# vue概念

是一套用于构建用户界面的前端框架 构建出美观、舒适、好用的网页。

1. 传统

**结构编写：**

在传统的 Web 前端开发中，是基于 jQuery + 模板引擎 的方式来构建用户界面的。

基于模板引擎技术，把数据渲染到页面上。 优点：初步解放了前端开发者，从此不用手动拼接字符串来渲染网页结构了。 缺点：1.需要定义大量的模板结构；2.缺少语法高亮和智能提示；3.数据变化时需要重新调 用模板编译的函数，否则页面结构不会更新；

**交互处理：**

基于 jQuery 技术，处理用户和网页之间的交互行为。 优点：屏蔽了 DOM API 之间的兼容性，提高了 DOM 操作的效率和体验。 缺点：当业务复杂时、数据变化频繁时，前端程开发者需要把大量的时间和精力浪费在 DOM 的操作上，而不是核心业务的处理上。

1. vue基本简介

**结构编写：**

基于 vue 中提供的指令，可以方便快捷的渲染页面的结构（乐不思蜀）。 数据驱动视图（只要页面依赖的数据源变化，则页面自动重新渲染） Ps：指令是 vue 为开发者提供的模板语法，用来辅助开发者渲染页面的结构。

**交互处理：**

基于 vue 中提供的事件绑定，可以轻松处理用户和页面之间的交互行为。 Ps：开发者把工作的重心放在核心业务的实现上

**vue 全家桶**

官方给 vue 的定位是前端框架，因为它提供了构建用户界面的一整套解决方案（俗称 vue 全家桶）：

⚫ vue（核心库）

⚫ vue-router（路由方案）

⚫ vuex（状态管理方案）

⚫ vue 组件库（快速搭建页面 UI 效果的方案）

**以及辅助 vue 项目开发的一系列工具：**

⚫ vue-cli（npm 全局包：一键生成工程化的 vue 项目 - 基于 webpack、大而全）

⚫ vite（npm 全局包：一键生成工程化的 vue 项目 - 小而巧）

⚫ vue-devtools（浏览器插件：辅助调试的工具）

⚫ vetur（vscode 插件：提供语法高亮和智能提示）

1. vue特性

**数据驱动视图（单向**数据绑定**）**

在使用了 vue 的页面中，vue 会监听数据的变化，从而自动重新渲染页面的结构。

**好处：**当页面数据发生变化时，页面会**自动重新渲染**！

**注意：**数据驱动视图是**单向**的数据绑定。

**双向数据绑定**

在填写表单时，双向数据绑定可以辅助开发者在**不操作 DOM** 的前提下，**自动**把用户填写的内容**同步到数据源** 中。

**好处**：开发者不再需要手动操作 DOM 元素，来获取表单元素最新的值！

1. MVVM概念

**M**: Model表示当前页面渲染时所依赖的数据源。

**V**: View 表示当前页面所渲染的 DOM 结构。

**VM**: ViewModel 表示 vue 的实例

1. 模板语法

### ① 内容渲染指令

**v-text** 会**覆盖**元素内**默认的值**

**{{ }} 插值表达式 用的多**

**v-html**

v-text 指令和插值表达式只能渲染纯文本内容。如果要把包含 **HTML 标签的字符串**渲染为页面的 **HTML 元素**， 则需要用到 v-html 这个指令：

### ② 属性绑定指令

v-bind： 等价于 ：

### ③ 事件绑定指令

### ④ 双向绑定指令

### ⑤ 条件渲染指令

### ⑥ 列表渲染指令