

# Problemes VueJS

Unless otherwise specified, the following setup can be assumed for all exercises

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Page Title</title>
5   <script src="app.js"></script>
6 </head>
7 <body>
8   <div id="app"></div>
9   <script type="text/javascript" src="Pregunta6.js"></script>
10 </body>
11 </html>
```

## Question 1 :

Create web page that counts from 0 to infinity. Create a vue instance vm with:

- a counter property in its data option, which is initially set to 0,
- a template that interpolates the counter, and
- use the following piece of code to increase the counter.
- `setInterval(() => vm.counter++, 100);`

### Solution:

El codi empleat es:

```
1 let options = {
2   data: function(){
3     return {
4       counter : 0,
5     }
6   },
7   created:
8     function(){setInterval(() => this.counter++, 100);
9
10  },
11   template: '<span> {{counter}} </span>',
12 }
13
14 var app = Vue.createApp(options);
```

```
15 const vm = app.mount('#app');
```

### Question 2 :

Create a vue instance that has a template with

- two `<input>` controls bound to variables a and b, respectively, and
- after these two controls, an interpolation of the addition of a and b

#### Solution:

El codi empleat es:

```
1 let options = {
2   data: function(){
3     return {
4       counter : 0,
5     }
6   },
7   created:
8     function(){setInterval(() => this.counter++, 100);
9
10  },
11   template: '<span> {{counter}} </span>',
12 }
13
14 var app = Vue.createApp(options);
15 const vm = app.mount('#app');
```

**Note:** the term bound, as employed here, means that there is a data bindings between the variable and the form control. It may either be an one-way or a two-way binding (with v-bind or v-model, respectively).

Hint: use parseFloat

### Question 3 :

Create a vue instance with a single `<button>` that disappears when clicked.

#### Solution:

El codi empleat es:

```
1 let options = {
2   data: function(){
```

```
3      return {
4          button: 'click me!'
5      }
6  },
7  methods:{
8      del: function(){
9          if(this.button)
10             return this.button = this.button.slice(0,0);
11      }
12  },
13  template: '<div>
14      <button v-if=" button!="' " v-on:click="del()">{{button}}</button
15      >
16  </div>',
17  }
18 var app = Vue.createApp(options);
19 const vm = app.mount('#app');
```

**Question 4 :**

Create a vue instance with a single `<input>` that disappears when clicked.

**Solution:**

El codi empleat es:

```
1 let options = {
2     data: function(){
3         return {
4             flag: true,
5             txta: ''
6         }
7     },
8     watch:{
9         txta: function(){
10             if(this.txta.length > 4)
11                 {
12                     this.txta = '';
13                 }
14         }
15     },
16     template: '<div>
17     <input type="txt" v-model="txta">
```

```
18     </div>`,  
19 }  
20  
21 var app = Vue.createApp(options);  
22 const vm = app.mount('#app');
```

**Question 5 :**

Create a vue instance with a single `<input>` that disappears when clicked.

- Hint: `v-on:keydown`

**Solution:**

El codi empleat es:

```
1 let options = {  
2   data: function(){  
3     return {  
4       classObject:{  
5         'background-color':'white',  
6       }  
7     }  
8   },  
9   methods:{  
10    color: function(){  
11      this.classObject = {  
12        'background-color':'red',  
13      }  
14    },  
15    color2: function(){  
16      this.classObject = {  
17        'background-color':'white',  
18      }  
19    }  
20  },  
21  template: `<div >  
22    <input type='txt' v-bind:style="classObject" v-on:keydown='  
23      color()' v-on:keyup='color2()'>  
24  </div>  
25  `,  
26 }  
27 var app = Vue.createApp(options);  
28 const vm = app.mount('#app');
```

**Question 6 :**

Using the following template, create an instance that changes the ‘redness’ of the **AM I RED?** text according to the value in the range slider. Hide the **YES!** text when redness is under 70

**Solution:**

El codi empleat es:

```
1 let options = {
2   data: function(){
3     return {
4       flag : false,
5       value : 0,
6       colora: 0,
7       classObject: {'color': 'hsl(0,0%,50%)',},
8     }
9   },
10  watch: {
11    value: function(value){
12      console.log(value)
13      this.classObject={
14        'color': 'hsl(0,'+value+'%,50%)',
15      }
16      if(value<70){this.flag = false;}
17      else{this.flag = true;}
18    }
19  },
20  template:
21  `
22    <div>
23      <div v-bind:style="classObject">AM I RED?</div>
24      <input type="range" min="0" max="100" v-model="value">
25      <div v-if="flag">YES!</div>
26    </div>
27  `,
28 }
29 var app = Vue.createApp(options);
30 const vm = app.mount('#app')
```

**Question 7 :**

Create a vue instance with:

- the properties a, b, c and d in its data option (initially set to false), and
- a template with an `<input type=checkbox>` bound to a, followed by the interpolation of the four variables (a, b, c and d).

Create a watch function for the variable a that sets b equal to a. Similarly, create a watch function for the variable b that sets c equal to b, and a watch function for the variable c that sets d equal to c.

**Solution:**

El codi empleat es:

```
1 let options = {
2   template:
3   '
4   <exercice-seven></exercice-seven>
5   '
6 }
7 var app = Vue.createApp(options);
8
9 app.component('exercice-seven',{
10  data:function(){
11    return {
12      a:false,
13      b:false,
14      c:false,
15      d:false
16    }
17  },
18  watch:{
19    a:function(){
20      this.b = this.a
21    },
22    b:function(){
23      this.c = this.b
24    },
25    c:function(){
26      this.d = this.c
27    }
28  },
29  template:
30  '
31  <input type=checkbox v-model="a">{{a}} {{b}} {{c}} {{d}}
32  '
33 })
34 const vm = app.mount('#app')
```

**Question 8 :**

Create a vue instance that displays the following phone book as shown in the accompanying figure.

**Solution:**

El codi empleat es:

```
1 let options = {
2   template:
3   '
4   <phone-book></phone-book>
5   '
6 }
7 var app = Vue.createApp(options);
8
9 app.component('phone-book',{
10  data:function(){
11    return {
12      phone:[
13        { name: 'Jaime Sommers', phone: '311-555-2368' },
14        { name: 'Ghostbusters', phone: '555-2368' },
15        { name: 'Mr. Plow', phone: '636-555-3226' },
16        { name: 'Gene Parmesan: Private Eye', phone:
17          '555-0113' },
18        { name: 'The A-Team', phone: '555-6162' },
19      ],
20      classObject:{
21        'border-collapse' : 'collapse'
22      },
23      classtd:{
24        'border':'1px solid black'
25      }
26    },
27    template:
28    '
29    <table v-bind:style="classObject"><tr v-bind:style="classtd"><th
30      v-bind:style="classtd">Name</th><th v-bind:style="classtd">
31      Phone number</th></tr>
32      <tr v-for="i in phone" v-bind:style="classtd"><td v-bind:
33        style="classtd">{{i.name}}</td><td v-bind:style="classtd
34        ">{{i.phone}}</td></tr>
35    </table>
36    '
37  })
```

```
34 const vm = app.mount('#app')
```

### Question 9 :

Suppose this is a 'sempahore':

Create a web page that:

- renders the semaphore in a vue template,
- has a state variable, which is an integer representing which light is on, and
- has a <button>that switches the semaphore state.

A value of 0 for state denotes a green light, a value of 1 denotes a yellow light, and a value of 2 denotes a red light. The initial state is 0.

Use the following css colors to represent when lights are on: red, yellow, and lawngreen; and use the following to represent when lights are off: indianRed, khaki, and seagreen.

### Solution:

El codi empleat es:

```
1 let options = {
2   template:
3     '
4     <semaphor></semaphor>
5     '
6 }
7 var app = Vue.createApp(options);
8
9 app.component('semaphor',{
10   data:function(){
11     return {
12       integer : 0,
13       classObject1:{'height': '30px',
14         'background-color': 'indianRed'},
15       classObject2:{'height': '30px',
16         'background-color': 'khaki'},
17       classObject3:{'height': '30px',
18         'background-color': 'green'},
19     }
20   },
21   methods:{
22     adder:function(){
23       this.integer = this.integer + 1;
```



```
24     }
25 },
26 watch:
27 {
28     integer: function(){
29         if(this.integer%3==0)
30         {
31             this.classObject1 = {
32                 'height': '30px',
33                 'background-color': 'indianRed'
34             };
35             this.classObject2 = {
36                 'height': '30px',
37                 'background-color': 'khaki'
38             };
39
40             this.classObject3 = {
41                 'height': '30px',
42                 'background-color': 'green'
43             };
44         }
45         if(this.integer%3==1)
46         {
47             this.classObject1 = {
48                 'height': '30px',
49                 'background-color': 'indianRed'
50             };
51             this.classObject2 = {
52                 'height': '30px',
53                 'background-color': 'yellow'
54             };
55
56             this.classObject3 = {
57                 'height': '30px',
58                 'background-color': 'seagreen'
59             };
60         }
61         if(this.integer%3==2)
62         {
63             this.classObject1 = {
64                 'height': '30px',
65                 'background-color': 'red'
66             };
67             this.classObject2 = {
68                 'height': '30px',
```

```
69         'background-color': 'khaki'
70     };
71
72     this.classObject3 = {
73         'height': '30px',
74         'background-color': 'seagreen'
75     };
76 }
77 }
78 },
79 template:
80 '
81 <div style="display: inline-block; width:30px;">
82   <div v-bind:style="classObject1"></div>
83   <div v-bind:style="classObject2"></div>
84   <div v-bind:style="classObject3"></div>
85   <button v-on:click="add()">switch</button>
86 </div>
87 '
88 })
89 const vm = app.mount('#app')
```

**Question 10 :**

Create a vue instance that displays the following phone book as shown in the accompanying figure.

**Solution:**

El codi empleat es:

```
1 let options = {
2   template:
3   '
4     <phone-book></phone-book>
5   '
6 }
7 var app = Vue.createApp(options);
8
9 app.component('phone-book', {
10   data: function() {
11     return {
12       phone: [
13         { name: 'Jaime Sommers', phone: '311-555-2368' },
```

```

14         { name: 'Ghostbusters', phone: '555-2368' },
15         { name: 'Mr. Plow', phone: '636-555-3226' },
16         { name: 'Gene Parmesan: Private Eye', phone:
17             '555-0113' },
18     ],
19     classObject:{
20         'border-collapse' : 'collapse'
21     },
22     classtd:{
23         'border':'1px solid black'
24     }
25 }
26 },
27 methods:{
28     deleter: function(index){
29         this.phone.splice(index, 1)
30     }
31 },
32 template:
33 '
34 <table v-bind:style="classObject">
35 <tr v-bind:style="classtd">
36 <th v-bind:style="classtd">Name</th>
37 <th v-bind:style="classtd">Phone number</th>
38 </tr>
39 <tr v-for="(item, index) in phone" v-bind:style="classtd">
40 <td v-bind:style="classtd">{{item.name}}</td>
41 <td v-bind:style="classtd">{{item.phone}}</td>
42 <button v-bind:style="classtd" v-on:click="deleter(index)">
43     delete </button></tr>
44 </table>
45 '
46 })
47 const vm = app.mount('#app')

```

Hints:

- list.splice,
- v-for="(item,index) in list".