

## 02.渲染和挂载

学习要点：

- 1.渲染挂载
- 2.搜索方式

本节课我们来开始了解 Vue3.x 创建使用的声明渲染，了解数据的动态改变。

### 一.渲染挂载

1. 前端框架基本都是为了简化：模版渲染、事件绑定和用户交互问题；
2. Vue 的 MVVM 模型：即视图(View)-视图模型(ViewModel)-模型(Model)；
3. Vue 的操作核心：即使用模版语法声明式的将数据渲染进 DOM 中去；
4. 比如，创建一个 DOM 结构，内部需要一个动态变化数据使用 Vue 处理；

```
<div id="app">  
  计数器：0                                // 这里的 0 需要动态改变  
</div>
```

5. 那么，首先引入 Vue.js，然后在下方编写需要的 JS 代码；

```
<script src="../../vue.global.js"></script>
```

6. 其次，在引入 Vue 的下面编写一个对象，用于实现 0 的动态改变；

```
// 声明一个选项对象  
const App = {  
  // 初始数据  
  data() {  
    return {  
      counter : 100  
    }  
  }  
}
```

PS：选项对象内部包含初始数据，方法、事件、声明周期等等要素；

7. 这个选项对象 App，是需要当作参数传递到 Vue 的全局 API：createApp()；
8. createApp()方法会返回应用实例，再通过这个实例来挂载对应的 DOM 即可；

```
// 全局 API 对象.创建实例(App).挂载('#app')  
Vue.createApp(App).mount('#app')
```

PS：如果想查阅这个全局 API，更多的文档了解，选择 API 参考：

<https://vue3js.cn/docs/zh/api/global-api.html#createapp>

9. 最后一步：需要通过 Vue 改变的 Counter 值在 DOM 中渲染出来；

```
<div id="app">  
  计数器：{{counter}}  
</div>
```

PS: {{插值}}，这里用双大括号表示，插值对应的是 data() 里的 counter 属性；

## 二. 搜索方式

1. 一般来说，课程中只会拎出重点来阐述，建议学习完毕后再过一遍文档补全；
2. 对于盲区部分，比如 MVVM 是啥？前期其实不用理解，学完后会后能理解七七八八；
3. 强迫症，可以去搜索 MVVM 了解下即可；但学完后，回来再理解效果更好；
4. Vue2.x 中使用 data : {} 的方式，现在是 data() { return }，怎么补全理解？
5. 可通过搜索：vue data return 三个关键字看文档理解；