***VUE JS***

**SetUp New Project**

* npm run install –g @vue/cli
* vue create project\_name
* cd project\_name
* npm run serve

vuejs created

If you link the created hook, you can directly execute code as soon as the program starts.

**Receive data from child to props**

Method 1

<AddPost v-on:add-post="addPost" />

 addPost() {

      const newPost = {

        title: this.title,

        body: this.body

      };

      //send to parent

      this.$emit("add-post", newPost);

      (this.title = ""), (this.body = "");

    }

Method 2

Life Cycle Hooks

1. beforeCreated---Vue initialize instance’s reactivity and calls the createdHook
2. created

then it makes some checks to decide if and how to mount the instance in the DOM. When this phase is done it calls the before mount hook

1. beforeMount

then the mounted hook when the template is rendered

1. mounted

before update and update hook runs when the data changes

1. beforeUpdated
2. Update
3. beforeDestroyed
4. destroyed

beforeCreated(){

console.log(‘Before created’, this.list) // Before Created undefined

}

created(){

console.log(‘created’, this.list) // Created [array]}

}

Created hook is great time to call ajax call, so it has some time to load before the component is visible to the user.

beforeMount(){

console.log(‘Before mount, this.list) // before mounted [array]}

}

Once the virtual DOM is updated vue moves on and updated the real DOM

(virtual DOM is the local and simplified copy of real DOM)

Mouted(){

console.log(‘mounted’, this.list, this.$el) // // mounted [array]}

}

Here we have access to extra property. This $el property contains the DOM element

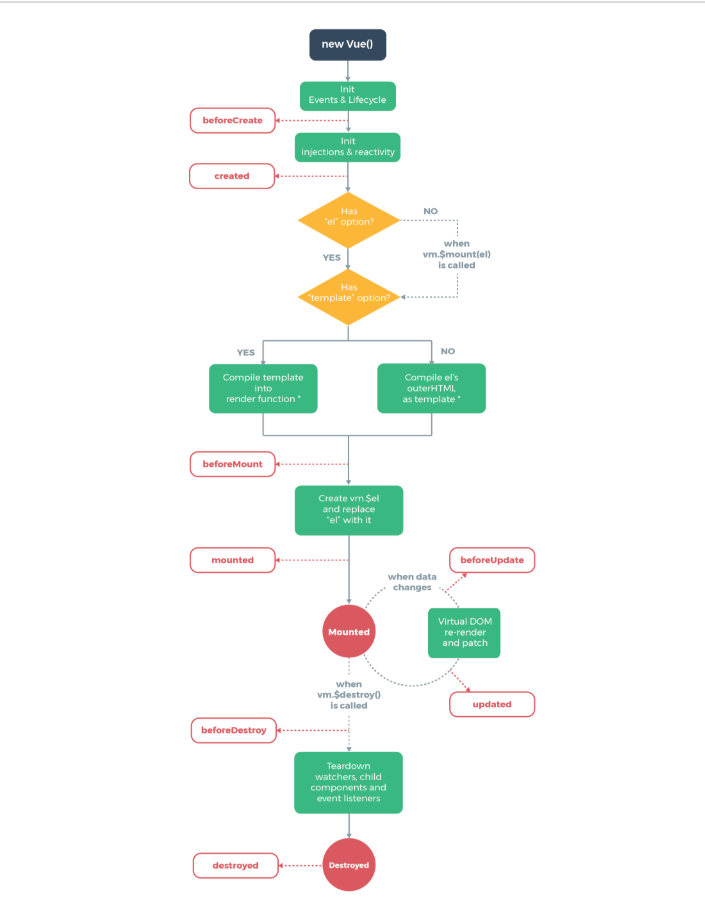
We can use it to access the element’s attribute like this.$el.innerText

beforeDestoyed(){

console.log(‘mounted’, this.list, this.$el) // // mounted [array]}

}

//whenever you want to turn off event listners like firewall database events this is a best place to do it.When a Vue instance is created, it adds all the properties found in its data object to Vue’s **reactivity system**.



For example, the [**created**](https://vuejs.org/v2/api/#created) hook can be used to run code after an instance is created:

new Vue({

data: {

a: 1

},

created: function () {

// `this` points to the vm instance

console.log('a is: ' + this.a)

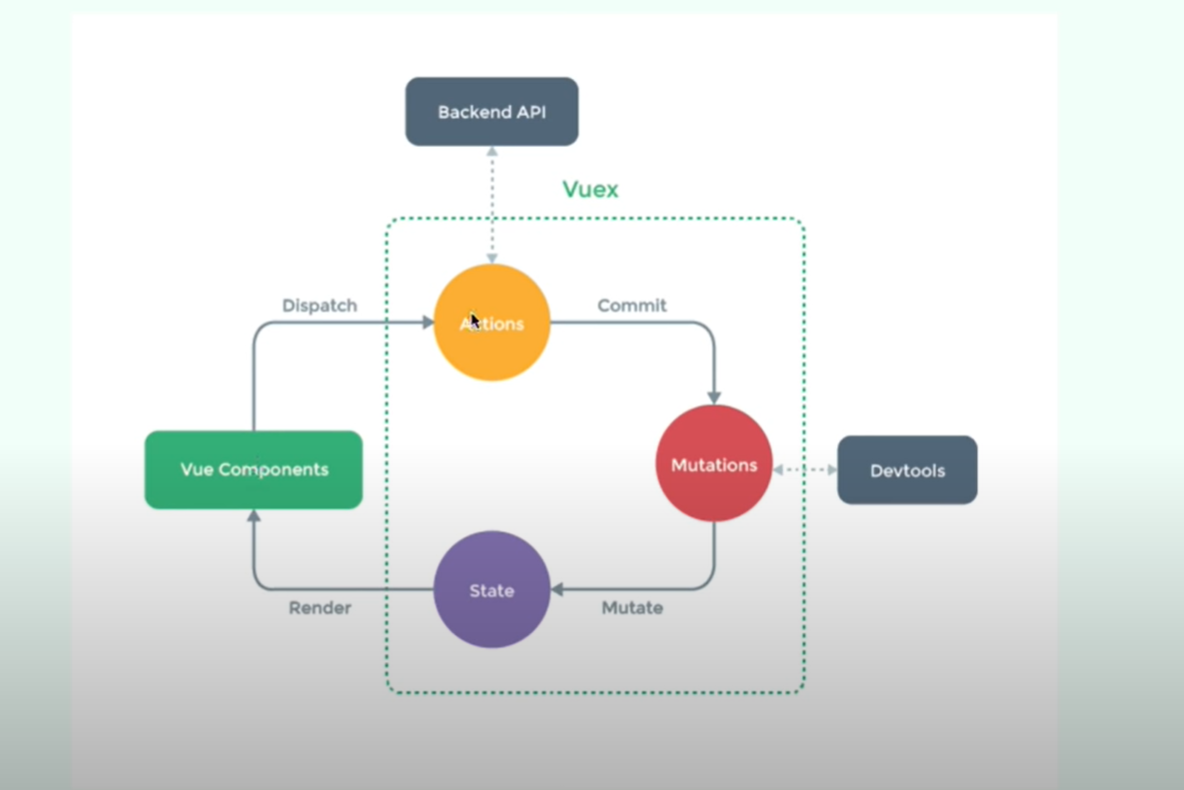
}

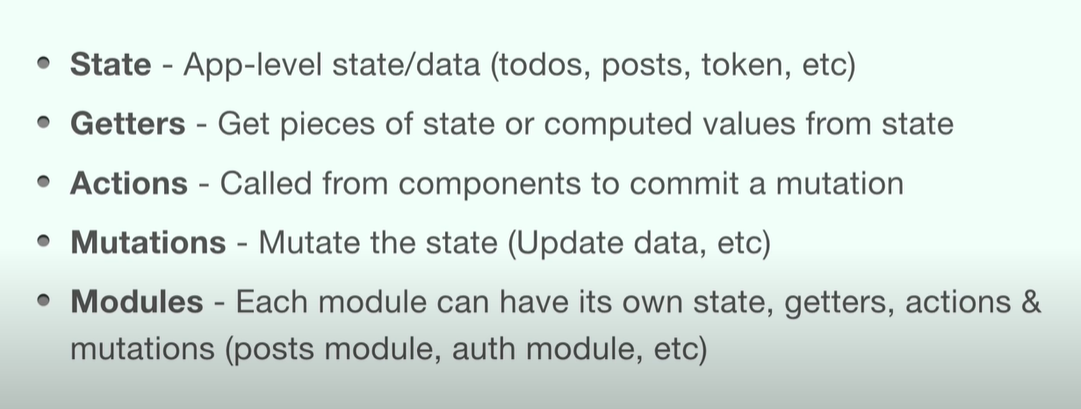
})

// => "a is: 1"

**State Management**

* We use VUEX tool for the state management.





State Management:

1. Make store folder
2. Make index.js file
3. Import this store in main.js file
4. Make module todos in store - >todos.js file
5. In todos.js file make state, getters, actions, mutations
6. Now use these in your Todos.vue file.
7. Import mapGetters in Todos.vue file (mapGetters is similar to connect in react)

* Mapgetters are defined in computed properties
* computed: mapGetters(['allTodos'])

To call actions Import mapActions.

To call the actions before the component loads we use lifecycle function

created() and call the function inside it.