Vue预习课

核心知识05——生命周期

每个 Vue 实例在被创建时都要经过一系列的初始化过程——例如，需要设置数据监听、编译模板、将实例挂载到 DOM 并在数据变化时更新 DOM 等，称为Vue实例的生命周期。

使用生命周期钩子

在Vue实例的生命周期过程中会运行一些叫做**生命周期钩子**的函数，这给用户在不同阶段添加自己代码的机会。

范例：异步获取列表数据

```
// 模拟异步数据调用接口
function getCourses() {
    return new Promise(resolve => {
        setTimeout(() => {
            resolve(['web全栈', 'web高级'])
        }, 2000);
    })
}
const app = new Vue({
    // created钩子中调用接口
    async created() {
        const courses = await getCourses()
        this.courses = courses
    },
}
```

> created还是mounted?

探讨生命周期

从一道面试题开始

关于Vue的生命周期，下列哪项是不正确的？()[单选题]
A、Vue 实例从创建到销毁的过程，就是生命周期。
B、页面首次加载会触发beforeCreate，created，beforeMount，mounted，beforeUpdate，updated。
C、created表示完成数据观测，属性和方法的运算，初始化事件，$el属性还没有显示出来。
D、DOM渲染在mounted中就已经完成了。

测试代码

```
<!DOCTYPE html>
<html>
<head>
    <title>Vue源码剖析</title>
    <script src="vue.js"></script>
</head>

<body>
    <div id="demo">
        <h1>初始化流程</h1>
        <p>{{foo}}</p>
    </div>
    <script>
        // 创建实例
        const app = new Vue({
            el: '#demo',
            data:{
                foo: 'foo'
            },
            beforeCreate(){console.log('beforeCreate')},
            created(){console.log('created '+this.$el)},
            beforeMount(){console.log('beforeMount')},
            mounted(){
                setTimeout(() => {
                    this.foo = 'fooooo'
                }, 2000);
                console.log('mounted '+this.$el)},
            beforeUpdate(){console.log('beforeUpdate')},
            updated(){console.log('updated')},
        });
    </script>
</body>
</html>
```

生命周期图示

生命周期图示、生命周期列表

结论：

- 三个阶段：初始化、更新、销毁
- 初始化：beforeCreate、created、beforeMount、mounted
- 更新：beforeUpdate、updated
- 销毁：beforeDestroy、destroyed

使用场景分析

```
{
    beforeCreate(){} // 执行时组件实例还未创建，通常用于插件开发中执行一些初始化任务
    created(){} // 组件初始化完毕，各种数据可以使用，常用于异步数据获取
    beforeMounted(){} // 未执行渲染、更新，dom未创建
    mounted(){} // 初始化结束，dom已创建，可用于获取访问数据和dom元素
    beforeUpdate(){} // 更新前，可用于获取更新前各种状态
    updated(){} // 更新后，所有状态已是最新
    beforeDestroy(){} // 销毁前，可用于一些定时器或订阅的取消
    destroyed(){} // 组件已销毁，作用同上
}
```