**Vue Notes**

**Course:** <https://www.udemy.com/course/vuejs-fast-crash-course/>

***Index Html***

<!DOCTYPE html>

<html lang="en">

        <div id="app">

            {{message}}

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                message : "Hello"

            }

        })

    </script>

</html>

* *Directly used vue with the script src=*https://cdn.jsdelivr.net/npm/vue/dist/vue.js
* *Created a new Vue in script tag with element app and a datum message with value Hello. Then in div with tag app, displayed message in curvy brackets {{}}*

***String Interpolation***

<!DOCTYPE html>

<html lang="en">

        <div id="app">

            <h2>{{message}}</h2>

            <h2>{{number + 1}}</h2>

            <h2>{{isActive ? 'is Active' : 'is Not Active'}}</h2>

*<!--if isActive == True then print is Active, else print is Not Active-->*

            <h2>{{message.split('').reverse('').join('')}}</h2>

*<!--Message will be printed in reverse order.*

*Split message to letters H e l l o,*

*Reverse order o l l e H,*

*Join them back olleH -->*

        </div>

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

    <script>

*/\**

*You can change the values in Vue model, vm, through Console in Inspect of the web page.*

*To do that you can type vm.message = "Hallo" for example*

*vm.isActive = false*

*\*/*

        let vm = new Vue({

            el: "#app",

            data: {

                message : "Hello",

                number: 10,

                isActive: false

            }

        })

    </script>

</html>

* *Created new variables, changed them through console, implemented a basic if else statement, split reverse joined a string value.*

***Data Binding***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Data Binding</title>

    </head>

    <body>

        <div id="app">

            <a v-bind:href="url">{{url}}</a>

            <h1 v-bind:data="dynamicId">{{dynamicId}}</h1>

*<!--Use v-bind or : before binding a value from script to have it displayed.-->*

        </div>

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

    <script>

        let vm = new Vue({

            el: "#app",

            data: {

                dynamicId:51,

                url: "https://www.github.com/3rt4nm4n"

            }

        })

    </script>

    </body>

</html>

* *Created a url string value, binded it to href using v-bind. For short of v-bind:href you can type :href as well.*

***Twoway Binding***

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <meta http-equiv="X-UA-Compatible" content="ie=edge">

        <title>Twoway Databinding</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">

    </head>

    <body>

        <div class="container">

            <div id="app">

                <form action="">

                    <div class="col-sm-6">

                        <input v-model="message" type="text" class="form-control">

                    </div>

                </form>

                {{message}}

*<!--Whatever you type in form box the message in previous line will be the same as it's*

*Can be changed through console as well, like vm.message="3rt4nm4n"-->*

            </div>

        </div>

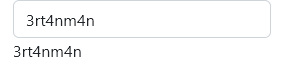
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

    <script>

        let vm = new Vue({

            el: "#app",

            data: {

                message : ""

            }

        })

    </script>

    </body>

</html>

***Twoway Databinding-2***

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <meta http-equiv="X-UA-Compatible" content="ie=edge">

        <title>Twoway Databinding</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">

    </head>

    <body>

        <div class="container">

            <div id="app">

                <form action="">

                    <div class="col-sm-6">

                        <input v-model="options" type="text" class="form-control">

*<!--It shows the options that you checked, separated each with comma.-->*

                    </div>

                </form>

                <fieldset class="form-group">

                    <legend>Radio buttons</legend>

                    <div class="form-check">

                        <label class="form-check-label">

                            <input v-model="options" type="checkbox" class="form-check-input" name="optionsRadios" id="optionsRadio1" value="option-1" checked>

                            Option one

                        </label>

                    </div>

                    <div class="form-check">

                        <label class="form-check-label">

                            <input  v-model="options" type="checkbox" class="form-check-input" name="optionsRadios" id="optionsRadio2" value="option-2">

                            Option two

                        </label>

                    </div>

                    <div class="form-check">

                        <label class="form-check-label">

                            <input  v-model="options" type="checkbox" class="form-check-input" name="optionsRadios" id="optionsRadio3" value="option-3">

                            Option three is disabled

                        </label>

                    </div>

                </fieldset>

                <p> You selected {{options.join(', ')}}</p>

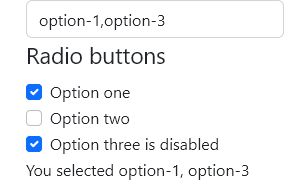
*<!--It shows the options that you checked, separated each with comma.-->*

            </div>

        </div>

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

    <script>

        let vm = new Vue({

            el: "#app",

            data: {

                options: [],

            }

        })

    </script>

    </body>

</html>

***Directives***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Directives</title>

    </head>

    <body>

        <div id="app">

            <a v-bind:href="url">{{url}}</a>

            <h1 v-on:click="number++">{{number}}</h1>

            <h1 @click="number--">Decrease</h1>

            <form action="" @submit.prevent="number++">

*<!--@submit.prevent prevents submit button to refresh the page after clicked.-->*

                <button type="submit">SUBMIT</button>

            </form>

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                number:10,

                url: "https://www.github.com/3rt4nm4n"

            }

        })

    </script>

    </body>

</html>

* *v-on:click and @click are the same.*
* *Submit.prevent was used to stop refreshing the page after clicking submit button*

***Inline Style Binding***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Class Binding</title>

    </head>

    <body>

        <style>

            .active{

                color:green;

            }

            .notActive{

                color:red;

            }

        </style>

        <div id="app">

            <h1 class="static" :class="{active: isActive, 'notActive': !isActive}">{{name}}</h1>

*<!--if isActive is true it will use active class type*

*if isActive is false it will use notActive class type-->*

            <h1 :class="[isActive ? 'active' : 'notActive']">{{name}}</h1>

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                isActive:false,

                name:"Onder"

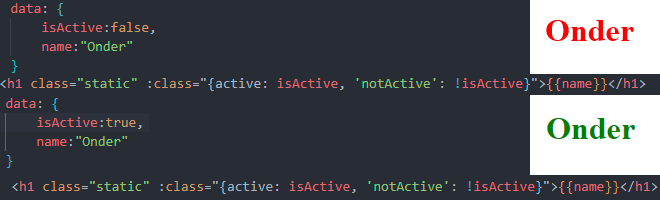
            }

        })

    </script>

    </body>

</html>



***Event Binding***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Inline Style Binding</title>

    </head>

    <body>

        <div id="app">

            <button @click="increaseFont">Increase</button>

            <button @click="decreaseFont">Decrease</button>

           <div :style="{color:'red', fontSize:fontSize+'px'}">

            {{ message }}

           </div>

        </div>

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

    <script>

        let vm = new Vue({

            el: "#app",

            data: {

                message : "Hello",

                fontSize:12,

            },

            methods:{

                increaseFont(){

                        this.fontSize++;

                },

                decreaseFont: function(){

                        this.fontSize--;

                }

            }

        })

    </script>

    </body>

</html>

* *this. is used to access the values in the current Vue app.*
* *When Increase is clicked fontSize will increase by 1 px,* 
  + 
* *When Decrease is clicked fontSize will decrease by 1px.*
  + 

***Conditional Rendering***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Control Structures</title>

*<!--AKA Conditional Rendering-->*

    </head>

    <body>

        <div id="app">

            <h1 v-if="isActive">{{name}}</h1>

*<!--if isActive is false name will not show up, if true it will be shown-->*

            <h1 v-else-if="name==='3rt4nm4n'">Hello Onder</h1>

*<!--if name is 3rt4nm4n it will print Hello Onder-->*

            <h1 v-else>NO USER</h1>

*<!--if isActive is false and name isnt '3rt4nm4n' above line will show-->*

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                isActive:false,

                name : "3rt4nm4n"

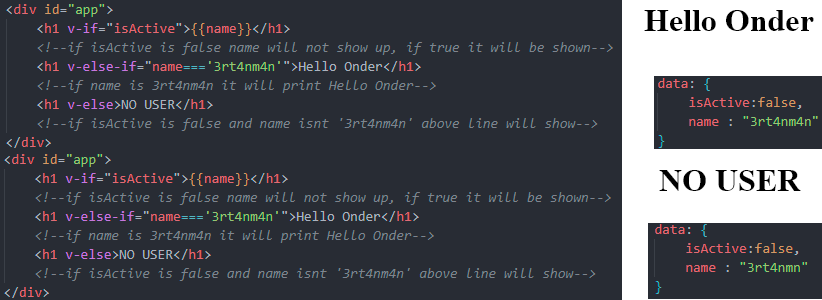
            }

        })

    </script>

    </body>

</html>



***List Rendering***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Control Structures</title>

    </head>

    <body>

        <div id="app">

            <ul>

                <li v-for="post in posts" :key="post.id">{{post.title}}</li>

            </ul>

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                 posts: [

                    {id:1,title:'Post1',body:'dsnfhgrdflkgdf'},

                    {id:2,title:'Post2',body:'dsnfhgrdflkgdf'},

                    {id:3,title:'Post3',body:'dsnfhgrdflkgdf'},

                    {id:4,title:'Post4',body:'dsnfhgrdflkgdf'}

                 ]

            }

        })

    </script>

    </body>

</html>

* *v-for=”post in posts” :key=”post.id”*

***Computed Properties***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Index</title>

    </head>

    <body>

        <div id="app">

            <h2>{{reverseMessage}}</h2>

        </div>

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

    <script>

        new Vue({

            el: "#app",

            data: {

                message : "Hello"

            },

            computed:{

                reverseMessage(){

                    return this.message.split('').reverse('').join('')

                }

            }

        })

    </script>

    </body>

</html>

***Watchers***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Watchers</title>

    </head>

    <body>

        <div id="app">

            <h1>Onder Ertan {{isActive === 'active' ? 'is Active' : 'is not Active'}}</h1>

        </div>

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

    <script>

        let vm = new Vue({

            el: "#app",

            data: {

                isActive: 'not active'

            },

            watch:{

                isActive: function(*newValue*){

                    console.log('Sending an email - '+*newValue*)

                }

*//whenever isActive is 'active' this will be printed in console log.It is watching it.*

            }

        })

    </script>

    </body>

</html>

***Registering Components***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Index</title>

    </head>

    <body>

        <div id="app">

            <posts></posts> *<!--Renders component posts-->*

        </div>

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

    <script>

        Vue.component('posts',{

            template:`<h1>Hello there</h1>`

        })

       let vm = new Vue({

            el: "#app",

            data: {

                message : "Hello"

            }

        })

    </script>

    </body>

</html>

***Inline Template***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Index</title>

    </head>

    <body>

        <div id="app">

            <posts inline-template>

                <h1 @click="display">HELLOss</h1>

            </posts> *<!--whatever you put inside inline template becomes your data*

*when clicked on, youll get console logged-->*

        </div>

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

    <script>

        Vue.component('posts',{

            data(){

                return{

                }

            },

            methods: {

                display(){

                    console.log('hello')

                }

            }

        })

       let vm = new Vue({

            el: "#app"

        })

    </script>

    </body>

</html>

    <body>

        <div id="app">

            <posts data="34">

            </posts>

        </div>

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

    <script>

        Vue.component('posts',{

            props:['data'],

            template: `

                  <h1 @click="display">{{data}}</h1>

            `,

            data(){

                return{

                    message: 'Hello User'

                }

            },

            methods: {

                display(){

                    console.log('hello')

                }

            }

        })

***Nesting Components***

<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Index</title>

    </head>

    <body>

        <div id="app">

            <posts></posts>

        </div>

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

    <script>

        Vue.component('posts',{

            template: `

                  <div>

                      <ul>

                        <post v-for="post in posts" :key="post.id">{{post.title}}</post>

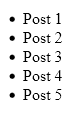
                      </ul>

                  </div>

            `,

            data(){

                return{

                    posts:[

                        {id:1,title:'Post 1'},

                        {id:2,title:'Post 2'},

                        {id:3,title:'Post 3'},

                        {id:4,title:'Post 4'},

                        {id:5,title:'Post 5'},

                    ]

                }

            },

            methods: {

                display(){

                    console.log('hello')

                }

            }

        });

        Vue.component('post',{

            template:`<li><slot></slot></li>`

        })

       let vm = new Vue({

            el: "#app"

        })

    </script>

    </body>

</html>