

Javascript Framework

For Front-End

Thavorac

Do you know all these?

Bootstrap

Ember

React

Bulmar

Angular

Vue

Ionic

Jquery

Quasar

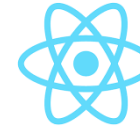
CSS Framework



Javascript Framework



Javascript Library



Hybrid Framework



And a lot more!

VueJs

**The progressive framework
for building user interface**



Vue is a framework and ecosystem that covers most of the common features needed in frontend development.

Vue can be used in different ways:

- Enhancing static HTML without a build step
- Embedding as Web Components on any page
- Single-Page Application (SPA)
- Fullstack / Server-Side-Rendering (SSR)
- JAMStack / Static-Site-Generation (SSG)
- Targeting desktop, mobile, WebGL or even the terminal

Installation

CDN

```
<script src="https://unpkg.com/vue@3"></script>
```

Or with build tool

```
> npm init vue@latest
```

There are other ways to install VueJs but you don't need to worry now

HTML

```
1 <div id="counter">
2   Counter: {{ counter }}
3 </div>
```

html

JS

```
1 const Counter = {
2   data() {
3     return {
4       counter: 0
5     }
6   }
7 }
8
9 Vue.createApp(Counter).mount('#counter')
```

js

CSS

```
.demo {
  font-family: sans-serif;
  border: 1px solid #eee;
  border-radius: 2px;
  padding: 20px 30px;
  margin-top: 1em;
  margin-bottom: 40px;
  user-select: none;
  overflow-x: auto;
}
```

VueJs use the concept of **declarative rendering** to render the UI

2 concepts of **rendering**:

Declarative Rendering

Imperative Rendering

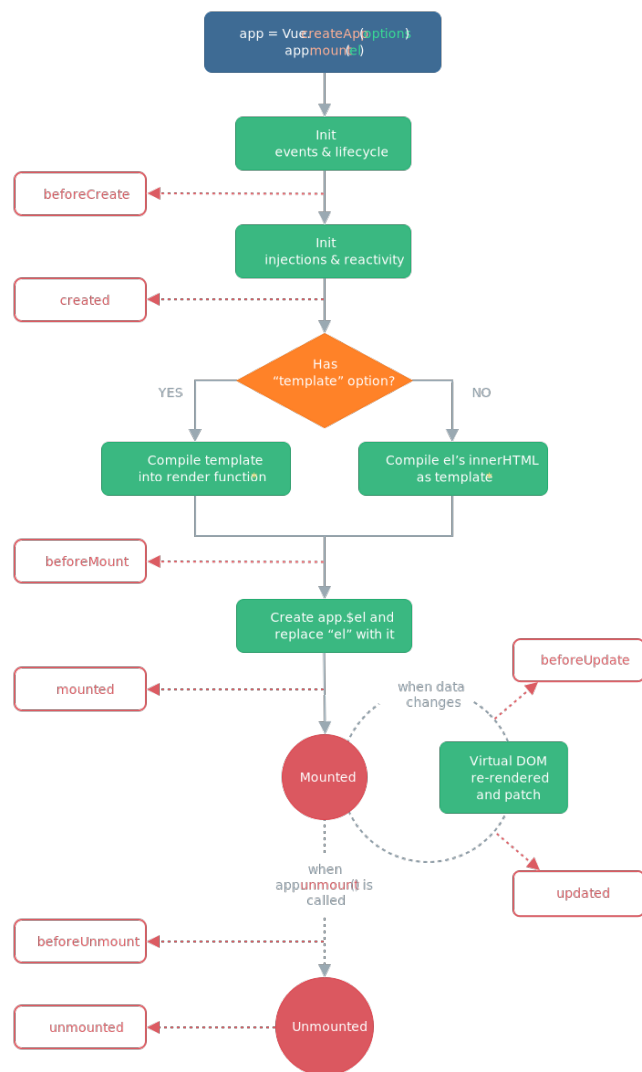
Non-code example:

Mom wants a kid to tidy up its room.

- **Imperative**: “Pick up the toys, then empty the bin, then remove dust and finally vacuum the floor.”
- **Declarative**: “Here’s a picture of how your room looks like clean. I want you to get it into that state in 30 minutes.”



Jquery is imperative rendering type!



* Template compilation is performed ahead-of-time if using a build step, e.g., with single-file components.

Each **Vue instance** goes through a series of lifecycle

the **created** hook can be used to run code after an instance is created

There are also other hooks which will be called at different stages of the instance's lifecycle, such as **mounted**, **updated**, and **destroyed**. All lifecycle hooks are called with their **this** context pointing to the Vue instance invoking it.

Exercise 1:

Create a vue instance and display its cycle name.

Tip: it is a bit tricky to initiate destroy event. Check this out `$destroy()`;

The 3 mostly used concept of Vue

Directive

a directive is some special token in the markup that tells the library to do something to a DOM element

```
1 <div v-text="message"></div>
```

HTML

```
1 <div v-if="Math.random() > 0.5">
2   Now you see me
3 </div>
4 <div v-else>
5   Now you don't
6 </div>
```

html

Method

Methods are likely something you're going to use a lot while working with Vue. They're aptly named as, in essence, we're hanging a function off of an object.

Data

The `data` option for a component is a function. Vue calls this function as part of creating a new component instance. It should return an object, which Vue will then wrap in its reactivity system and store on the component instance as `$data`.

Here is how your vuejs application look like:

```
<script src="https://unpkg.com/vue@3"></script>

<div id="app">{{ message }}</div>

<script>
  Vue.createApp({
    data() {
      return {
        message: 'Hello Vue!'
      }
    }
  }).mount('#app')
</script>
```

html

Practice:

Pull a demo code from:

Repository: `git@github.com:thavorac/gic2021.git`

Branch: `calculator`

Update this code, to allow all button are clickable and display the value on screen.