**Vue JS**

**1st input application in Vue js**

<!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>Vue Js</title>

<script src="vue.js"></script>

</head>

<body>

<div id="app" >

<input type="text" v-on:input="changeTitle">

<p> {{title}} Vue Js</p>

</div>

<script>

new Vue({

el:'#app',

data:{

title:'Hello World'

},

methods:{

changeTitle:function(event){

this.title = event.target.value;

}

}

})

</script>

</body>

</html>

**DOM INTERACTION**

{{}}---string interpulution

Calling function in string interpulution

<p> {{changeTitle()}} Vue Js</p>

new Vue({

el:'#app',

data:{

title:'Hello World'

},

methods:{

changeTitle:function(event){

this.title = event.target.value;

}

}

})

**v-bind:href**

<p> {{changeTitle()}} Vue Js <a v-bind:href="link">Google</a></p>

<script>

new Vue({

el:'#app',

data:{

title:'Hello World',

link:"http://google.com"

},

methods:{

changeTitle:function(){

return this.title;

}

}

})

</script>

### [v-bind Shorthand](https://vuejs.org/v2/guide/syntax.html#v-bind-Shorthand)

<!-- full syntax -->

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

<!-- shorthand -->

<a :href="url"> ... </a>

<!-- shorthand with dynamic argument (2.6.0+) -->

<a :[key]="url"> ... </a>

**v-once ---just one render**

**with out v-once it will re-render the all data**

<h1 v-once>{{title}}</h1>

<p> {{changeTitle()}} Vue Js <a v-bind:href="link">Google</a></p>

<script>

new Vue({

el:'#app',

data:{

title:'Hello World',

link:"http://google.com"

},

methods:{

changeTitle:function(){

this.title="Hello"

return this.title;

}

}

})

</script>

**Hello World**

Hello Vue Js [Google](http://google.com)

**V-html**

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

data:{

title:'Hello World',

link:"http://google.com",

findLink:`<a href="http://google.com">Google</a>`

},

**v-on:click**

<button v-on:click="clickCounter">CLick</button>

<button v-on:click="clickLess"> LessCount</button>

<hr>

<p>{{count}}</p>

data:{

count:0

}

methods:{

clickCounter:function(){

this.count++

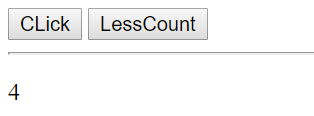
},

clickLess:function(){

this.count--

}

}



**Access for argument and event**

<button v-on:click="clickCounter(2,$event)">CLick</button>

clickCounter:function(step,event){

this.count+= step

},

**v-on:mousemove**

<p v-on:mousemove="updateCoordinates"> Coordinates : {{x}}/{{y}}</p>

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

data:{

x:0,

y:0

},

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

methods:{

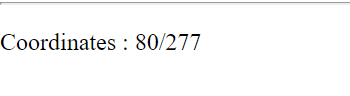
updateCoordinates:function(event){

this.x = event.clientX,

this.y = event.clientY

}

}



<p v-on:mousemove="updateCoordinates">

Coordinates : {{x}}/{{y}}

<span v-on:mousemove.stop="">Dead SPOT</span>

</p>

**Stop the propagation**

**It will stop the mouse hover**

### **[v-on Shorthand](https://vuejs.org/v2/guide/syntax.html" \l "v-on-Shorthand" \o "v-on Shorthand)**

<!-- full syntax -->

<a v-on:click="doSomething"> ... </a>

<!-- shorthand -->

<a @click="doSomething"> ... </a>

<!-- shorthand with dynamic argument (2.6.0+) -->

<a @[event]="doSomething"> ... </a>

**Component**

**Create Component 1st then create date file and html viwer**

Vue.component('todo-item', {

props: ['todo'],

template: '<li>{{ todo.text }}</li>'

})

var app7 = new Vue({

el: '#app-7',

data: {

groceryList: [

{ id: 0, text: 'Vegetables' },

{ id: 1, text: 'Cheese' },

{ id: 2, text: 'Whatever else humans are supposed to eat' }

]

}

})

**-------------------------------------------------------------------------------------------**

<div id="app-7">

<ol>

<!--

Now we provide each todo-item with the todo object

it's representing, so that its content can be dynamic.

We also need to provide each component with a "key",

which will be explained later.

-->

<todo-item

v-for="item in groceryList"

v-bind:todo="item"

v-bind:key="item.id"

></todo-item>

</ol>

</div>