**Your First Vue.js App**

* Vue can be added by

<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js" defer></script>

* **Or**
  + Yarn install vite
    - Lets us use vue.js with hot reload
  + **Create two files**
    - **Index.js and index.html**
  + **Initalize template with “app” id**
    - **This is were vue renders**

**Text

Description automatically generated**

**Running Vue application (development environment)**

* Yarn vite

**Using Vue.js (hello world)**

**Text

Description automatically generated**

**Adding a template Property**

**Separating js from index.html**

**Text

Description automatically generated**

Template box (like angular)



* **Left:** index.html
* **Right:** index.js

**Using Interaction with methods**

* Options API

Text

Description automatically generated

Data Option

Template option



* + Is a classic way of building vue component
  + Methods layer is where we define how Vue interacts with data

Text

Description automatically generated

Method Option



* Composition API

**Control flow with v-if and v-else**

1. Using logic in template
   * Add condition to `**v-if**`

Text

Description automatically generated

1. Using logic in method
   * Add method to `**v-if**`

Text

Description automatically generated

**Loops with v-for**

* Loop is done using **v-for**
* **V-show** renders components but puts it to **display: none**
  + **V-show is useful when using animation**
* **v-if** doesn’t render the component
  + **This is used in general**

Text

Description automatically generated

**Computed Properties**

* Allows to modify array list properties before being used
  + Useful when creating subset list of list
  + This is cooler than AngularJS

Text

Description automatically generated

Computed properties



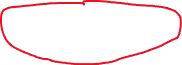
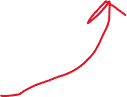
**Class bindings**

* **V-bind:attr\_name**
  + Short form 🡪 :attr\_name
* Is a way of telling to not use string literal but javascript expression

Text

Description automatically generated

Binding



* **V-on** short handed form 🡪 @

Text

Description automatically generated



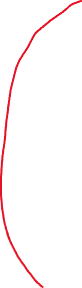
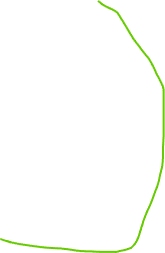
**Input Validation**

* Use **v-bind:attr\_name=”vue\_data”** or **:attr\_name=”vue\_data”** to pick up value of data
* **Use v-on:input=”vue\_function” or @input=”vue\_function”** to register value of data

**Text

Description automatically generatedv**

Registers value



**Amazing v-model**

* Is equivalent to 2-way binding in angular.js
  + Combines both v-bind:input and v-bind:attr\_name

Text

Description automatically generated

The same

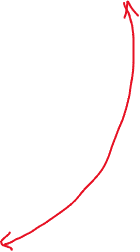


* Can be used with checkbox

Text

Description automatically generated

v-model with checkbox

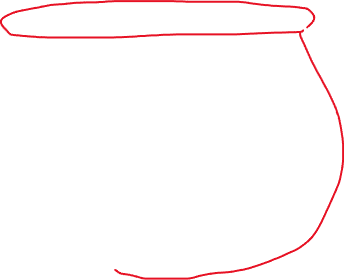


* Can accept array of values for checkbox
  + When uses the same **v-model**

**Text

Description automatically generated** **Graphical user interface, application

Description automatically generated**



**Your First Component**

* Component imported inside `component` section
* Variable 🡪 always strats with uppercase
* Component 🡪 always starts with lowercase

Text

Description automatically generated



Component used here



Component imported here



Created component



**Component Props**

* Component imported inside `component` section
  + Is passed using **v-bind**
  + must be declared which props are coming in before user
* Variable 🡪 always strats with uppercase
* Component 🡪 always starts with lowercase

Text

Description automatically generated

Property used here



Custom component declared here



List of incoming props



Example

Text

Description automatically generated

Reducing unnecessary tags

* Done by injecting v-for directory to component

Text

Description automatically generated

**Child-Parent Communication with Events**

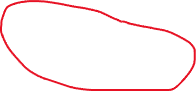
* is done by calling this.$emit(emit\_name, val) in child component

**Example**

**Text

Description automatically generated**

Emit called here



Called here

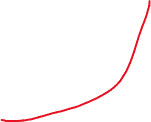
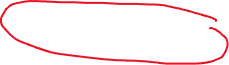


**Text

Description automatically generated**

Called function

Emitted value (from child component)



**Form Validation**