推荐大家安装的 VScode 中的 Vue 插件

1. Vue 3 Snippets https://marketplace.visualstudio.com/items?itemName=hollowtree.vue-snippets
2. Vetur https://marketplace.visualstudio.com/items?itemName=octref.vetur

什么是 vue

1. 构建用户界面
    - 用 vue 往 html 页面中填充数据，非常的方便
2. 框架
    - 框架是一套现成的解决方案，程序员只能遵守框架的规范，去编写自己的业务功能！
    - 要学习 vue，就是在学习 vue 框架中规定的用法！
    - vue 的指令、组件（是对 UI 结构的复用）、路由、Vuex、vue 组件库
    - 只有把上面老师罗列的内容掌握以后，才有开发 vue 项目的能力！

vue 的两个特性

1. 数据驱动视图：
    - 数据的变化会驱动视图自动更新
    - 好处：程序员只管把数据维护好，那么页面结构会被 vue 自动渲染出来！
2. 双向数据绑定：

> 在网页中，*form* 表单负责采集数据，*Ajax* 负责提交数据。

    - js 数据的变化，会被自动渲染到页面上
    - 页面上表单采集的数据发生变化的时候，会被 vue 自动获取到，并更新到 js 数据中

> 注意：数据驱动视图和双向数据绑定的底层原理是 *MVVM*（*Mode* 数据源、*View* 视图、*ViewModel* 就是 *vue* 的实例）

vue 指令

1. 内容渲染指令

1. `v-text` 指令的缺点：会覆盖元素内部原有的内容！(用的不多)。所以要想不被覆盖则需要用到{{ }}插值表达式。
2. `{{ }}` 插值表达式：在实际开发中用的最多，只是内容的占位符，不会覆盖原有的内容！插值表达式不能用在属性节点，只能用在内容节点(用的比较多)。

   因为**v-text**和**{{ }}**会把标签也会直接当作字符串处理，只渲染纯文本内容，所以就出现**v-html**来渲染**html**元素。
3. `v-html` 指令的作用：可以把带有标签的字符串，渲染成真正的 HTML 内容！

2. 属性绑定指令

> 注意：*{{ }}*插值表达式只能用在元素的内容节点中，不能用在元素的属性节点中！

- 在 vue 中，可以使用 `v-bind:` 指令，为元素的属性动态绑定值；
- 简写是英文的 `:`
- 在使用 v-bind 属性绑定期间，如果绑定内容需要进行动态拼接，则字符串的外面应该包裹单引号，例如：

```
<div :title="'box' + index">这是一个 div</div>
```

3. 事件绑定

1. `v-on:` 简写是 `@`
2. 语法格式为：

```
<button @click="add"></button>

methods: {
   add() {
            // 如果在方法中要修改 data 中的数据，可以通过 this 访问到
            this.count += 1
   }
}
```

3. `$event` 的应用场景：如果默认的事件对象 e 被覆盖了，则可以手动传递一个 $event。例如：

```
<button @click="add(3, $event)"></button>

methods: {
   add(n, e) {
            // 如果在方法中要修改 data 中的数据，可以通过 this 访问到
            this.count += 1
   }
}
```

4. 事件修饰符：

- `.prevent`

```
<a @click.prevent="xxx">链接</a>
```

- `.stop`

```
<button @click.stop="xxx">按钮</button>
```

4. v-model 指令

1. input 输入框

- type="radio"

- type="checkbox"
- type="xxxx"

2. textarea
3. select

5. 条件渲染指令

1. `v-show` 的原理是：动态为元素添加或移除 `display: none` 样式，来实现元素的显示和隐藏
    - 如果要频繁的切换元素的显示状态，用 v-show 性能会更好
2. `v-if` 的原理是：每次动态创建或移除元素，实现元素的显示和隐藏
    - 如果刚进入页面的时候，某些元素默认不需要被展示，而且后期这个元素很可能也不需要被展示出来，此时 v-if 性能更好

> 在实际开发中，绝大多数情况，不用考虑性能问题，直接使用 *v-if* 就好了！！！

v-if 指令在使用的时候，有两种方式：

1. 直接给定一个布尔值 true 或 false

```
<p v-if="true">被 v-if 控制的元素</p>
```

2. 给 v-if 提供一个判断条件，根据判断的结果是 true 或 false，来控制元素的显示和隐藏
3.

```
<p v-if="type === 'A'">良好</p>
```