

## 03.Composition API

学习要点：

### 1.组合式 API 简介

本节课我们来了解一下 Vue3.x 中的组合 API 的简单介绍。

#### 一. 组合式 API

1. 由于 options(选项)配置方案，即：data、computed、methods、watch 等；
  - (1) . 这种方案在组件不是很大的情况下，可读性和维护性还尚可；
  - (2) . 但在不断扩展组件的逻辑时，就会导致组件难以理解和维护；
  - (3) . 我们需要把相同的逻辑点组织在一起，从而提高维护性和阅读性；
2. 为了解决这些问题，Vue3.x 提供了 Composition API，即：组合式 API；
3. Composition API 使用 setup()入口函数，在这个函数里实现 options；

```
import { ref,computed } from 'vue'

export default {
  name: "About",

  //入口
  setup() {
    // 响应式变量，类似 data 设置
    const count = ref(0)
    // 方法设置，类似 methods 设置
    const increment = () => {
      count.value++
    }
    // 计算属性，类似 computed 设置
    const version = computed(() => 'v1.0')

    return {
      count,
      increment,
      version
    }
  }
}
```

PS：我们将同一个逻辑点放在一起，提高了可读性，那如何提高复用性呢？

4. 我们可以在 `script` 区域，就是在 `export default` 外封装复用函数；

// 这里方法是逻辑点 1，也可以分离到单独文件，提高复用性；

```
const test = () => {  
    ...  
}
```

PS: 这里的...就是上面 `setup()` 里面的全部代码；

5. 空出来的 `setup()` 调用这个逻辑点 1 的复用函数，具体如下：

//入口

```
setup() {  
    const { count, increment, version } = test()    //逻辑 1  
    return {  
        count,  
        increment,  
        version  
    }  
}
```