

Vue组件化实现机制

测试代码

```
<div id="demo">
  <h1>Vue组件化机制</h1>
  <comp></comp>
</div>

<script>
  Vue.component('comp', { template: '<div>I am comp</div>' })
  new Vue({ el: '#demo' })
</script>
```

观察渲染函数

```
(function anonymous(
) {
  with(this){return _c('div',{attrs:{"id":"demo"}},[
    _c('h1',[_v("Vue组件化机制")]),_v(" "),
    _c('comp')],1)}
})
```

_createElement

组件声明

Vue.component()

initAssetRegisters(Vue) `src/core/global-api/assets.js`

1. 转化组件配置对象为构造函数
2. 注册到Vue的options.components中

组件创建及挂载

整体流程

首先创建的是根组件，首次_render()时，会得到整棵树的VNode结构

整体流程：new Vue() => \$mount() => vm._render() => createElement() => createComponent()

创建自定义组件VNode

_createElement src\core\vdom\create-element.js

_createElement实际执行VNode创建的函数，由于传入tag是非保留标签，因此判定为自定义组件通过createComponent去创建

createComponent src\core\vdom\create-component.js

创建组件VNode，保存了上一步处理得到的组件构造函数，props，事件等

创建自定义组件实例

根组件执行更新函数时_update()，会递归创建子组件和子元素实例

createEle() core\vdom\patch.js

首次执行_update()时，patch()会通过createEle()创建根元素，子元素创建研究从这里开始

createComponent core\vdom\patch.js

自定义组件创建、挂载