

VUE

- vue基础

- 历史介绍

- angular 09年
 - react 2013
 - vue 2014

- 前端框架与库的区别

- jquery库——>DOM (操作DOM) +请求
 - art-template库——>模板引擎 将数组当做列表直接输出
<https://www.cnblogs.com/shiyou00/p/6841801.html>
<http://aui.github.io/art-template/docs/>
 - 框架=全方位功能齐全
 - 简易的DOM体检+发请求+模板引擎+路由功能
 - KFC的世界里，库就是一个小套餐，框架就是全家桶
 - 代码上的不同
 - 一般使用库的代码，是调用某个函数，我们自己把控库的代码的调用
 - 一般使用框架，其框架在帮助我们运行我们编写好的代码
 - 框架：
 - 初始化自身的一些行为
 - 执行你所编写的代码
 - 释放一些资源

- Vue起步

- 引包
 - 启动 new Vue({el: 目的地, template: 模板内容})
 - options
 - vue目录结构

- Vue文件介绍

- 插值表达式
 - 什么是指令
 - vue中的常用指令
 - vue复习

- 注册全局组件

- 应用场景：多处使用的公共性功能组件，就可以注册成全局组件
- 全局API：Vue.component("组件名",组件对象)

-

```
//注册公共的全局组件，省略很多的components
Vue.component(
  'my-btn',{
    template: `<button>公共的按钮</button>`
  })
```

- 附加功能：过滤器&监视改动

- slot 父向子传DOM

- 组件的生命周期 生命周期钩子

- 所有的生命周期钩子自动绑定this上下文到实例中，因此你可以访问数据，对属性和方法进行运算。这意味着——你不能使用箭头函数来定义一个生命周期方法——(例如 created: () => this.fetchTodos())。这是因为箭头函数绑定了父上下文，因此 this 与你期待的 Vue 实例不同，this.fetchTodos 的行为未定义。

- beforeCreate

- 类型function
- 详细
 - 在实例初始化之后，数据观测(data observer)和 event/watcher事件配置之前被调用

- created

- 类型function
- 详细
 - 在实例创建完成后被立即调用。在这一步，实例已完成以下配置：数据观测(data observer),属性和方法的运算，watch/event事件回调。然而，挂载阶段还没开始，\$el属性目前不可见。

- beforeMount

- 类型function
- 详细
 - 在挂载开始之前被调用：相关的render函数首次被调用。
 - 该钩子在服务器端渲染期间不被调用。

- mounted

- 类型function
- 详细

- el被新创建的vm.\$el替换，并挂载到实例上去之后调用该钩子。如果root实例挂载了一个文档内元素，当mounted被调用时vm.\$el也在文档内。
- 注意mounted不会承诺所有的子组件也都一起被挂载。如果你希望等到整个视图都渲染完毕，可以用vm.\$nextTick替换掉mounted：

```
• mounted: function () {  
  • this.$nextTick(function () {  
    • // Code that will run only after the  
    • // entire view has been rendered  
  • })  
• }
```

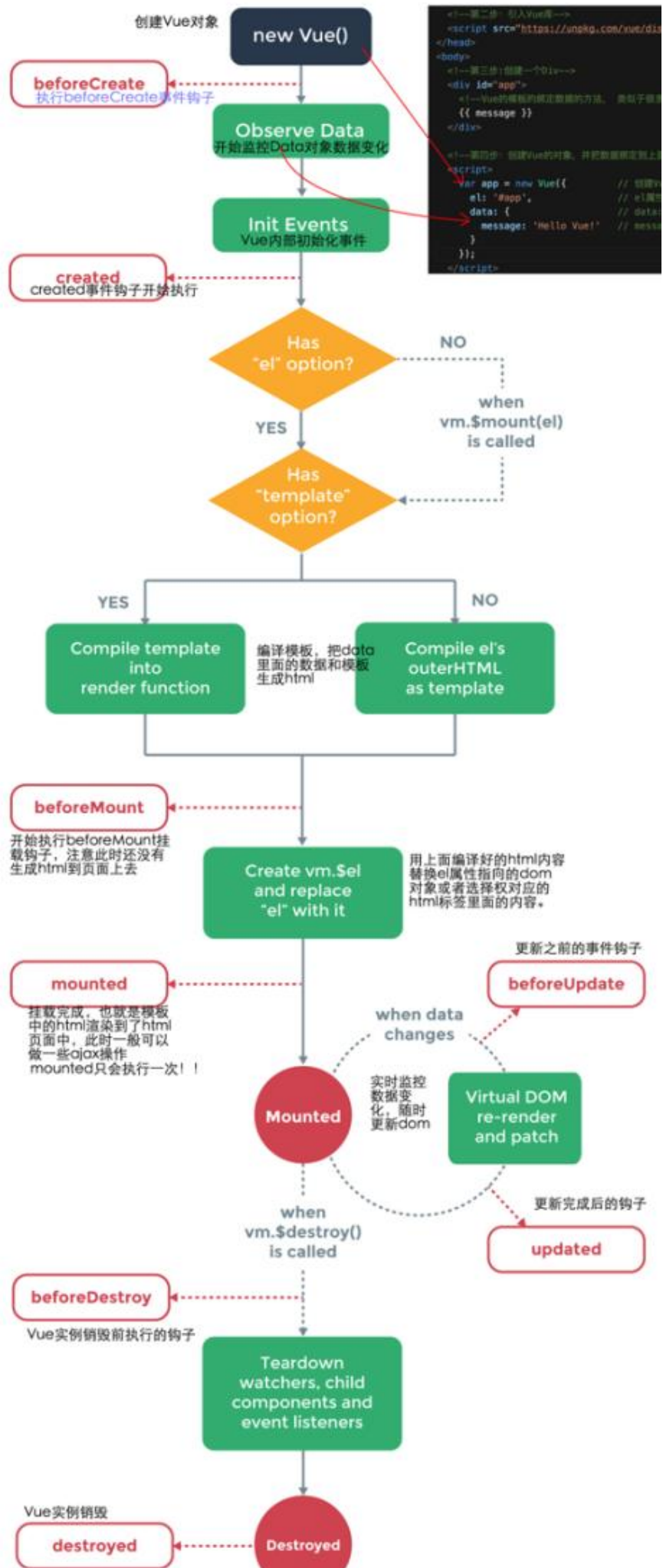
- 该钩子在服务器端渲染期间不被调用。

- 其他钩子———

<https://cn.vuejs.org/v2/api/#%E9%80%89%E9%A1%B9-%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F%E9%92%A9%E5%AD%90>

- 生命周期图示

-



- 应用
- keep-alive
- vue补充
- 路由
 - 路由中有三个基本的概念route,routes,router
 - 1.route, 它是一条路由, 由这个英文单词也可以看出来, 它是单数, Home按钮=>home内容, 这是一条route, about按钮=>about内容, 这是另一条路由。
 - 2.routes是一组路由, 把上面的每一条路由组合起来, 形成一个数组。[{home按钮=>home内容},{about按钮=>about内容}]
 - 3.router是一个机制, 相当于一个管理者, 它来管理路由。因为routes只是定义了一组路由, 它放在那里是静止的, 当真正来了请求肿么办? 就是当用户点击home按钮的时候, 肿么办? 这是router就起作用了, 它到routes中去查找, 去找到对应的home内容, 所以页面中就显示了home内容。
 - 4.客户端中的路由, 实际上就是dom元素的显示和隐藏。当页面中显示home内容的时候, about中的内容全部隐藏, 反之也是一样。客户端路由有两种实现方式: 基于hash和基于HTML5 history api.
 - 路由原理
 - 传统开发方式url改变后, 立刻发送请求, 响应整个页面, 渲染整个页面
 - SPA锚点值改变后不会发起整个页面请求, 发起ajax请求, 局部改变页面数据
 - 页面不跳转 用户体验更好
- SPA
- 基本使用
- 阶段总结
 - vue-router使用步骤:
 - 1引入 `<script src="vue-router.js"></script>`
 - 2安装插件 `Vue.use(VueRouter)`
 - 3创建路由实例 `var myRouter=new VueRouter({ routes : [] })`
 - 4配置路由规则 `routes:[{},{, {name:'login',path:'/login',component:Login}]` Login为插件
 - 5将路由对象关联vue `new Vue({router:myRouter})`
 - 6留坑 `templates:'<router-view></router-view>'`

- router-link to='/xxx' 命名路由
 - 1.在路由规则对象中加入name属性 routes:


```
[[{name:'login',path:'/login',component:Login}]]
```
 - 2.在router-link中 :to="{name:'login'}"
- 生僻API梳理
 - 1.Vue.use(插件对象);//过程中会注册一些全局组件，及给vm或者组件对象挂载属性
 - 2给vm及组件对象挂载的方式：
 - Object.defineProperty(Vue.prototype,'\$router',{
 - get:function(){
 - return 自己的router对象
 - }
 - })

• router-link传递参数

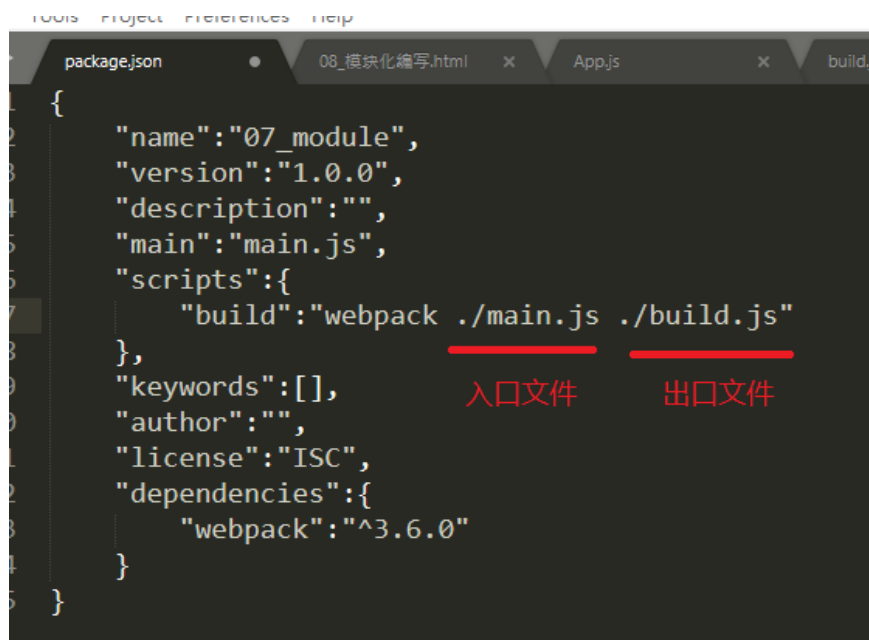
• vue-router中的对象

- \$route路由信息对象，只读对象
- \$router路由操作对象，只写对象

• 嵌套路由

• 模块化

- 配置文件 package.json



```

{
  "name": "07_module",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "build": "webpack ./main.js ./build.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "webpack": "^3.6.0"
  }
}
  
```

入口文件 出口文件

- import App from './App.js'; //导入
- export default app; //导出
- import {num1,num2} from './app.js' //导入
- export var num1=1; //作为一整个对象中的key导出 声明并导出

- var num2=2; export {num2}; //作为整个对象中的key导出，先声明再导出
 -
 - axios
 - <https://www.kancloud.cn/yunye/axios/234845>
- 多视图和嵌套路由
-
- 合并请求
 -