

Note : on remarque que pour parer au comportement naturel de l'événement submit, on passe \$event en argument de addTask

Il faut souligner que \$event est un mot clé de Vue.js.

```
<> index.html x
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width">
6      <script src="vue2.5.17.js"></script>
7      <title>Hello Vue.js</title>
8    </head>
9
10   <body>
11     <div id="app">
12       <h2>{{title}}</h2>
13       <!--le formulaire a la responsabilité de gerer la soumission-->
14       <form v-on:submit="addTask($event)"> <!--IMPORTANT $event est un mots clé dans VUEJS-->
15         <input type="text" placeholder="tâche a faire" v-model="task"><br>
16         <button type="submit" >ajouter</button>
17       </form>
18     </div>
19     <script>
20       console.log(window.a)
21       new Vue({ //création de l'objet Vue, on y a accée grace à la liaison avec le fichier vue.2.5.17.js
22         el: '#app', //propriété "el" qui dit simplement que notre application Vue va prendre le control d'un élément qui a pour id app
23         data: { //propriété "data" permet de lié des données de Vue et de pouvoir les afficher dans la aptie html
24           title: 'A faire',
25           task: '',
26           tasks: [] // création d'un tableau qui prend nos tâches
27         },
28         methods:{
29           addTask :function(event){ //fonction d'ajout de tâche
30             event.preventDefault();// permet quand on submit de ne pas rafraichir la page
31             this.tasks.push(this.task);
32             console.log(this.tasks);
33             this.task = '';
34           }
35         }
36       })
37     </script>
38   </body>
39 </html>
```

Permet de faire la même chose, c'est à dire prévenir le comportement par défaut de l'événement submit mais avec du sucre syntaxique bien plus court.

```
<> index.html x
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width">
6      <script src="vue2.5.17.js"></script>
7      <title>Hello Vue.js</title>
8    </head>
9
10   <body>
11     <div id='app'>
12       <h2>{{title}}</h2>
13       <!--le formulaire a la responsabilité de gerer la soumission-->
14       <form v-on:submit.prevent='addTask()'> <!--IMPORTANT $event est un mots clé dans VUEJS-->
15         <input type='text' placeholder='tâche a faire' v-model='task'><br>
16         <button type="submit" >ajouter</button>
17       </form>
18       <ul>
19         <li v-for='task in tasks'>{{task}}</li> <!-- v-for Nous permet de recrachter les task du tableau tasks-->
20       </ul>
21     </div>
22     <script>
23       console.log(window.a)
24       new Vue({ //création de l'objet Vue, on y a accès grâce à la liaison avec le fichier vue.2.5.17.js
25         el: '#app', //propriété "el" qui dit simplement que notre application Vue va prendre le control d'un élément qui a pour id app
26         data : { //propriété "data" permet de lier des données de Vue et de pouvoir les afficher dans la partie html
27           title: 'A faire',
28           task: '',
29           tasks: [] // création d'un tableau qui prend nos tâches
30         },
31         methods:{
32           addTask :function(){ //fonction d'ajout de tâche
33             this.tasks.push(this.task);
34             console.log(this.tasks);
35             this.task = '';
36           }
37         }
38       })
39     </script>
40   </body>
41 </html>
```

A faire

Elements Console Sources Network Performance Me

top Filter Defau

⚠ The key "width-device-width" is not recognized and ignored.

undefined

You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at <https://vuejs.org/guide/deployment.html>

▶ ["tache 1", __ob__: Observer]

▶ (2) ["tache 1", "tache 2", __ob__: Observer]

▶ (3) ["tache 1", "tache 2", "tache 3", __ob__: Observer]

>