

## el的两种写法

---

### new Vue 时候配置el属性

默认写法

```
<script>
  // el 的两种写法
  const v = new Vue({
    // el:'#a',           /第一种写法
    data:{
      name:"我的名字"
    }
  })
</script>
```

### 通过 vm.\$mount('#a') 执行

先创建 Vue实例，然后在通过 vm.\$mount('#a') 执行el的值

```
<script>
  // el 的两种写法
  const v = new Vue({
    // el:'#a',           /第一种写法
    data:{
      name:"我的名字"
    }
  })
  v.$mount('#a');        //第二种写法，可以外部调用，mount 挂载的意思
</script>
```

## data的两种写法

---

在学习组件时，data必须使用**函数式**，否则会报错

### 对象式

```
new Vue({
  el:"#a",
  // data 的第一种写法：对象式
  data:{
    name:"我的名字"
  },
})
```

# 函数式

## 第一种写法,显示function方法

```
<script>
  const x = new Vue({
    el: '#root',
    // data的第二种写法: 函数式
    // 显示function方法
    data: function() {
      return {
        name: 'YF'
      }
    }
  })
</script>
```

## 第二种写法, 不显示function

```
<script>
  const x = new Vue({
    el: '#root',
    // data的第二种写法: 函数式
    // 不显示function方法
    data() {
      return {
        name: 'YF'
      }
    }
  })
</script>
```