```
$ hexo n vue-skill reuse
```

前言

关于这些东西，相似又不太一样的，我们一起来讲讲。

- mixin(混入)
- plugin(插件)
- extend(扩展)

<!-- more -->

mixin

> 混入 (mixin) 提供了一种非常灵活的方式，来分发 Vue 组件中的可复用功能。一个混入对象可以包含任意组件选项。当组件使用混入对象时，所有混入对象的选项将被“混合”进入该组件本身的选项。

简单说这东西就是针对vue**组件重复的选项**可以提取为一个mxin，然后给不同的组件引入实现复用。

比如官网的例子，（官网的例子用了Vue.extend，我们暂时不用，待会再讲）， 定义一个混入对象，可以给多个组件使用。

```
// 定义一个混入对象
var myMixin = {
  created: function () {
    this.hello()
  },
  methods: {
    hello: function () {
      console.log('hello from' + this.$options.name+ '\'s mixin!')
    }
  }
}



// 定义两个个使用混入对象的组件
var Component1 = new Vue({
 name: 'Component1',
  mixins: [myMixin]
```

```
})
var Component2 = new Vue({
 name: 'Component2',
  mixins: [myMixin]
})

// 输出
// hello fromComponent1's mixin!
// hello fromComponent2's mixin!
```

选项合并

当组件和混入对象含有同名选项时，这些选项将以恰当的方式进行“合并”

比如，数据对象在内部会进行递归合并，并在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，因此都将被调用。另外，混入对象的钩子将在组件自身钩子之前调用。

值为对象的选项，例如 methods、components 和 directives，将被合并为同一个对象。两个对象键名冲突时，取组件对象的键值对。

注意：Vue.extend() 也使用同样的策略进行合并。

这里也不难理解，官方看一遍就能理解；不过vue也提供了相应的配置可以自定义选项合并策略

插件

插件通常用来为 Vue 添加全局功能。插件的功能范围没有严格的限制——一般有下面几种：

1. 添加全局方法或者属性。如: vue-custom-element
2. 添加全局资源：指令/过滤器/过渡等。如 vue-touch
3. 通过全局混入来添加一些组件选项。如 vue-router
4. 添加 Vue 实例方法，通过把它们添加到 Vue.prototype 上实现。
5. 一个库，提供自己的 API，同时提供上面提到的一个或多个功能。如 vue-router

好的，只要是能给vue全局增加功能的js模块就是插件；所以vuejs推荐的资料仓库也叫 awesome-vue