

# Problemes Vue

May 2, 2020

```
[0]: //Ejercicio 1
var vm = new Vue({
  el: "#app",
  data: {
    counter: 0,
  },
  template: '<div>{{counter}}</div>',
});

setInterval(() => vm.counter++, 1000);

//Ejercicio 2
var vm = new Vue({
  el: '#app',
  data: {
    a: 0,
    b: 0,
  },
  methods: {
    add: function() {
      return parseFloat(this.a) + parseFloat(this.b);
    }
  },
  template: `<div>
    <input v-model="a">
    <input v-model="b">
    {{ add() }}
  </div>
`,
  //Nota1: {{ interpolacion }}
  //Nota2: Todo el contenido de template va dentro de 1 elemento raíz!
  //Nota3: v-model -> two way // v-bind -> one way
});
```

```
//Ejercicio 3
var vm = new Vue({
  el: '#app',
  data: {
    visibilidad: true,
  },
  template: `<div>
    <button v-if="visibilidad == true" v-on:click="visibilidad=false"> Click me!
    </button>
  </div>
`,
})
```

```
//Ejercicio 4
var vm = new Vue({
  el: '#app',
  data: {
    texto: "",
  },
  watch: {
    texto: function(){
      if(this.texto.length >= 5){
        this.texto = ""
      }
    }
  },
  template: `<div>
    <input v-model="texto"> {{texto}}
  </div>
`,
})
```

```
//Ejercicio 5
var vm = new Vue({
  el: '#app',
  data: {
    texto: '',
    estilo: '',
  },
  //Nota: v-on:keydown -> apretar tecla, v-on:keyup -> soltar tecla
  template: `<div>
    <input
      v-model="texto"
      v-bind:style="estilo"
      v-on:keydown="estilo = 'background-color: red'"
    >
  </div>
`
```

```

        v-on:keyup="estilo = 'background-color: white'"
      >
    </div>
  },
})

```

*//Ejercicio 6*

```

var vm = new Vue({
  el: '#app',
  data: {
    cantidadRojo: 0,
    estilo: "",
    visibilidad: "",
  },
  watch: {
    cantidadRojo: function() {
      this.estilo = 'color: hsl(0, '+this.cantidadRojo+'%, 50%)'

      if(this.cantidadRojo >= 70){
        this.visibilidad = "true";
      } else {
        this.visibilidad = "false";
      }
    },
  },
  template: `<div>
    <div v-bind:style="estilo"> AM I RED? </div>
    <input v-model="cantidadRojo" type="range" min="1" max="100">
    <div v-if="visibilidad == 'true'"> YES! </div>
  </div>
  `,
})

```

*//FORMA 2 más compacta*

```

var vm = new Vue({
  el: '#app',
  data: {
    cantidadRojo: 0,
  },

  template: `<div>
    <div v-bind:style="'color: hsl(0, '+this.cantidadRojo+'%, 50%)'"> AM I RED?
    ↪</div>
    <input v-model="cantidadRojo" type="range" min="1" max="100">
    <div v-if="cantidadRojo >= 70"> YES! </div>
  </div>
  `
})

```

```

    `
  },
})

//Ejercicio 7
var vm = new Vue({
  el: '#app',
  data: {
    a: false,
    b: false,
    c: false,
    d: false,
  },
  watch: {
    a: function() {
      this.b = this.a;
    },
    b: function() {
      this.c = this.b;
    },
    c: function() {
      this.d = this.c;
    }
  },
  template: `<div>
    <input v-model="a" type=checkbox> {{a}} {{b}} {{c}} {{d}}
  </div>
  `
})

```

```

//Ejercicio 8
/* Hay que añadirlo al index.html
<style>
table { border-collapse: collapse; }
table th,td { border: 1px solid black; }
</style>
*/

var vm = new Vue({
  el: '#app',
  data: {
    personas: [
      { name: 'Jaime Sommers', phone: '311-555-2368' },
      { name: 'Ghostbusters', phone: '55-2368' },
      { name: 'Mr. Plow', phone: '636-555-3226' },
    ]
  }
})

```

```

    { name: 'Gene Parmesan: Private Eye', phone: '555-0113' },
    { name: 'The A-Team', phone: '555-6162' },
  ]
},

template: `<div>
<table>
  <tr>
    <th>Name</th>
    <th>Phone number</th>
  </tr>
  <tr v-for="persona in personas">
    <td>{{persona.name}}</td>
    <td>{{persona.phone}}</td>
  </tr>
</table>
</div>
`,
})

//Ejercicio 9
var vm = new Vue({
  el: '#app',
  data: {
    estado: 0,
  },

  watch: {
    estado: function() {
      if(this.estado == 3){
        this.estado = 0;
      }
    }
  },

  template: `<div>
    <div style="display: inline-block; width:30px;">
    <div v-if="estado == 0 || estado == 1" style="height: 30px;
↪background-color: indianRed"></div>
    <div v-if="estado == 2" style="height: 30px; background-color: red"></div>
    <div v-if="estado == 0 || estado == 2" style="height: 30px;
↪background-color: khaki"></div>
    <div v-if="estado == 1" style="height: 30px; background-color: yellow"></
↪div>
    <div v-if="estado == 1 || estado == 2" style="height: 30px;
↪background-color: seagreen"></div>
    <div v-if="estado == 0" style="height: 30px; background-color: green"></div>
    <button v-on:click="estado++"> Switch! </button>
  `
})

```

```

        </div>
    </div>
    `
  },
})

//Ejercicio 10
var vm = new Vue({
  el: '#app',
  data: {
    personas: [
      { name: 'Jaime Sommers', phone: '311-555-2368' },
      { name: 'Ghostbusters', phone: '55-2368' },
      { name: 'Mr. Plow', phone: '636-555-3226' },
      { name: 'Gene Parmesan: Private Eye', phone: '555-0113' },
      { name: 'The A-Team', phone: '555-6162' },
    ],
  },
  methods: {
    eliminarElemento: function(index) {
      this.personas.splice(index,1);
    }
  },
  template: `<div>
    <table>
      <tr>
        <th>Name</th>
        <th>Phone number</th>
        <th>Delete</th>
      </tr>
      <tr v-for="(persona, index) in personas">
        <td>{{persona.name}}</td>
        <td>{{persona.phone}}</td>
        <td><button v-on:click="eliminarElemento(index)">Delete</button></td>
      </tr>
    </table>
  </div>
  `
})

//Ejercicio 11
Vue.component('words-to-list', {
  data: function(){
    return {
      palabrasSeparadas: "",
    }
  }
})

```

```

    },
    props: ['words'], //Coge el tag/atributo del html
    template: `
    <div>
      <ul v-for="palabra in palabrasSeparadas">
        <li>{{palabra}}</li>
      </ul>
    </div>
    `,
    created: function() { //Se ejecuta al crear una nueva instancia del
    ↪ componente.
      //Prueba que demuestra que es un vinculo reactivo (se modifica el tag desde
      // fuera del propio html). //this.words = this.words += ' hola';
      this.palabrasSeparadas = this.words.split(' ');
    },
  })

var vm = new Vue({
  el: "#app",
  template: `
    <words-to-list words="Lorem ipsum dolor sit amet"></words-to-list>
  `,
})

//Ejercicio 12
Vue.component('card', {
  data: function(){
    return {
      person: {
        name: 'My Name',
        picture: `
        AAAAFcSJAAAADULEQVR42mM82Mz1HwAFqgJP3gasfwAAAABJRU5ErkJggg==`,
        email: 'me@somerandomdomain.com',
        phone: '+00 00 000 0000',
      }
    };
  },
  props: ['src'],
  template: `
    <div class="card">
      <div>
        
      </div>
      <div><h1>{{person.name}}</h1></div>
      <div>{{person.email}}</div>
      <div>{{person.phone}}</div>
    </div>
  `

```

```

    },
  },
})

var vm = new Vue({
  el: "#app",
  template: `
    <div style="display:flex;">
      <card v-bind:personal-data="person"></card>
    </div>
  `,
})

//Ejercicio 13
Vue.component('switch-button', {
  data: function(){
    return {
      encendido: true,
    };
  },
  props: ['state'],
  methods: {
    cambiarEstado: function() {
      if(this.encendido == true){
        this.encendido = false;
        //this.$emit('off') -> lo mismo que el emit en el template
      } else {
        this.encendido = true;
        //this.$emit('on') -> lo mismo que el emit en el template
      }
    },
  },
  //@click -> v-on:click //Same
  template: `
    <div style="border:solid;">
      <button v-bind:disabled="!encendido" @click="$emit('on')">␣
↪@click="cambiarEstado()">ON</button>
      <button v-bind:disabled="encendido" @click="$emit('off')">␣
↪@click="cambiarEstado()">OFF</button>
    </div>
  `,
})

var vm = new Vue({
  el: "#app",
  data: { state: null },
  template: `

```



```

    <div>
      <switch-button
        v-on:on="state='just turned on'"
        v-on:off="state='just turned off'">
      </switch-button>{{state}}
    </div>
  `,
})

//Ejercicio 14
Vue.component('color-selector', {
  data: function(){
    return {
      red: 0,
      green: 0,
      blue: 0,
      hexColor: "#ffffff",
    };
  },
  methods: {
    RGBtoHex: function(valor){
      var hex = parseInt(valor, 10).toString(16);

      if(hex.length < 2) {
        hex = "0" + hex;
      }
      return hex;
    },
    colorEnHex: function() {
      this.hexColor = "#" + this.RGBtoHex(this.red) + this.RGBtoHex(this.green) +
      ↪+ this.RGBtoHex(this.blue);
      return this.hexColor;
    },
  },
  template: `
    <div style="border:solid; display:flex;">
      <div v-bind:style="'background-color:'+this.hexColor+'; width:110px; height:
      ↪110px;'"></div>
      <div style="display:flex; flex-direction:column; padding:10px;">
        <div>R: <input type="range" min=0 max=255 v-model="red" v-on:
        ↪click="$emit('color', colorEnHex())"> red value</div>
        <div>G: <input type="range" min=0 max=255 v-model="green" v-on:
        ↪click="$emit('color', colorEnHex())"> green value</div>
        <div>B: <input type="range" min=0 max=255 v-model="blue" v-on:
        ↪click="$emit('color', colorEnHex())"> blue value</div>
      </div>
    </div>
  `
})

```

```

    `,
  });
  var vm = new Vue({
    el: "#app",
    data: { color: null },
    template: `
    <div style="border:solid red; display:flex;">
      <color-selector v-on:color="color = $event"></color-selector>
      <div v-bind:style="'color:' + color">TEXT</div>
    </div>
    `,
  });

//Ejercicio 15
Vue.component('magic-input', {
  props: ['values'],
  data: function(){
    return {
      magicText: '',
    };
  },
  created: function() {
    this.magicText = this.value.replace(/./g,
      x => x.toUpperCase() == x ? x.toLowerCase() : x.toUpperCase());
  },
  watch: {
    value: function() {
      this.magicText = this.value.replace(/./g,
        x => x.toUpperCase() == x ? x.toLowerCase() : x.toUpperCase());
    },
    magicText: function(){
      this.$emit('input', this.magicText.replace(/./g,
        x => x.toUpperCase() == x ? x.toLowerCase() : x.toUpperCase()));
    }
  },
  template: `
  <input v-model="magicText">
  `,
});

var vm = new Vue({
  el: "#app",
  data: { text: "Prueba" },
  template: `
  <div>

```

```
<magic-input v-model="text"></magic-input>
<input v-model="text">
  {{text}}
</div>
`,
})
```

[0]: