning Movement</li>
    <li>Meditate</li>
    <li>Brush Teeth</li>
    <li>Shower</li>
  </ul>
</div>
```

Next you will need to paste the following css code in between the `<style></style>` tag to give your list a little style.

```
.todo-list {
  width:450px;
  background: #fff;
  margin: 50px auto;
  font-family: 'Roboto Condensed', sans-serif;
  border-radius: 10px;
}
h1 {
  background:#f2f2f2;
  color:#4b86A6;
```

```
padding:10px;
font-size:24px;
text-transform: uppercase;
text-align: center;
margin-bottom: 0px;
border-top-left-radius: 10px;
border-top-right-radius: 10px;
}
ul {
  list-style-type: none;
  margin:0px;
  padding:0px;
}
li {
  font-size: 24px;
  border-bottom:1px solid #f2f2f2;
  padding:10px 20px;
}
li:last-child{
  border:0px;
}
```

We will not need the `<script></script>` tag for this example so feel free to just leave it empty.

## App.vue

Now that you have your component built you need to display it on screen. To use it in your application you need to first import it. Open up `App.vue` and in the beginning of your JavaScript section add a new import line to import your new component.

```
import TodoList from './components/TodoList';
```

Then add the component to the components section of the Vue object:

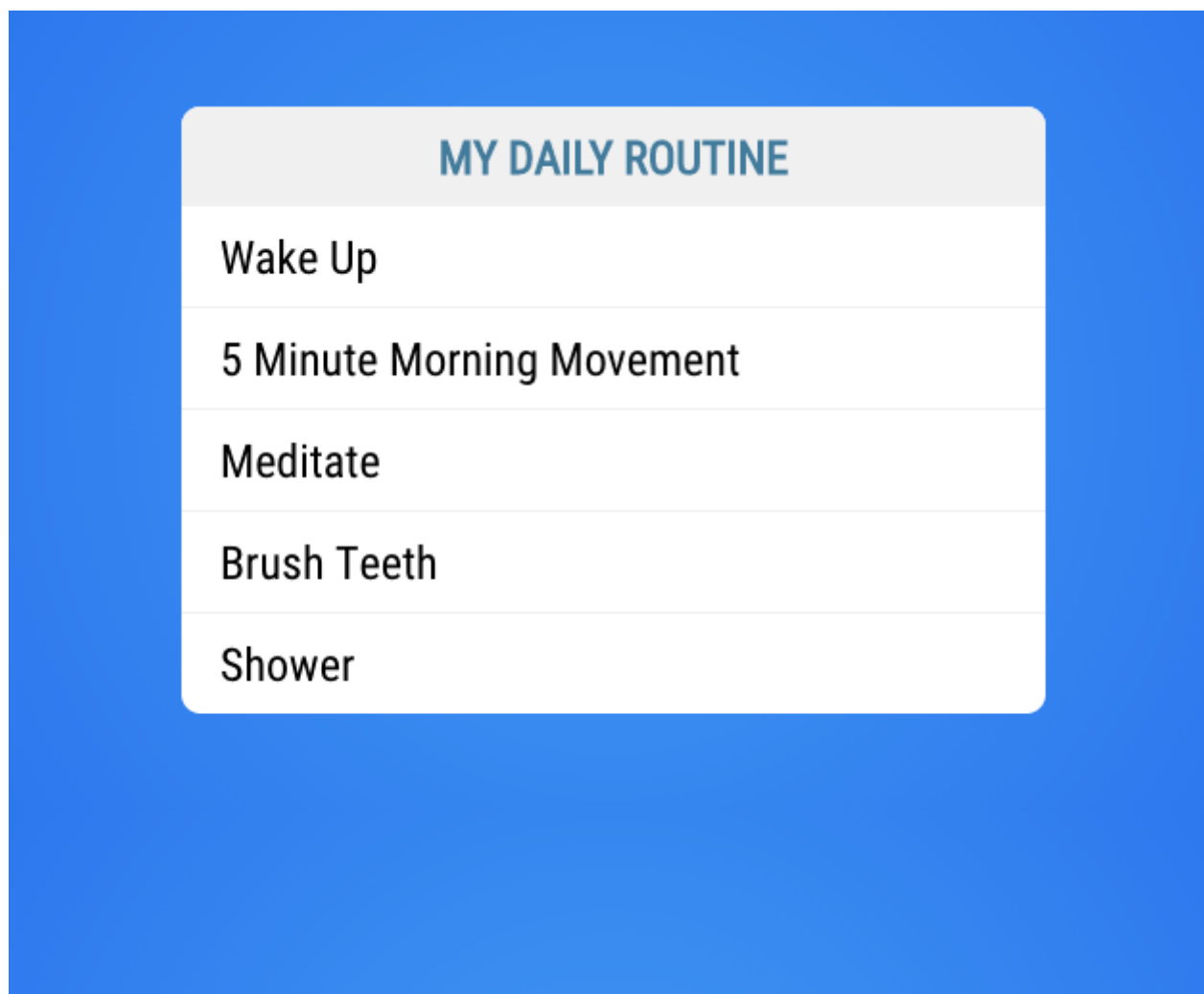
```
components: {
  TodoList
}
```

Now that you have imported the component we can use it in our App. The name of the component is `TodoList`. If you follow the standard all custom made HTML elements should be all lower case and contain a hyphen to separate them from the standard HTML elements. This means that your component will be `<todo-list></todo-list>`

```
<template>
  <div id="todo-app">
    <todo-list></todo-list>
  </div>
</template>
```

## Running your Todo Application

If you run the application using `npm run serve` and visit <http://localhost:8080> you should see something that looks like this.



Congratulations on building a custom component in Vue!