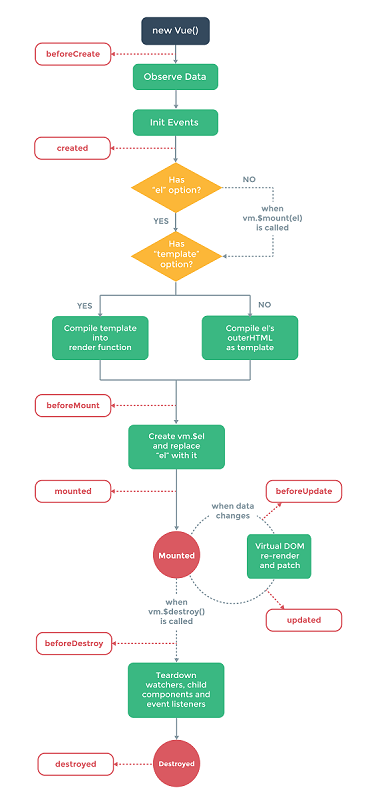
[vuejs2.0 vue实例的生命周期](https://www.cnblogs.com/caizhenbo/p/6418291.html)

https://www.cnblogs.com/caizhenbo/p/6418291.html

每个 Vue 实例在被创建之前都要经过一系列的初始化过程。例如，实例需要配置数据观测(data observer)、编译模版、挂载实例到 DOM ，然后在数据变化时更新 DOM 。下图展示的就是一个vue实例对象的生命周期。



从图上我们可以看到vue在生命周期中有这些状态，**beforeCreate,created,beforeMount,mounted,beforeUpdate,updated,beforeDestroy,destroyed**。Vue在实例化的过程中，会调用这些生命周期的钩子，给我们提供了执行自定义逻辑的机会。那么，在这些vue钩子中，vue实例到底执行了那些操作，我们先看下面执行的例子

[复制代码](javascript:void(0);)

var vm = new Vue({

el: "#container",

data: {

test : 'hello world'

},

beforeCreate: function(){

console.log(this);

showData('创建vue实例前',this);

},

created: function(){

showData('创建vue实例后',this);

},

beforeMount:function(){

showData('挂载到dom前',this);

},

mounted: function(){

showData('挂载到dom后',this);

},

beforeUpdate:function(){

showData('数据变化更新前',this);

},

updated:function(){

showData('数据变化更新后',this);

},

beforeDestroy:function(){

vm.test ="3333";

showData('vue实例销毁前',this);

},

destroyed:function(){

showData('vue实例销毁后',this);

}

});

function realDom(){

console.log('真实dom结构：' + document.getElementById('container').innerHTML);

}

function showData(process,obj){

console.log(process);

console.log('data 数据：' + obj.test)

console.log('挂载的对象：')

console.log(obj.$el)

realDom();

console.log('------------------')

console.log('------------------')

}

[复制代码](javascript:void(0);)



1. 我们可以看到，vue对象初始化过程中，会执行到**beforeCreate,created,beforeMount,mounted** 这几个钩子的内容
2. **beforeCreate** ：数据还没有监听，没有绑定到vue对象实例，同时也没有挂载对象
3. **created** ：完成数据观测、属性和方法的运算，初始化事件，$el属性还没显示出来。数据已经绑定到了对象实例，但是还没有挂载对象
4. **beforeMount**: 相关的render函数首次被调用。模板已经编译好了，根据数据和模板已经生成了对应的元素对象，将数据对象关联到了对象的 $el属性，$el属性是一个HTMLElement对象，也就是这个阶段，vue实例通过原生的createElement等方法来创建这个html片段，准备注入到我们vue实例指明的el属性所对应的挂载点。（简单来说就是已经生成了新的内容）
5. **mounted**:将$el的内容挂载到了el，相当于我们在jquery执行了$(el).html($el),生成页面上真正的dom，上面我们就会发现dom的元素和我们$el的元素是一致的。在此之后，我们能够用方法来获取到el元素下的dom对象，并进 行各种操作。（就是可以对生成的内容进行各种操作）

当我们的data发生改变时，会调用beforeUpdate和updated方法



**beforeUpdate** ：数据更新到dom之前，我们可以看到$el对象已经修改，但是我们页面上dom的数据还没有发生改变

**updated**: dom结构会通过虚拟dom的原则，找到需要更新页面dom结构的最小路径，将改变更新到dom上面，完成更新

实例的销毁



**beforeDestroy：**在实例销毁前调用，实例还是可使用。只是解绑了事件的监听还有watcher对象数据与view的绑定，即数据驱动

**destroed** : 在实例销毁之后调用。调用后，所有的事件监听器会被移除，所有的子实例也会被销毁。该钩子在服务器端渲染期间不被调用。