

Création d'un fichier style.css qu'on oublie pas de
lier à notre fichier index.html

index.html

style.css x

```
1  li{
2      list-style-type: none; /* retire les points par défaut des listes*/
3      cursor: pointer; /* Transforme la souris en pointeur a l'approche de la liste */
4      margin-bottom: 10px;
5  }
6
7  .completed{
8      text-decoration: line-through; /*Permet de rayer la tache*/
9      color: green;
10 }
```

v-bind:class permet de lier notre class css à l'élément courant de la liste on lui dit "tu appliques la class css completed quand task.done c'est à dire quand elle est à true.

v-on:click permet de réagir à un événement click, on lui dit "Quand il y a un événement click task.done est égale au contraire de la valeur du task.done courant"

```
<?index.html x # style.css
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     <link rel="stylesheet" href="/style.css">
8     <title>Hello Vue.js</title>
9   </head>
10
11   <body>
12     <div id="app">
13       <h2>{{title}}</h2>
14       <!-- le formulaire a la responsabilité de gérer la soumission -->
15       <form v-on:submit.prevent="addTask()"> <!-- IMPORTANT: prevent est un mot clé dans VUEJS -->
16         <input type="text" placeholder="tâche à faire" v-model="task"><br>
17         <button type="submit">ajouter</button>
18       </form>
19       <ul>
20         <li
21           v-bind:class="{completed: task.done}"
22           v-on:click="task.done = !task.done"
23           v-for="task in tasks">{{task.description}}</li> <!-- v-for nous permet de récupérer les task du tableau tasks -->
24         </li>
25       </ul>
26     </div>
27     <script>
28       const vm = new Vue({ // création de l'objet Vue, on y a accès grâce à la liaison avec le fichier vue.2.5.17.js
29         el: '#app', // propriété "el" qui dit simplement que notre application Vue va prendre le contrôle d'un élément qui a pour id app
30         data: { // propriété "data" permet de lier des données de Vue et de pouvoir les afficher dans la partie html
31           title: 'À faire',
32           task: null,
33           tasks: [] // création d'un tableau qui prend nos tâches
34         },
35         methods: {
36           addTask: function() { // fonction d'ajout de tâche
37             this.tasks.push({
38               id: Date.now(),
39               description: this.task,
40               done: false
41             });
42             console.log(this.tasks);
43             this.task = null;
44           }
45         }
46       });
47     </script>
48   </body>
49 </html>
```

A faire

tâche a faire

ajouter

~~tâche 1~~

tâche 2

tâche 3



Elements

Console

Sources

Network

Performance

Memory

Applic



Ready. Detected Vue 2.5.17.



Filter components

Select Format

<Root>



Filter inspected data

<Root> = \$vm0

▼ data

task: null

▼ tasks: Array[3]

▼ 0: Object

description: "tâche 1"

done: true

id: 1533763656148

► 1: Object

► 2: Object

title: "A faire"