### Vue的优点：

1、轻量级框架，大小只有几十kb。

2、双向数据绑定：

Vue自动对页面中某些数据的改动做出同步的响应，即input中的v-model属性。通过MVVM思想来实现双向绑定，让开发者不再需要操作dom对象，而只需要专注思考业务逻辑。

此处就可以区别于jQuery。jQuery的特点就是通过操作dom属性，来方便地对页面进行修改。而这也导致了浏览器需要不断地去渲染dom树，导致页面卡顿。而vue的虚拟dom会根据页面变动（如js操作）预先进行dom的操作计算，判定一些页面操作是否需要重新渲染浏览器dom树，最后将优化的最终计算结果交给浏览器，提升了性能。

3、组件化

Vue将单页面应用中的各个模块拆分到各个单独的子组件中，子组件只需要考虑自己的业务逻辑，使得各个模块的开发互不影响，如同编程时的模块封装。

### Vue获取dom：

1、给需要获取的便签添加属性ref=’dom’

通过this.$refs.dom即可开始操作该dom元素

2、如果是需要用原生document获取，则函数应该方法mounted中，因为此时所有dom节点都已经渲染。

### Vue的常用指令：

1、v-if：根据表达式的值的真假条件渲染元素。在切换时元素及它的数据绑定 / 组件被销毁并重建。

2、v-show：根据表达式之真假值，切换元素的 display CSS 属性。

3、v-for：循环指令，基于一个数组或者对象渲染一个列表，vue 2.0以上必须需配合 key值 使用。

4、v-bind：动态地绑定一个或多个特性，或一个组件 prop 到表达式。

5、v-on：用于监听指定元素的DOM事件，比如点击事件。绑定事件监听器。

6、v-model：实现表单输入和应用状态之间的双向绑定

7、v-pre：跳过这个元素和它的子元素的编译过程。可以用来显示原始 Mustache 标签。跳过大量没有指令的节点会加快编译。

8、v-once：只渲染元素和组件一次。随后的重新渲染，元素/组件及其所有的子节点将被视为静态内容并跳过。这可以用于优化更新性能。

### Vue-loader与其作用：

vue-loader是解析 .vue 文件的一个加载器，可以把template/js/style中的一些非原生语法转换成 js 模块。

### Vue的v-on监听多个方法：

即键值对方式，key=时间，value=方法名。对象形式赋值

### Vue的data为什么是函数：

基于js的函数才有作用域的特点。

为了让每个组件都拥有自己的数据的作用域，而不会被其他组件污染。

### Vue的渐进式理解：

即vue只做好自己的擅长的事情，力求轻量级。功能的增加按照业务需求循序渐进。

### Vue的v-if和v-for的优先级：

v-for大于v-if。避免在同一个标签使用，而是放到computed中计算并获得符合条件的数据。

### Vue的常用修饰符：

v-model后拥有：

Lazy：不再在输入时触发同步，而是在光标离开input输入框时改变。

Number：自动转为数值类型

Trim：消除首尾空白

事件@修饰符：

Stop：防止事件冒泡，点击子事件不会捕获父事件

Prevent=“fun”：阻止默认事件，转向fun自定义事件。

Once：只执行一次

Capture：由外到内的冒泡顺序捕获

Self：只触发自己范围内的事件，不包含子元素

### Vue的两个核心：

1. 数据驱动
2. 组件化

### Vue-router传递参数的两种方式：

一、params方法

1. 在路由配置文件中，在目标路由，如/detail的尾部添加/:uid，即/detail/:uid
2. 在前往该路由的地方，在url参数里添加/uiddata
3. 即可在目标路由中通过$route.params.uid获得参数

二：query方法：

1. 在前往目标路由的地方，传入对象，而不是url
2. 对象内容{

path: ‘/detail’,

query: {

uid: data

}

}

3、在目标路由里面通过$route.query.uid获取