Vue.Js Setup :

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**links :** [**https://www.youtube.com/watch?v=huCmyh58GO0&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=7**](https://www.youtube.com/watch?v=huCmyh58GO0&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=7)

**Start Project Setup & All code in Vue.js >>>**

**# Prerequisites:**

Node.js installed

Vue CLI installed (npm install -g @vue/cli)

**1. Create a Vue.js Project:**

**First, create a new Vue.js project using the Vue CLI:**

# vue create blog

>> ask one question need to install default or any setting ( so now install default )

**after create project in Vue then please go to dir and run server >>>**

# cd blog

**How to serve the server command is >>>**

# npm run serve

when its running you can open any url in your machine ..

App running at:

- Local: <http://localhost:8080/>

- Network: <http://10.24.42.12:8080/>

if you create build then command is >>>

# npm run build

**### folder stucture in Vue.js >>>>**

# README.md ( this file you can mention how to project is runing or create type of docs you can push on git.

# package.json ( this file you can see all project dependency or install after you can check its working or not ).

# package-lock.json ( if this file is delete then project is also running & if package.json removed then you can create again this file via npm install ) / all node module & dependency have in this file so you can check easy to search.

# babel.config.js ( this file used for all lib or framework for every conversion we can use this file only ). # you got only presets command.

# gitignore ( this file need to use which folder or file is not push on git then we can define file ot folder name then its not pushing in git ).

# node\_modules ( this folder have all node modules and dependency of project store in the same )

# public folder ( this folder you got two files only first is index.html file this default execute html file then render all code default its running & favcon.ico this is logo default show in title of image ).

**# html entry file is ( public/index.html )**

# src folder >>>

**# Js entry file is ( src/main.js )**

# components folder >>>

components is special task ( its unique identity for any component index file code )

**# like sidebar , header , topbar , footer these are components.**

**@ likewise**

**App.vue (** this file is used any file component you create after pass the component value import components & use inside file its working **)**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Every file need to name in dobule word like ( Home is wrong & HomePage is right ).

in app.vue file you can use import component & used inside script import and pass to export & use in html inside templete

**code like wise >>>**

**App.vue file >>>**

**inside templete tag use >>>**

<Home data="This is home page props data with components" msg="Data render successfully based on vue Js"/>

**inside script you can use >>>**

**<**script>

// import HelloWorld from './components/HelloWorld.vue'

import Home from './components/Home.vue'

export default {

name: 'App',

components: {

// HelloWorld,

Home

}

}

</script>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Home.vue file >>>**

<template>

<h1>Hello from Home Component</h1>

<h2>{{ data }}</h2>

<h3>{{ msg }}</h3>

<button @click="hello('Hello function called in vue 3')">Call Hello Function On Vue 3</button> <!-- this is syntax for vue 3 -->

<button v-on:click="hello('Hello function called in vue 2')">Call Hello Function On Vue 2</button> <!-- this is syntax for vue 2 -->

</template>

<script>

export default {

name:'HomePage',

props: {

data:String,

msg:String

},

methods:{

hello(anyData){

console.log("hello console called");

alert(anyData)

}

}

};

</script>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

now its working for while you load home page ..

if condition ( conditional statement )

backticks ( while user use html tag render )

backtick means >>>>

<h1 **v-if="show"**>Conditional rendering in vue.js</h1>

v-if="show" is backticks ..

Ex :

<template>

**<h1 v-if="show">Conditional rendering in vue.js if block is showed</h1>**

**<h1 v-else>Conditional rendering in vue.js else block is showed</h1>**

</template>

<script>

export default {

name:'HomePage',

props: {

data:String,

msg:String

},

data(){

**return {show:false};**

}

};

</script>

**# notes ( v-if means vue if statement & else statement also in sentense ) .**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**conditional render | if condition >>>>>**

<template>

<div>

<h3 v-if="show">Conditional rendering in vue.js if block is showed</h3>

<h3 v-else>Conditional rendering in vue.js else block is showed</h3>

<button @click="display()">Toggle button</button>

</div>

</template>

<script>

export default {

name:'HomePage',

props: {

data:String,

msg:String

},

data(){

return {show:true};

},

methods: {

display(){

this.show=!this.show; // reverse if hide then show , if show then hide

}

}

};

</script>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

components >> component have javascript , style & html all code stored.

vue js is composing with component.

composing with component >>>>

**Home.vue file .....**

<template>

<div>

<h1>Home component</h1>

</div>

</template>

<script>

export default {

name: 'HomePage',

};

</script>

<style scoped>

div{

background-color: #dde5ec;

height: 500px;

}

</style>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Update some function string , interger & click me button etc ..

Home.vue file >>>>

<template>

<div>

<h1>Home component</h1>

<h2 v-once>{{ name }}</h2>

<h2>{{ name }}</h2>

<h3>{{ a+b }}</h3>

<button v-on:click="updateText">Click me</button>

</div>

</template>

<script>

export default {

name: 'HomePage',

data(){

return {

name:"Saurabh",

a:10,

b:20,

}

},

methods : {

updateText(){

this.name="Rishabh"

}

}

};

</script>

<style scoped>

div{

background-color: #dde5ec;

height: 500px;

}

</style>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Here is use some backtick function >>>

**# v-once ( this is work as constant value )**

**# v-if ( if condition )**

**# v-else ( else condition )**

**# v-for="item in items" :key "item.id" ( for loop in vue )**

**# v-on:click="functionName" ( onClick button working function )**

**# v-html ( render pure html for web template )**

**# v-bind:disabled="disabledBtn" ( this function do button disabled )**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Ex file >>>>

<template>

<div>

<h1>Home component</h1>

<p v-html="rowHtml"></p>

<h2 v-once>{{ name }}</h2>

<h2>{{ name }}</h2>

<h3>{{ a+b }}</h3>

<h3>{{ ok ? "Yes" : "No" }}</h3>

<button v-on:click="updateText">Click me</button>

<button v-bind:disabled="disabledBtn">Disabled Button</button>

</div>

</template>

<script>

export default {

name: 'HomePage',

data(){

return {

name:"Saurabh",

a:10,

b:20,

ok:true,

rowHtml:"<h2>Html render using function name</h2>",

disabledBtn:true

}

},

methods : {

updateText(){

this.name="Rishabh",

this.ok=!this.ok

}

}

};

</script>

<style scoped>

div{

background-color: #dde5ec;

height: 500px;

}

</style>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Class and Style Bindings >>>>>**

add class in html code and write style inside property code >>>

for style binding we can use the tag inside main >>>

# **v-bind:style="{color:'white'}"**  this one you can pass to bind style inside tag..

-------------------------------------------------------------------------------------------------------------

<template>

<div class="BodyColor">

<h1 **class="homePage"**>Home component</h1>

<p v-html="rowHtml"></p>

<h2 v-once>{{ name }}</h2>

<h2 v-bind:style="{color:'blue'}">{{ name }}</h2>

<h3>{{ a+b }}</h3>

<h3 **v-bind:style="{color:'orange'}"**>{{ ok ? "Yes" : "No" }}</h3>

<div>

<button v-on:click="updateText" class="btnClass1">Click me</button>

<button v-bind:disabled="disabledBtn" class="btnClass2">Disabled Button</button>

</div>

</div>

</template>

<script>

export default {

name: 'HomePage',

data(){

return {

name:"Saurabh",

a:10,

b:20,

ok:true,

rowHtml:"<h2>Html render using function name</h2>",

disabledBtn:true

}

},

methods : {

updateText(){

this.name="Rishabh",

this.ok=!this.ok

}

}

};

</script>

<style scoped>

**.homePage{**

**color:green;**

**}**

**.BodyColor{**

**background-color: yellow;**

**margin: 10px;**

**padding: 10px;**

**}**

**.btnClass1{**

**color:white;**

**background-color:blue;**

**margin:5px;**

**padding: 5px;**

**}**

**.btnClass2{**

**color:yellow;**

**background-color:gray;**

**margin:5px;**

**padding: 5px;**

**}**

</style>

some backtick function used after you can check with inline css & js code based on the same code like wise ...

<template>

<div>

<h1 class="homePage">Home component</h1> <!-- This is class style is below its working -->

<h1 v-bind:style="{color:'red'}"> Style Binding With Object </h1> <!-- This is style is its working like inline css -->

<h1 v-bind:class="{display:show}"> Class Binding With Object </h1> <!-- This is style is its working like inline css based on script code if pass data as true then display otherwise no display css -->

</div>

</template>

<script>

export default {

name: 'HomePage',

data(){

return {

show:true

}

}

};

</script>

<style scoped>

.homePage{

color:green;

}

.display{

color:yellow;

background-color:gray;

}

</style>

based on inline direct apply with JS.

<template>

<div>

<h1 class="homePage">Home component</h1> <!-- This is class style is below its working -->

<h1 v-bind:style="{color:'red'}"> Style Binding With Object </h1> <!-- This is style is its working like inline css -->

**<h1 v-bind:style="{color:colorKey}"> Style Binding With Object Replaced Direct Variable In Script</h1>** <!-- This is style is its working like inline css based on script components-->

<h1 v-bind:class="{display:show}"> Class Binding With Object </h1> <!-- This is style is its working like inline css based on script code if pass data as true then display otherwise no display css -->

<h1 v-bind:class="[aClass]"> Class Binding With Array </h1> <!-- This is style is its working like inline css based on script code if pass data as true then display otherwise no display css while pass array code -->

<h1 v-bind:class="[show ? 'YesShow' : 'NoShow']"> Class Binding With Ternary Condition </h1> <!-- using ternary condition -->

</div>

</template>

<script>

export default {

name: 'HomePage',

data(){

return {

show:true,

aClass:'arrayClass',

**colorKey:"orange",**

}

}

};

</script>

<style scoped>

.homePage{

color:green;

}

.display{

color:yellow;

background-color:gray;

}

.arrayClass {

background-color: cyan;

}

</style>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Data binding in Vue.Js >>>**

**Data binding >>** which data i am write in DOM then its refelect in vue Js & Js code in few seconds. ( view & model sync in one file this is Data binding ) .

databinding I am create one Input box and backtick function used

**v-model="message"**

its is over right or replace input value to text in paragraph , then you can check and verify.

Code Ex >>

**DataBind.vue >>>**

<template>

<div>

<h1>Data Binding in Vue Js</h1>

**<p>{{ message }}</p>**

**<input name="message" type="text" v-model="message">**

</div>

</template>

<script>

export default {

name:'DataBind',

data(){

return {

**message:"Hi Saurabh How are you !!",**

}

}

}

</script>

Note : Html is view part & inside script model part both are combined in same file this is Data-binding .

If you got multiple line read ...

<template>

<div>

<h1>Data Binding in Vue Js</h1>

<!-- for single line read -->

<h5>Single Line Bonding in Vue Js</h5>

<p>{{ message }}</p>

<input name="message" type="text" v-model="message">

<!-- for multiple line read -->

<h5>Multi Line Bonding in Vue Js</h5>

<p style="white-space: pre-line">{{ multiMessage }}</p>

<textarea name="multiMessage" type="text" v-model="multiMessage"></textarea>

</div>

</template>

<script>

export default {

name:'DataBind',

data(){

return {

message:"Hi Saurabh How are you !!",

multiMessage:"Hi How you are",

}

}

}

</script>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++