

## mixin 抽离公共逻辑

如果多个组件之间存在一些公共的逻辑，则可以抽离这些公共的逻辑，写在一个 js 文件中，在组件中通过 `mixins: ['myMixin']` 方式引入，一个组件可以引入多个 `mixin`。

**数据对象在内部会进行递归合并，数据发生冲突时，以组件数据优先。**

**同名钩子函数将合并为一个数组，都将被调用，混入对象的钩子将在组件自身钩子之前调用。**

```
<!-- index.vue -->
<template>
  <div>
    <div>姓名: {{ name }}</div>
    <div>性别: {{ sex }}</div>
    <div>年领: {{ age }}</div>
  </div>
</template>

<script>
import common from './common.js';

export default {
  name: 'MixinComponent',
  mixins: [common],
  components: {
  },
  data() {
    return {
      name: 'lisi',
      sex: 'male'
    }
  },
  methods: {
    getSex() {
      console.log("sex", this.sex);
    }
  },
  created() {
    this.getAge();
    this.getSex();
  },
  mounted() {
    console.log("index.vue", this.sex);
  }
}
</script>
```

```
// common.js
export default {
  data() {
    return {
      age: 18,
    }
  },
  methods: {
    getAge() {
      console.log("age", this.age);
    }
  },
  mounted() {
    console.log("common.js", this.age);
  }
}
```

在 `index.vue` 中，通过 `mixins: [common]` 将 `common.js` 中的内容混入（融入）`index.vue`。就实现了公共逻辑抽离的目标。

这样让 `index.vue` 在自身组件的基础上，增加了变量、函数。但 `mixin` 也有一些问题：

- 变量来源不明确，不利于阅读；
- 一个组件中引入多个 `mixin`，可能会造成命名冲突；
- `mixin` 和组件可能会造成多对多的关系，复杂度较高。