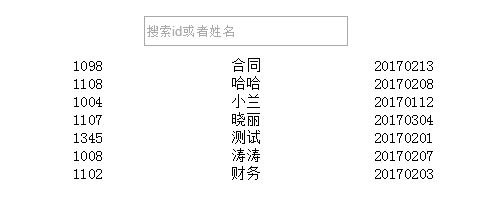
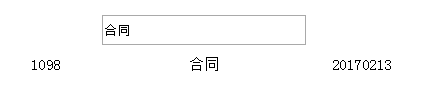
# 过滤器

## 一、[filterBy,orderBy实现搜索筛选，降序排序实例](https://segmentfault.com/a/1190000008304064)





当在输入框中输入'合同'，得到结果如图:



# 初始化方法

## 二、Vue生命周期中mounted和created的区别

转载：<https://blog.csdn.net/xdnloveme/article/details/78035065>

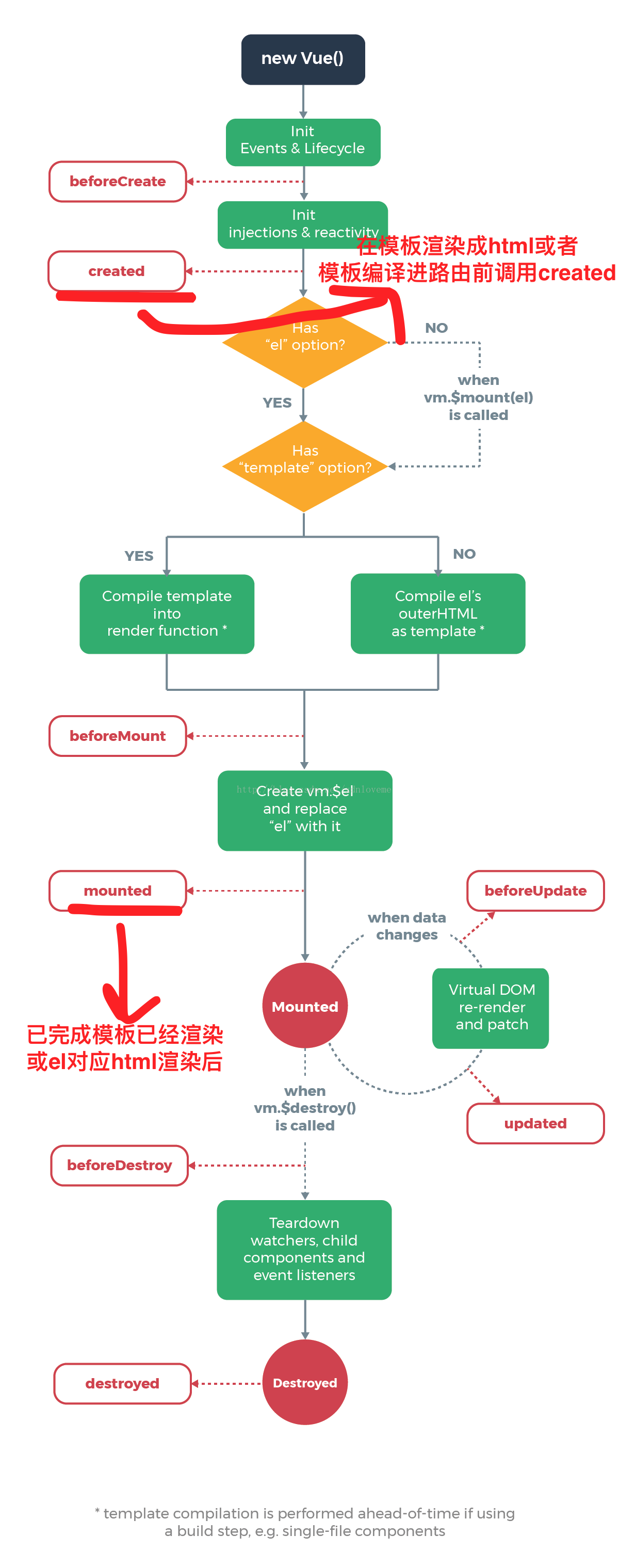
## 一、什么是生命周期？

用通俗的语言来说，就是Vue中实例或者组件从创建到消灭中间经过的一系列过程。虽然不太严谨，但是也基本上可以理解。

通过一系列实践，现在把所有遇到的问题整理一遍，今天记录一下created和mounted的区别：

## 二、created和mounted区别？

官方图解如下:



我们从图中看两个节点：

created:在模板渲染成html前调用，即通常初始化某些属性值，然后再渲染成视图。

mounted:在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。

其实两者比较好理解，通常created使用的次数多，而mounted通常是在一些插件的使用或者组件的使用中进行操作，比如插件chart.js的使用: var ctx = document.getElementById(ID);通常会有这一步，而如果你写入组件中，你会发现在created中无法对chart进行一些初始化配置，一定要等这个html渲染完后才可以进行，那么mounted就是不二之选。下面看一个例子（用组件）。

三、例子

Vue.component("demo1",{

data:function(){

return {

name:"",

age:"",

city:""

}

},

template:"<ul><li id='name'>{{name}}</li><li>{{age}}</li><li>{{city}}</li></ul>",

created:function(){

this.name="唐浩益"

this.age = "12"

this.city ="杭州"

var x = document.getElementById("name")//第一个命令台错误

console.log(x.innerHTML);

},

mounted:function(){

var x = document.getElementById("name")//第二个命令台输出的结果

console.log(x.innerHTML);

}

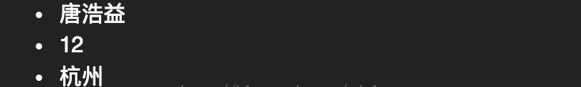
});

var vm = new Vue({

el:"#example1"

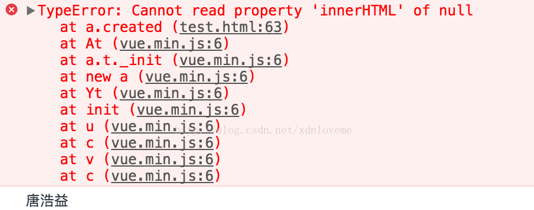
})

可以看到输出如下：



可以看到都在created赋予初始值的情况下成功渲染出来了。

但是同时看console台如下：



可以看到第一个报了错，实际是因为找不到id，getElementById(ID) 并没有找到元素，原因如下：

在created的时候，视图中的html并没有渲染出来，所以此时如果直接去操作html的dom节点，一定找不到相关的元素。

而在mounted中，由于此时html已经渲染出来了，所以可以直接操作dom节点，故输出了结果“唐浩益”。

以上就是我自己总结的mounted和mounted的区别，写的比较简陋，记录下来，加深印象。