

## 1. Vue.js

### a. 语句

- i. 循环语句: v-for

### b. 计算属性

- i. computed: 计算属性处理一些复杂的逻辑

- 1. computed setter: computed 属性默认只有getter, 在需要时你可以提供一个setter

- ii. reversedMessage

### c. 监听属性

- i. 通过watch 来响应数据的变化

### d. onclick定义一个

- i. v-on 定义一个属性来接收

### e. 表单: 用v-model指令在表单控件元素上创建双向数据绑定

- i. input 和textarea元素中使用v-model 实现双向数据绑定。

### f. 自定义事件

- i. 父组件使用props传递数据给子组件

- ii. Vue.js 自定义指令: 除了默认设置的

### g. 过渡和动画

- i. vue在插入、更新和使用的过程中

## 2. CSS动画

- a. 在动画v-center类名在节点插入DOM后不会立即删除, 而是在animationend事件触发时删除的

### b. 同时使用过渡和动画

- i. Vue设置相应的事件监听器, 可以知道过渡是否完成, 可以是transitionend 或

## 3. 基于Vue.js的可视化

- a. 富文本编辑: 文本输入、文本格式编辑、跳转

### b. 页面的存储方式:

- c. 动态: 各种拖拽、弹窗能力

- i. clone-element: 从候选组件列表中拖动到预览页面上。
    - ii. 组件从定方式上分为: 普通组件和绝对定位组件, 普通组件按照文档流自然排列, 绝对定位组件可以在容器内自由定位。
    - iii. mousedown、mousemove、mouseup
  - d. 旋转: 是为绝对定位组件提供的编辑能力
    - i. 旋转能力同样被封装在ElementWrapper中
  - e. 设置器: 设置器是可视化编辑的核心
    - i. 一个组件支持的设置项由组件开发者决定并声明
      - 1. 支持8种设置组件: 像素值、文字、数字、图片、颜色、单选、多选、文字域
      - 2. 设计的架构具有扩展新的设置组件的能力
  - f. Architecture、Pattern、FrameWork
    - i. 经典的JSP + Servlet + JavaBean 的MVC
    - ii. SSM (Spring+SpringMVC+Mybatis) 和 SSH (Spring+Struts+Hibernate) 的框架应用
    - iii. 前端框架:
      - 1. 前端架构的演变: 静态页面/原生JS ----> jQuery ----> Angular/Vue/React
4. 体系架构、框架、设计模式
- a. 体系架构: 确定了系统整体结构、层次划分, 不同部分之间的协作等设计考虑
  - b. 框架: 框架比架构更具体, 更偏重于技术。确定框架后, 软件体系结构也随之确定。对于同一软件体系结构 (比如web开发中的MVC), 可以有多重框架来实现。
  - c. 设计模式: 是某个问题的设计, 设计可被不同语言用不同方式来实现, 设计模式的思想可以在框架设计中进行应用。
5. 架构的起源:
- a. Architecture 软件架构是系统的一个草图。软件架构描述的对象是直接构成系统的抽象的组件。各个组件之间的连接则明确和相对细致地描述组件之间的通讯。在实现阶段, 这些抽象组件被细化成了实际的组件, 比如具体某个类或者对象。在面向对象领域, 组件之间的连接通常用接口来实现。
  - b. 软件架构: 在一定的设计原则基础上, 从不同角度对组成系统的各个部分进行搭配和安排。

6. 1

7. 1

8. 1

9. 1

10. 1

11. 1

12. 1

13. 1

14.