1/5 Vue的生命周期⭐、vue/cli安装

Vue的生命周期（钩子函数）（面试⭐⭐⭐）

重点：干啥的，什么时候用

Vue的生命周期就是一个Vue组件从出生到死亡的过程。

生命周期，你需要知道它是什么时候触发。

所有的Vue应用都是从这里开始的，当实例化出Vue对象时就已经进入了Vue的生命周期。

进入的生命周期第一个钩子函数就是**beforeCreate**。在这之前组件还没有真正的初始化。

在**beforeCreate**之后，Vue对data对象作了getter/setter处理，并且将对象放在一个Observe里面以便于监控对象，另外还有使用initEvents绑定事件

在组件初始化完成后，遇到第二个钩子函数：**created**。在这个阶段我们可以访问到了data的属性以及绑定的事件

通过了**created**阶段后组件的生命周期到了**beforemount**，在这个阶段DOM结构还没有生成，但是已经创建了el，组件挂载的根节点。在**beforemount**执行完成后开始渲染DOM，执行_render方法，_mount方法，然后会有new出一个watcher对象,形成VNode节点，然后会更新DOM

渲染完毕后组件就会到了下一个生命周期**mounted**，一般的异步请求都会写在这，这个阶段DOM已经渲染出来了。至此一个组件已经完整的出现在眼前了，但是生命周期却还没有停止。

当组件需要更新的时候生命周期会先到达**beforeUpdate**,在这个阶段显示数据并没有更新，但是DOM中的数据已经改变了，这是因为双向绑定的关系

走过**beforeUpdate**组件完成了更新，生命周期走到**updated**

完成更新后的组件应该被销毁了，**beforeDestroy**，这个阶段组件还没有被销毁

destroy这个是真正的销毁

各个生命周期：

第一个生命周期 `beforeCreate` ：在这之前组件还未真正的初始化
第二个生命周期 `created` ：可以访问data的属性以及绑定事件
第三个生命周期 `beforemount` ：DOM未生成，创建el，组件挂在根节点。执行完后开始渲染DOM

第四个生命周期`mounted`：异步请求写在这，组件完整呈现
第五个生命周期`beforeUpdate`：数据未更新，DOM数据已改变，因为双向绑定。
第六个生命周期`updated`：完成更新后，组件应被销毁，但`beforeDestroy`组件未被销毁
第七个生命周期`destroy`：销毁

各个生命周期详解：（此处每项不是很清楚）

beforeCreate：在组件初始化之前，当这个生命周期中组件的数据、方法、事件都还没有，
简单来说，new Vue之后，可以理解为第一句内容之前它就调用了，（new Vue( ) 瞬间它就直接调用了，）

 **created：**当数据、方法、事件初始化之后调用。简单点说，当数据有初始值之后调用。
一般都是在请求数据时用，关闭loading。
要么看看有没有el属性，没有就再看看实例下有没有，$mount(el)

created：

- 是Vue数据初始化之后，调用的函数。（是new Vue( )下面的一个属性），里面的this指向 Vue 。
- 当要请求数据的时候，就走created。

beforeMount：渲染之前，可以关掉loding

 **monted：**DOM渲染之后，获取页面的元素，主要是数据生成出来之前的元素。
当DOM 更新时触发，只要操作DOM直接使用

angular：脏值检查

beforeUpdate：数据更新之前

updated：数据更新之后其实直接使用computed

beforeDestroy：死亡前关掉定时器、状态的初始化

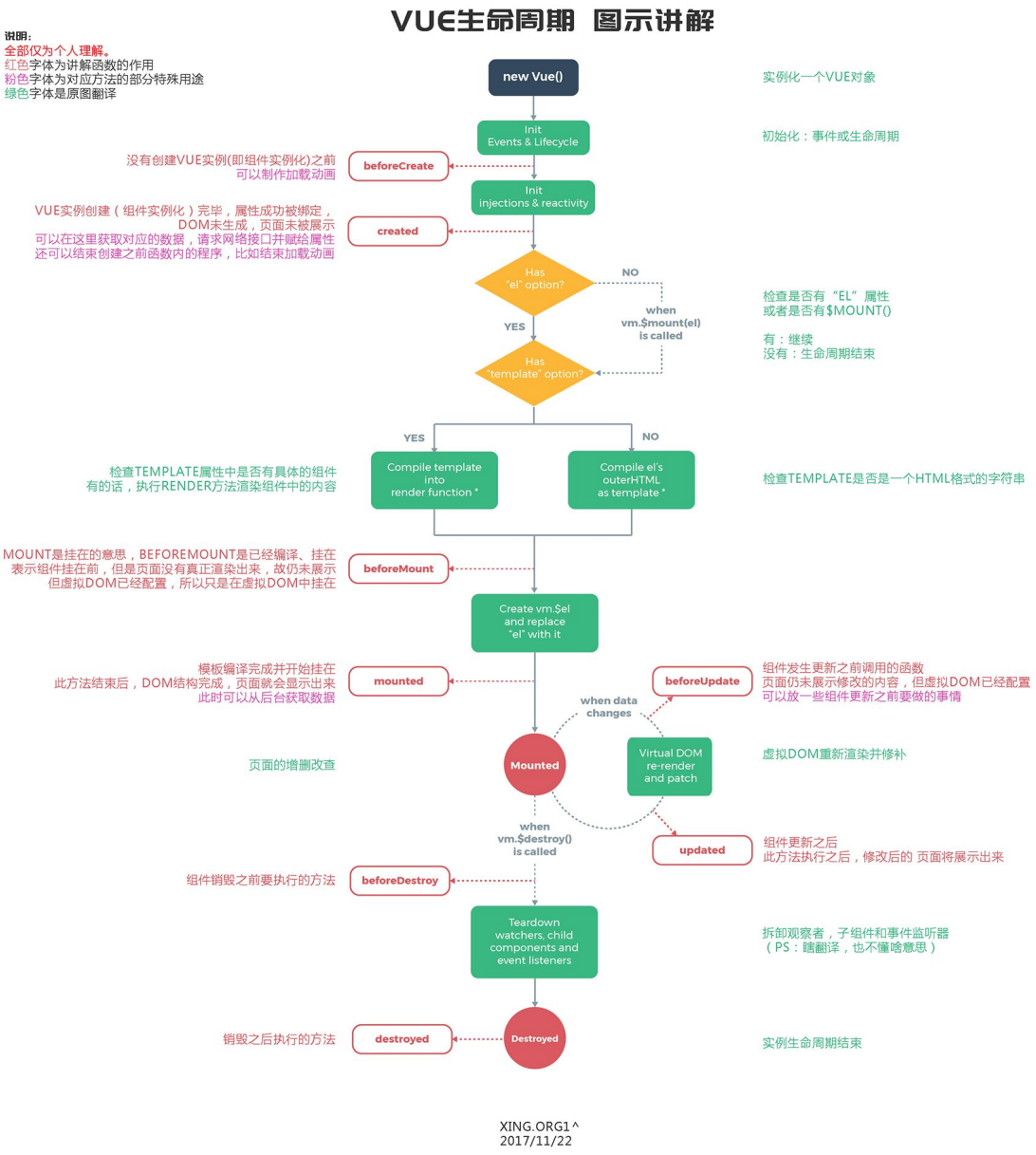
destroyed：没有路由的时候是人为手动触发，有路由，切换路由的时候上一个

> **template**属性会覆盖根节点
>
> - 1、template：`<div></div>`
> - 2、

```
<template id="idom1">
</template>
```

> 有模板走模板没有模板走根目录的innerhtml

他人的理解：



Vue-cli

vue-cli：官方配置的关于使用vue的各种功能，这种功能叫做脚手架。（和npm 一样的功能。）

1、安装：(只需要安装一次)

- npm install yarn
- 或者下面的安装方式
- yarn global add @vue/cli

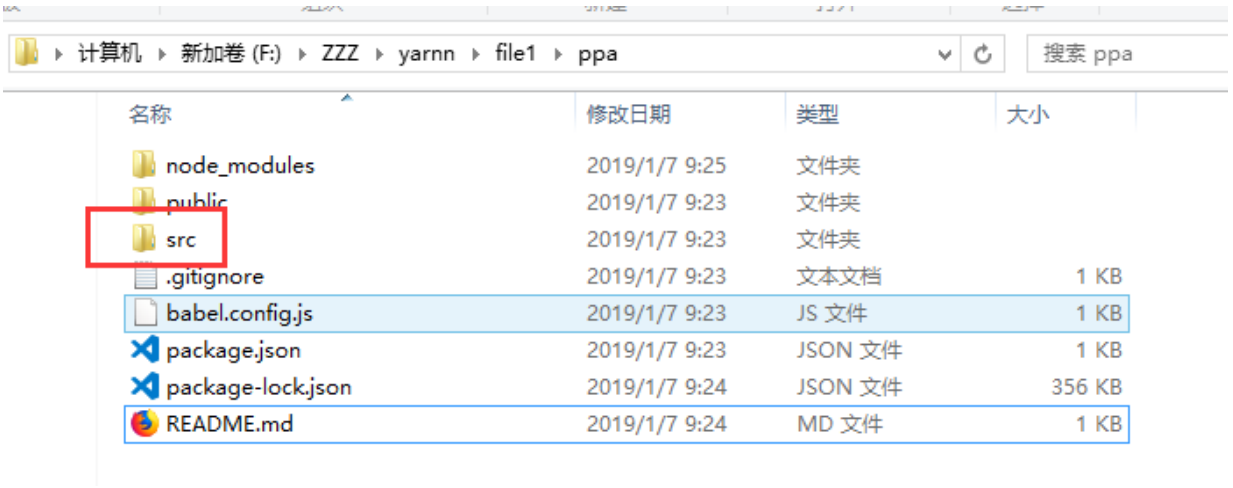
2、创建项目：

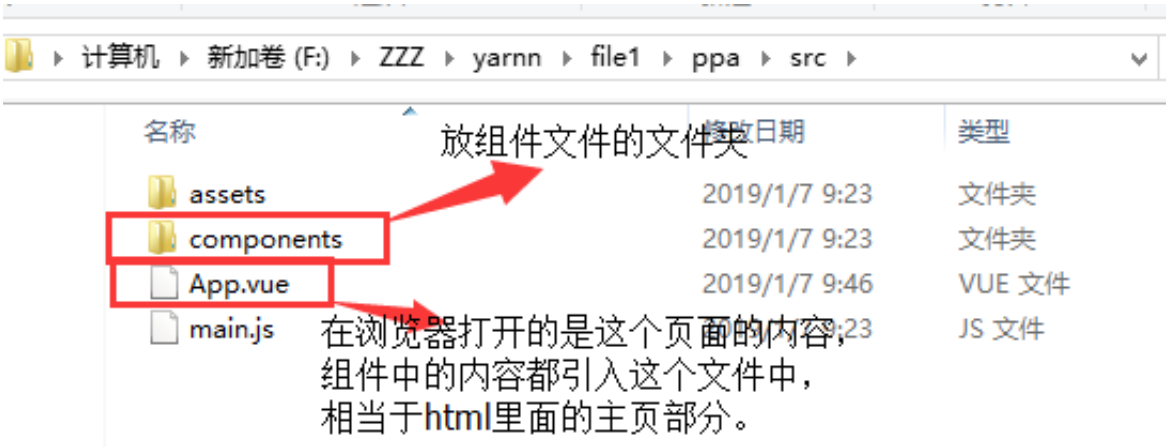
- vue create xxx（项目名称）

3、千万不要走default配置（不然会有esLint，一个不小心就会出现语法错误）使用上下键切换配置（Manually select features），

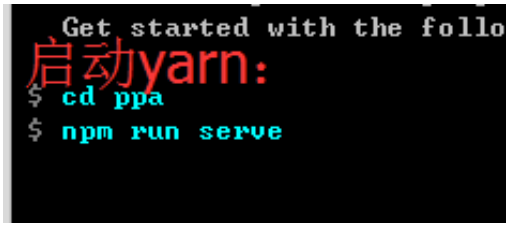
4、选择配置项， 按空格选中

创建项目后会出现以下几个文件：





每次运行都需要在项目文件中：启动yarn



App . vue文件 和组件文件 中主要分为三部分：

- template –> 代表的是html部分

- script部分–> 代表的是JS部分
- 还有style部分 –> 代表的是CSS部分。

```
<!--这是App.vue文件 和组件文件 中的大致内容：-->

<template>
  <div id="app">
    <div class="active">你好</div>
    <hr />
    <List></List>  // 使用组件
    <TextInput />  // 使用组件
  </div>
</template>

<script>
import List from './components/list';  // 引入组件
import TextInput from './components/text'; // 引入组件
export default {
  components:{
    List,
    TextInput
  }
}
</script>

<style scoped>  /* 设置CSS样式 scoped的作用是它的样式只作用于当前模块，实现的样式的私有化，注意：有可能调试公共样式的时候，有这个属性的模块不会被公共样式影响。*/
.active{
  color:red;
}
body{
  background: pink;
}
</style>


<!--
    App.vue中通过
    export default{
    }
    把这个App这个组件导出去，给min.js使用

    引入组件：
    1、import 引入
    2、components：{
        List: List,
        List
```

```
        }
        3、把组件放到template中

    -->
```