

组件化和模块化

从设计上来看，组件强调复用，模块强调职责(内聚、分离)

组件

组件（Component），或者构件。使用非常比较广泛，它的核心意义在于复用，相对模块，对于依赖性有更高的要求。

前端Web应用中的组件，是指一些设计为通用性的，用来构建较大型应用程序的软件，这些组件有多种表现形式。它可以是有UI（用户界面）的，也可以是作为“服务”的纯逻辑代码。

组件更偏UI层面，因为有视觉上的表现形式，UI组件更容易理解。UI组件简单的例子包括按钮、输入框和文本域。菜单按钮、标签页、日历、选项菜单或者所见即所得的富文本编辑器则是一些更加高级的例子。

我们将相关的一些功能组织在一起，把一切封装起来，而在组件的例子中，就可能是相关的功能逻辑和静态资源：JavaScript、HTML、CSS以及图像等。这就是我们所说的内聚。这种做法将让组件更容易维护，并且这么做之后，组件的可靠性也将提高。同时，它也能让组件的功能明确，增大组件重用的可能性。

模块

模块（Module），或模组。它的核心意义是分离职责，属于代码级模块化的产出。它本身是提供服务的功能逻辑，是一组具有一定内聚性代码的组合，职责明确。

模块化么偏代码（js）层面，把业务代码（功能）切分开。简单来说，模块化就是将一个程序按照其功能做拆分，分成相互独立的模块，以便于每个模块只包含与其功能相关的内容。模块我们相对熟悉，比如登录功能可以是一个模块，搜索功能可以是一个模块。