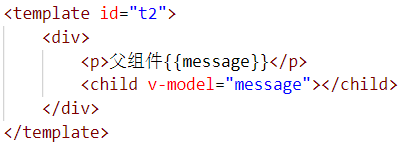
****1.VUE的特性以及优缺点****  
1.轻量级的框架 2.双向数据绑定 3.指令 4.插件化

****vue的优点****  
1、简单易用 2、灵活渐进式 3、轻量高效 压索之后20KB大小 4、MVVM  
（4-1）数据驱动视图 4-2）常规的操作方式都是DOM （4-3）、普通的javascript数据  
5、组件化  
组件化优点 提高开发效率 方便重复使用 简化调试步骤 提升整个项目的可维护性  
便于协同开发

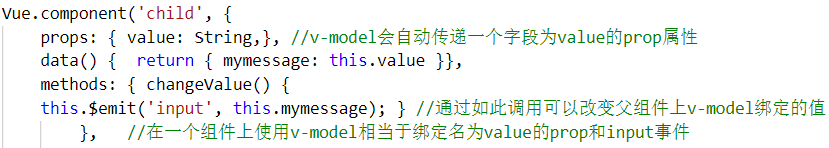
****vue的缺点****  
1、VUE不支持IE8 2、生态环境差不如angular和react

3.封 装程度更高，调试成本更大，难以定位 Bug。4、社区不大.功能仅限于 view 层，**Ajax 等功能**需要额外的库。对开发人员要求较高。开发的话，需要 webpack，不然很难用，最好配合 es6。不过Vue-cli把webpakc也隔离的差不多了

1. 由于变量声明自带不可删除属性，比较var num = 1 跟 num = 1，前者是**变量声明**，带不可删除属性，因此无法被删除；后者为**全局对象**window**的一个属性**，因此可以从全局变量中删除
2. vue中组件如何交互5:****props和$emit**** v-model

父组件通过v-model传递值给子组件时，会自动传递一个value的prop属性，在子组件中通过this.$emit(‘input’,val)自动修改v-model绑定的值,父组件同时改变

****vuex localstorage存储后传参****

1. b标签和strong有什么区别:1)b标签和strong标签是两种不同的标签。B标签表示粗体，<strong>标签是我们要强调的地方，这两个标签具有不同的含义 2)搜索引擎更关注strong标签下的关键词。strong标签更加面向内容。在HTML中，关键词被标记。
2. instance与typeof 区别

|  |  |  |  |
| --- | --- | --- | --- |
|  | typeof | Instanceof | ****constructor**** |
| 1 | 能够检测出了null之外的基本类型String、Number、Boolean、Null、Undefined、Symbol; | 用来判断 A 是否为 B 的**实例**, instanceof 检测的是**原型**。 | 存在于Object.prototype的**属性**,返回对象的构造函数的引用，不是函数名，语法： object.constructor  比如 JavaScript 数组 constructor 属性返回 function Array() { [native code] }  [1,2].constructor===  Array //true |
| 2 | 对于**对象类型**能判断出function,其他的都为Object | 但是instanceof可以判断出[]是Array的实例，同时也认为是Object的实例???  instanceof 只能用来判断两个对象是**否属于实例关系**， 而不能判断一个对象实例具体属于哪种**类型**。 |
| 3 | Instanceof本质是判断 A对象的原型链上是否存在B.prototype  语法:object(必须是对象) instanceof B  之后增加了Array.isArray()方法判断这个值是不是数组的。 |

JS内置对象类型(adore) **A**rray.Boolean.**D**ate、**O**bject.Reg**E**xp.(4爱:爱死 爱慕)String.Math.Number.Function

1. 有哪些可能引起前端安全的问题？
2. **跨站脚本攻击**（Cross-Site Scripting，**XSS**）：一种**代码注入**方式，为了与 CSS 区分所以被称为 XSS。早期常见于网络论坛，起因是网站没有对用户的输入进行严格的限制，使得攻击者可以**将脚本上传到帖子**让其他人在浏览到有**恶意脚本**的页面。其注入方式很简单，包括但不限于 JavaScript / VBScript / CSS / Flash 等；

2**)跨站请求伪造**（**Cross-Site** Request Forgeries，CSRF）：攻击者诱导受害者进入第三方网站(目标网站的cookie在该网站中可用)，在第三方网站中向被攻击网站发送跨站请求。利用受害者在被攻击网站已经获取的注册凭证，绕过后台的用户验证，达到冒充用户对被攻击的网站执行某项操作的目的。

GET 类型的 CSRF(普通的 HTTP 请求) POST 类型的 CSRF(自动提交的表单) 链接类型的 CSRF

|  |  |  |  |
| --- | --- | --- | --- |
| XSS防御 | | | CSRF防御 |
| 攻击特点 | | | |
| 攻击者提交恶意代码  浏览器执行恶意代码=>1)改成纯前端渲染，把代码和数据分隔开； | | 发生在第三方域名；  攻击者不能获取 cookie 信息，只能使用。 | |
| 2)内容安全策略（Content Security Policy，CSP）  严格的 CSP 在 XSS 的防范中可以起到以下的作用：  禁止加载外域代码，防止复杂的攻击逻辑；  禁止外域提交，网站被攻击后，用户的数据不会泄漏到外域；  禁止内联脚本执行.未授权的脚本执行；  合理使用上报可以及时发现 XSS，利于尽快修复问题。  可以使用以下两种方式开启内容安全策略：  HTTP Header 中的 **Content-Security-Policy**  <meta http-equiv="Content-Security-Policy">  3)输入内容长度控制  对于不受信任的输入，都应该先定一个**合理的长度。**虽然无法完全防止 XSS 的发生，但是可以增加 XSS 攻击的难度。 | 1)**阻止**不明外域的访问  同源检测:用 Origin Header 确定来源域名  Samesite Cookie  2)提交时要求**附加本域**才能获取的信息  **Token**:要求所有的用户请求都携带一个 CSRF 无法获取到的 Token，服务器判断 Token 是否正确，从而将正常的请求和攻击的请求区分开。  **双重 Cookie** 验证:在请求接口时，在请求参数中附带上对应的 cookie,攻击者无法获取 | | |

**3)iframe滥用**：iframe 中的内容是由**第三方来提**供的，默认情况下它们不受我们控制，它们可以在 iframe 中运行 **JavaScript 脚本、Flash 插件、弹出对话框**等，这些可能会破坏前端用户的体验；

严格来说这个不算是安全问题，但是如果是要防范自己网站被人用 iframe 嵌套的话：

添加响应头 在响应头中加一个 **X-Frame-Options** 取值有三种，大部分浏览器都支持：

DENY：**拒绝当**前页面加载任何 Frame 页面； SAMEORIGIN：Frame 页面只能加载在**同源网站**下；

ALLOW-FROM origin：origin 为允许 Frame 页面加载的地址。

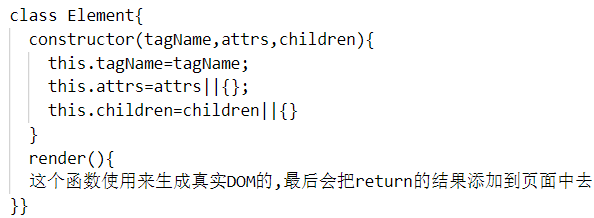
4)**恶意第三方库**：无论开发服务端还是客户端，绝大多数时候我们都在借助**开发框架和各种第三方工具库**进行快速开发，一旦第三方库被植入恶意代码人容易引起安全问题。

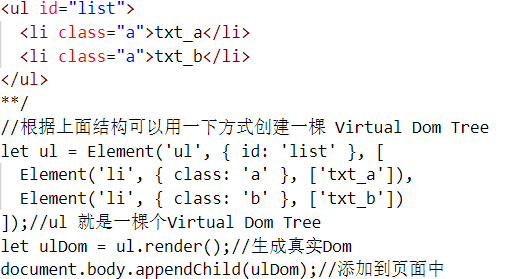
5)网络劫持(分类) **DNS劫持**：输入京东被强制跳转到淘宝就属于 DNS劫持。 **HTTP劫持**：例如访问百度，但是一直有贪玩蓝月的广告。由于 HTTP 是明文传输，运营商会修改你的 HTTP 响应内容（即加广告）。

网络劫持防御 DNS劫持由于涉嫌违法，已经被监管起来，现在很少会遇到 DNS 劫持。

但是 HTTP 劫持依然非常盛行，最有效的方法就是**使用 HTTPS，将 HTTP 加密起来**，运营商无法获取明文，就无法劫持你的响应内容。

8.Vue Virtual Dom 原理和 Diff原理(极简版)

1)用js来模拟DOM中的节点(树)。并渲染(render)这个DOM树

2)比较新老DOM树，得到比较的差异对象(patch)

3)把差异对象应用到渲染的DOM树。

* Diff(tree,ntree)算法就是进行虚拟节点

**对比**，并返回一个patch对象，用来存储两个节点不同的地方，最后用patch记录的消息去**局部更新**Dom( patch (node, patches){})。人话:diff的过程就是**调用名为patch的函数**，比**较新旧节点**，一边比较一边给真实的DOM打补丁. 两个节点之间的差异有总结起来有下面4种

0 直接替换原有节点 1 调整子节点,包括移动删除等3 修改节点属性 2 修改节点文本内容

1. Proxy 相比较于 defineProperty 的优势

|  |  |  |
| --- | --- | --- |
|  | Proxy | defineProperty |
| 监听 | 叫做代理器，它可以为一个对象设置代理，即监听**对象本身，**任何访问当前被监听的对象的操作，无论是对象本身亦或是对象的字段，都会被 Proxy 拦截，因此可以使用它来做一些双向绑定的操作。 | 对象的字段而非对象本身 |
| 因此无法监听动态插入对象的字段只能手动为其设置设置监听属性。  无法监听对象中数组的变化，因此其他基于 Object.defineProperty 都对数组做了一定的 Hack 处理。 |
| 鉴于兼容性的问题，目前仍然主要是使用 Object.defineProperty 更多，但是随着 Vue/3 的发布，Proxy 应该会逐渐淘汰 后者 | |