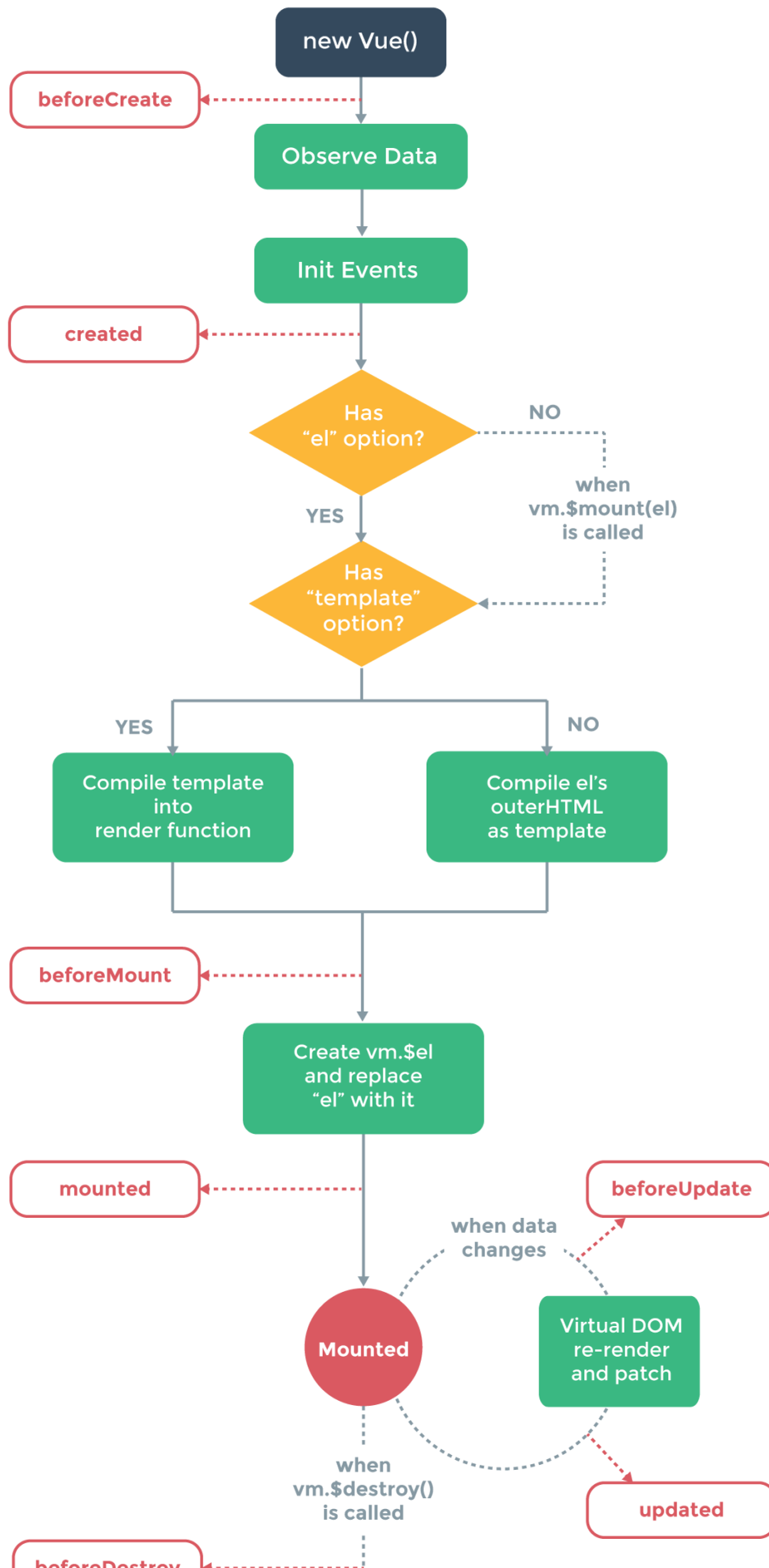
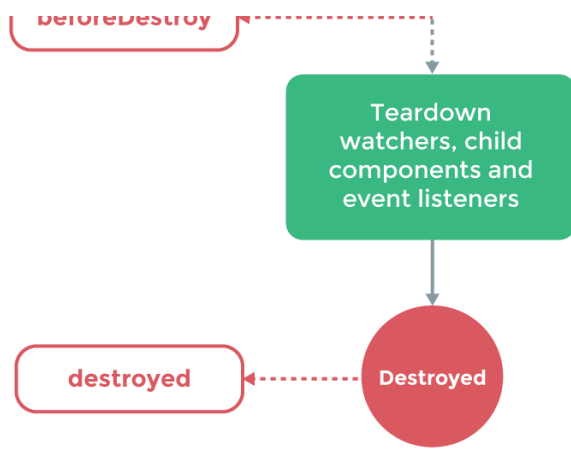


Vue生命周期





这是Vue文档里关于实例生命周期的解释图

那么下面我们来进行测试一下

```
<section id="app-8">
  {{data}}
</section>
```



```
var myVue=new Vue({
  el:"#app-8",
  data:{
    data:"aaaaa",
    info:"nono"
  },
  beforeCreate:function(){
    console.log("创建前=====")
    console.log(this.data)
    console.log(this.$el)
  },
  created:function(){
    console.log("已创建=====")
    console.log(this.info)
    console.log(this.$el)
  },
  beforeMount:function(){
    console.log("mount之前=====")
    console.log(this.info)
    console.log(this.$el)
  },
  mounted:function(){
    console.log("mounted=====")
    console.log(this.info)
    console.log(this.$el)
  },
  beforeUpdate:function(){
    console.log("更新前=====");
```

```

    },
    updated:function(){
      console.log("更新完成=====");
    },
    beforeDestroy:function(){
      console.log("销毁前=====")
      console.log(this.info)
      console.log(this.$el)
    },
    destroyed:function(){
      console.log("已销毁=====")
      console.log(this.info)
      console.log(this.$el)
    }
  })
}

```



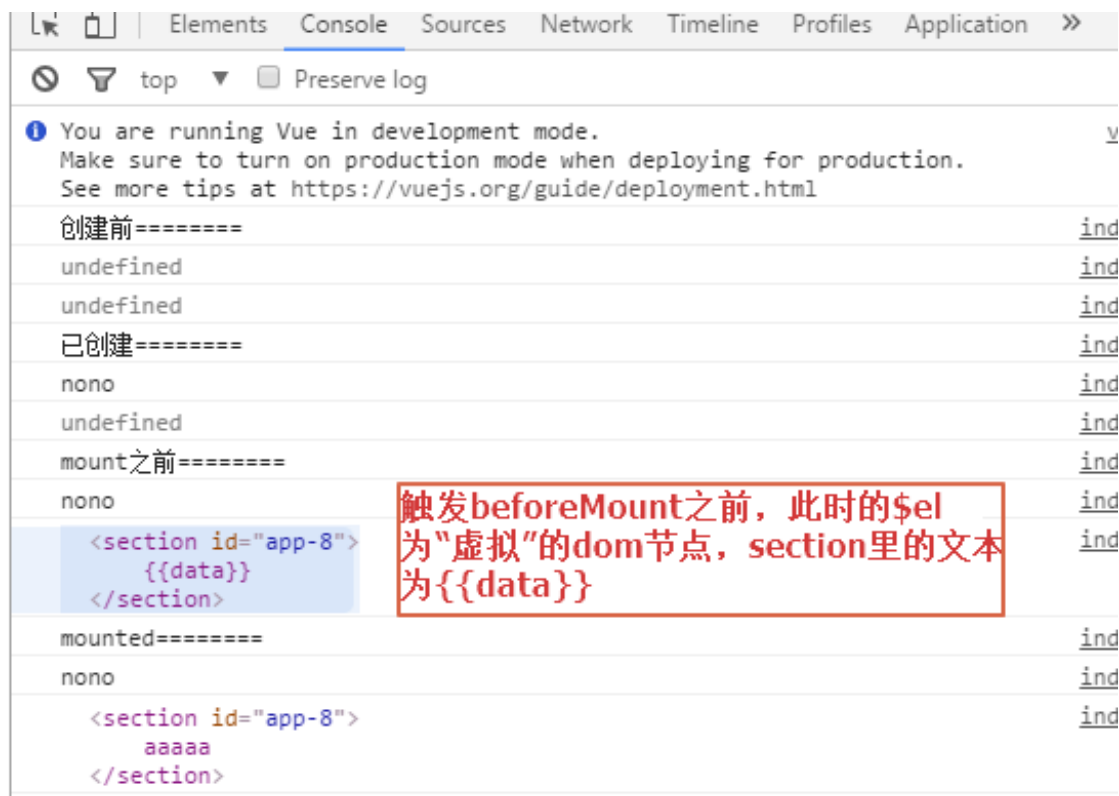
代码如上，浏览器开始加载文件

aaaaa

The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The logs are as follows:

Message	Source
Vue.js development mode warning: You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at https://vuejs.org/guide/deployment.html	vue.js:6109
创建前=====	index.html:60
undefined	index.html:61
undefined	index.html:62
已创建=====	index.html:65
nono	index.html:66
undefined	index.html:67
mount之前=====	index.html:70
nono	index.html:71
<section id="app-8"> {data} </section>	index.html:72
mounted=====	index.html:75
nono	index.html:76
<section id="app-8"> aaaaa </section>	index.html:77
Download the Vue Devtools extension for a better development experience: https://github.com/vuejs/vue-devtools	vue.js:6126

bbbbbb



bbbbbb

section#app-8 | 486 x 18



mounted触发之前，创建真实的dom元素替换“虚拟”的dom元素
此处我理解为，将数据渲染到“虚拟”的dom中并插入 此时的\$el=document.getElementById('***')

由上图可知：

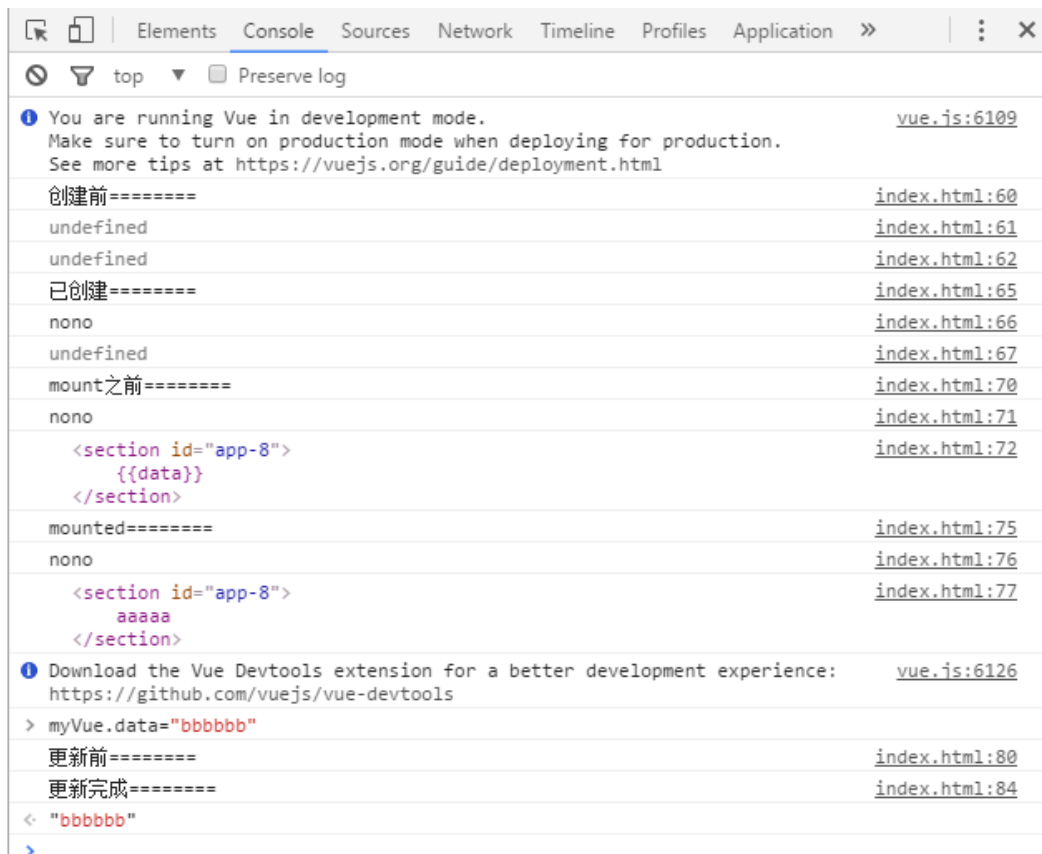
- 1、beforeCreate 此时\$el、data 的值都为undefined
- 2、创建之后，此时可以拿到data的值，但是\$el依旧为undefined
- 3、mount之前，\$el的值为“虚拟”的元素节点
- 4、mount之后，mounted之前，“虚拟”的dom节点被真实的dom节点替换，并将

其插入到dom树中，于是在触发mounted时，可以获取到\$el为真实的dom元素()

```
myVue.$el===document.getElementById("app-8") // true
```

接着，在console中修改data，更新视图

bbbbbb



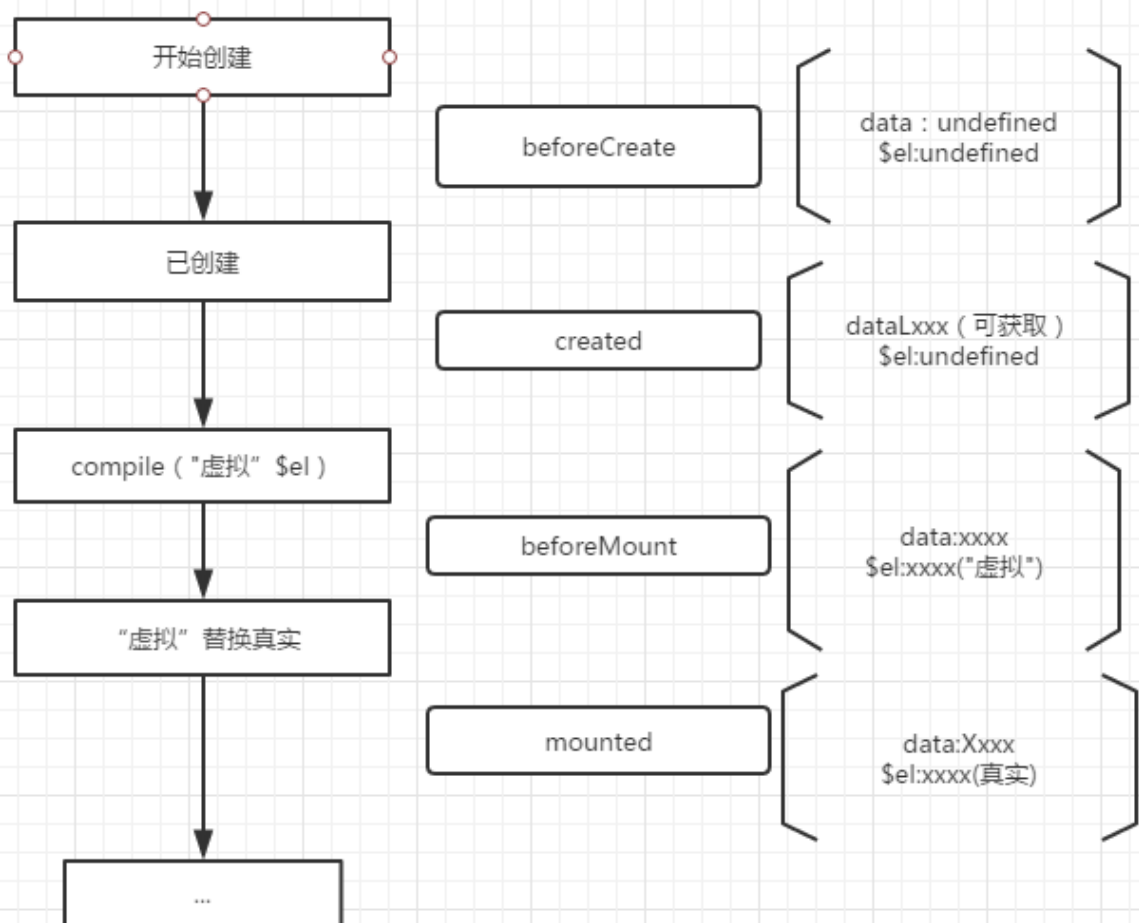
触发beforeUpdate 和updated

接着，执行myVue.\$destroy()

aaaaa



总结一下，对官方文档的那张图简化一下，就得到了这张图



文章中若有错误请指出，转载请注明出处，谢谢~

所以图中每个阶段都有一个描述，一般就是在mounted阶段做ajax，或者配合路由钩子做一些事情。

created阶段做一些初始化，实现函数自执行

beforeDestory阶段做confirm

进行加载、摧毁前询问时等场景使用。

beforeCreate 给个loading界面 created撤销loading

beforeDestory 你确认删除XX吗？ destoryed当前组件已被删除，清空相关内容