

## 过滤器

### 过滤器的注意点

1. 要定义到 **filters** 节点下，本质是一个函数
2. 在过滤器函数中，一定要有 **return** 值
3. 在过滤器的形参中，可以获取到“管道符”前面待处理的那个值
4. 如果全局过滤器和私有过滤器名字一致，此时按照“就近原则”，调用的是”私有过滤器“

## watch 侦听器

### 侦听器的格式

1. 方法格式的侦听器
  - 缺点1：无法在刚进入页面的时候，自动触发！！！！
  - 缺点2：如果侦听的是一个对象，如果对象中的属性发生了变化，不会触发侦听器！！！！
2. 对象格式的侦听器
  - 好处1：可以通过 **immediate** 选项，让侦听器自动触发！！！！
  - 好处2：可以通过 **deep** 选项，让侦听器深度监听对象中每个属性的变化！！！！

## 计算属性

特点：

1. 定义的时候，要被定义为“方法”

2. 在使用计算属性的时候，当普通的属性使用即可

好处：

1. 实现了代码的复用
2. 只要计算属性中依赖的数据源变化了，则计算属性会自动重新求值！

## axios

*axios* 是一个专注于网络请求的库！

### axios 的基本使用

1. 发起 GET 请求：

```
axios({
  // 请求方式
  method: 'GET',
  // 请求的地址
  url: 'http://www.liulongbin.top:3006/api/getbooks',
  // URL 中的查询参数
  params: {
    id: 1
  }
}).then(function (result) {
  console.log(result)
})
```

2. 发起 POST 请求：

```
document.querySelector('#btnPost').addEventListener('click',
  async function () {
    // 如果调用某个方法的返回值是 Promise 实例，则前面可以添加 await！
    // await 只能用在被 async “修饰”的方法中
    const { data: res } = await axios({
      method: 'POST',
      url: 'http://www.liulongbin.top:3006/api/post',
```

```
    data: {  
      name: 'zs',  
      age: 20  
    }  
  })  
  
  console.log(res)  
})
```

## vue-cli 的使用

1. 在终端下运行如下的命令，创建指定名称的项目：

```
vue create 项目的名称
```

2. vue 项目中 src 目录的构成：

**assets** 文件夹：存放项目中用到的静态资源文件，例如：css 样式表、图片资源

**components** 文件夹：程序员封装的、可复用的组件，都要放到 **components** 目录下

**main.js** 是项目的入口文件。整个项目的运行，要先执行 **main.js**

**App.vue** 是项目的根组件。