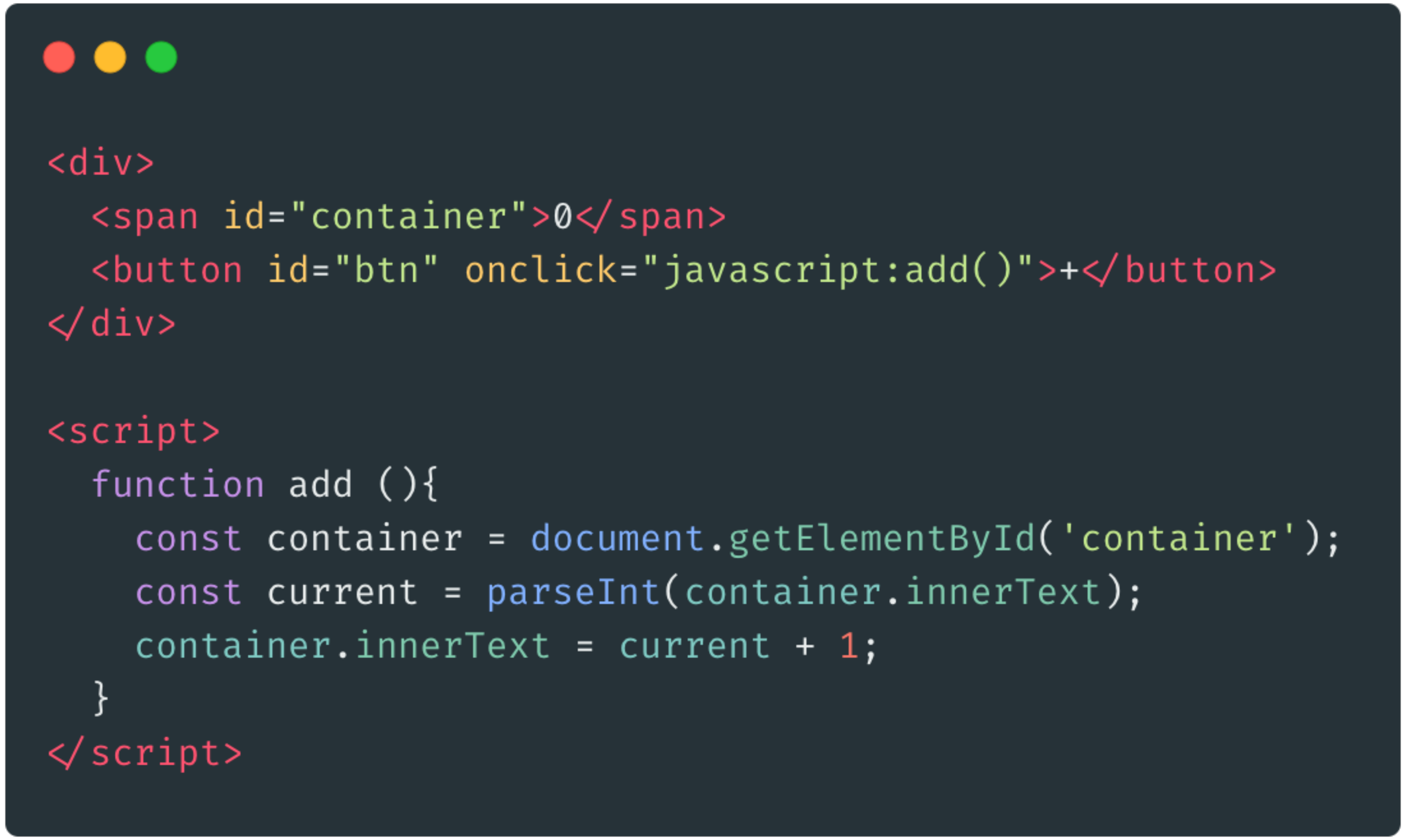
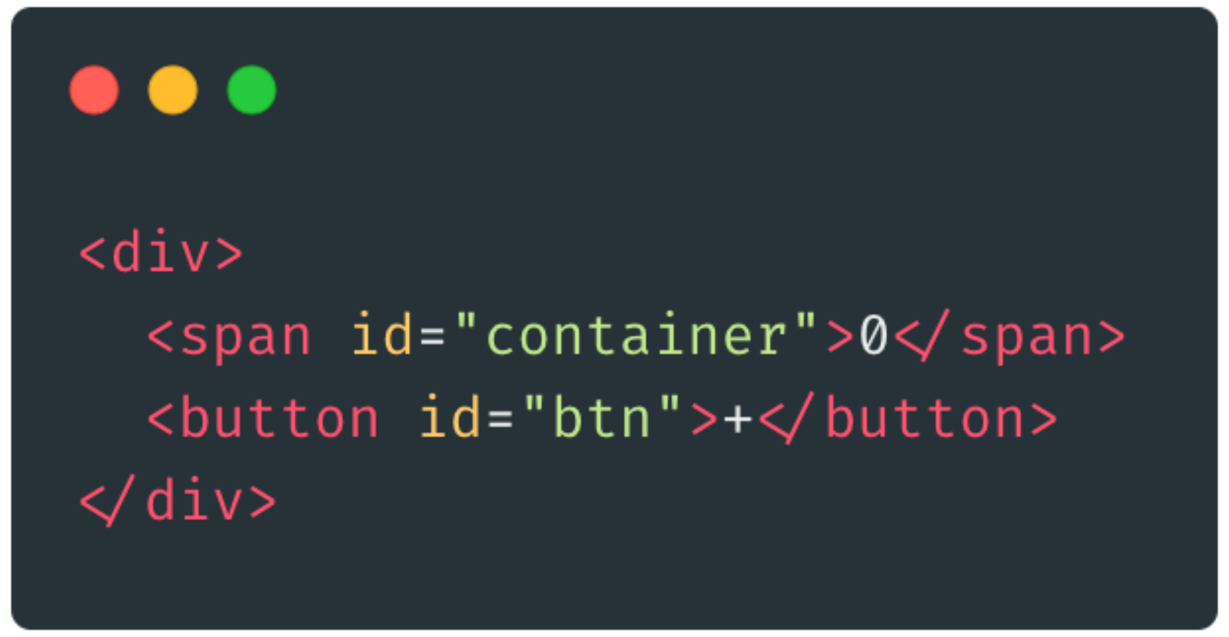
从一个简单的功能说起

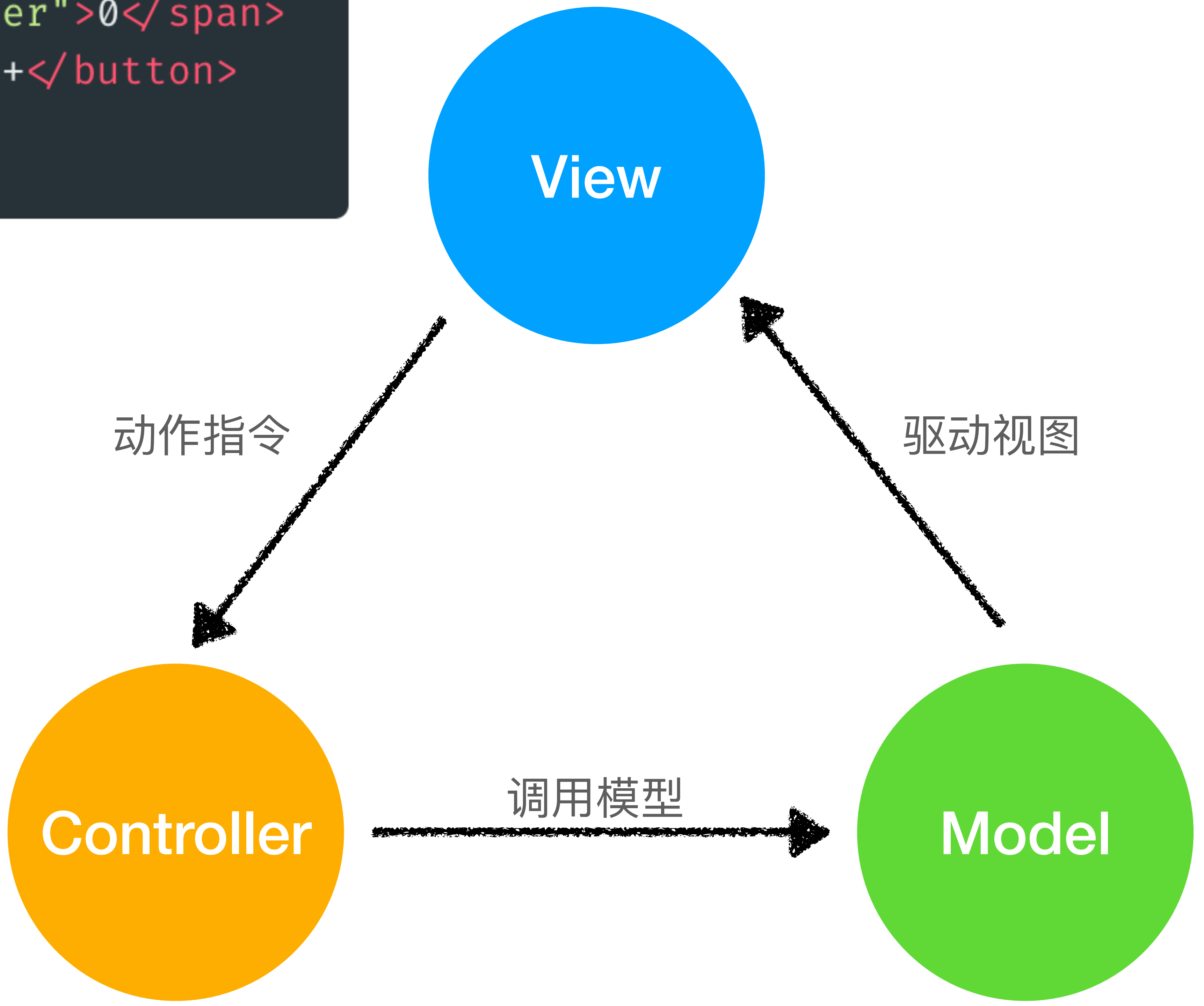


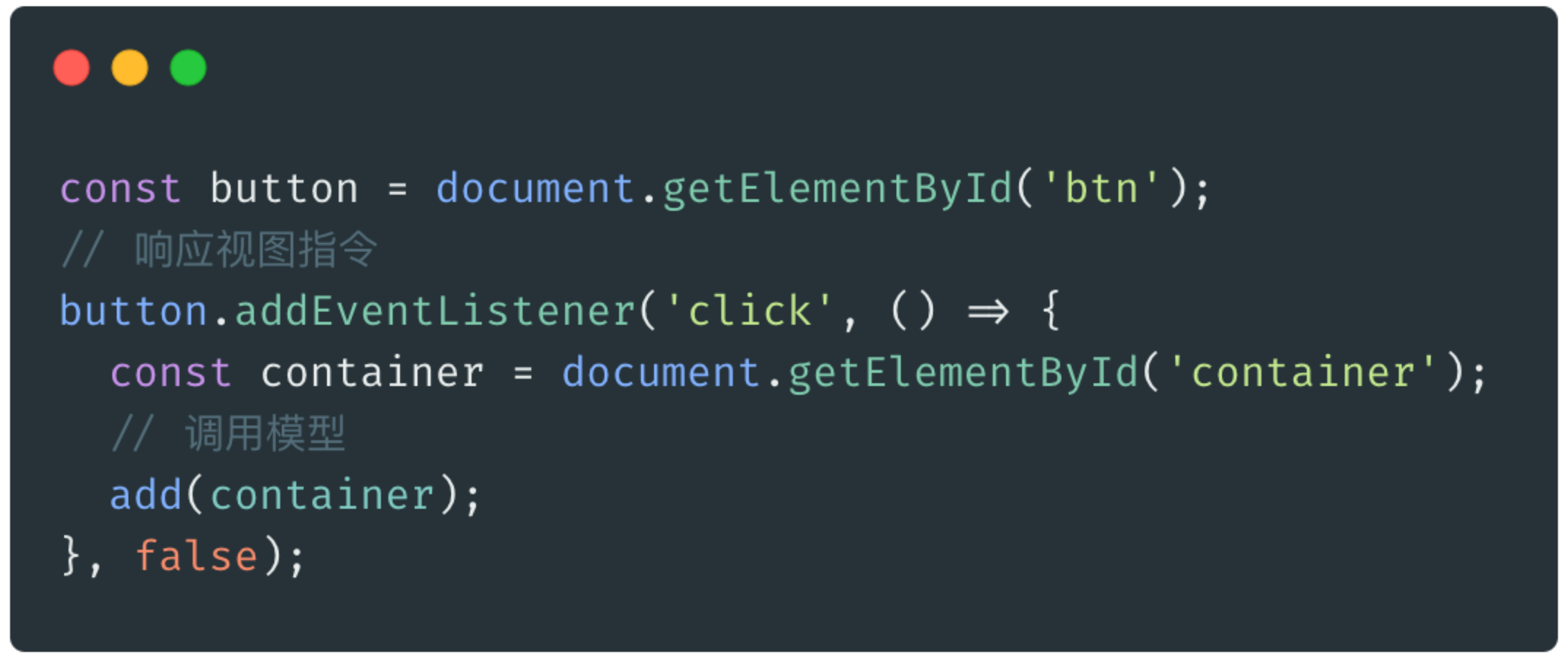
```
<div>
  <span id="container">0</span>
  <button id="btn" onclick="javascript:add()">+</button>
</div>

<script>
  function add (){
    const container = document.getElementById('container');
    const current = parseInt(container.innerText);
    container.innerText = current + 1;
  }
</script>
```

MVC框架

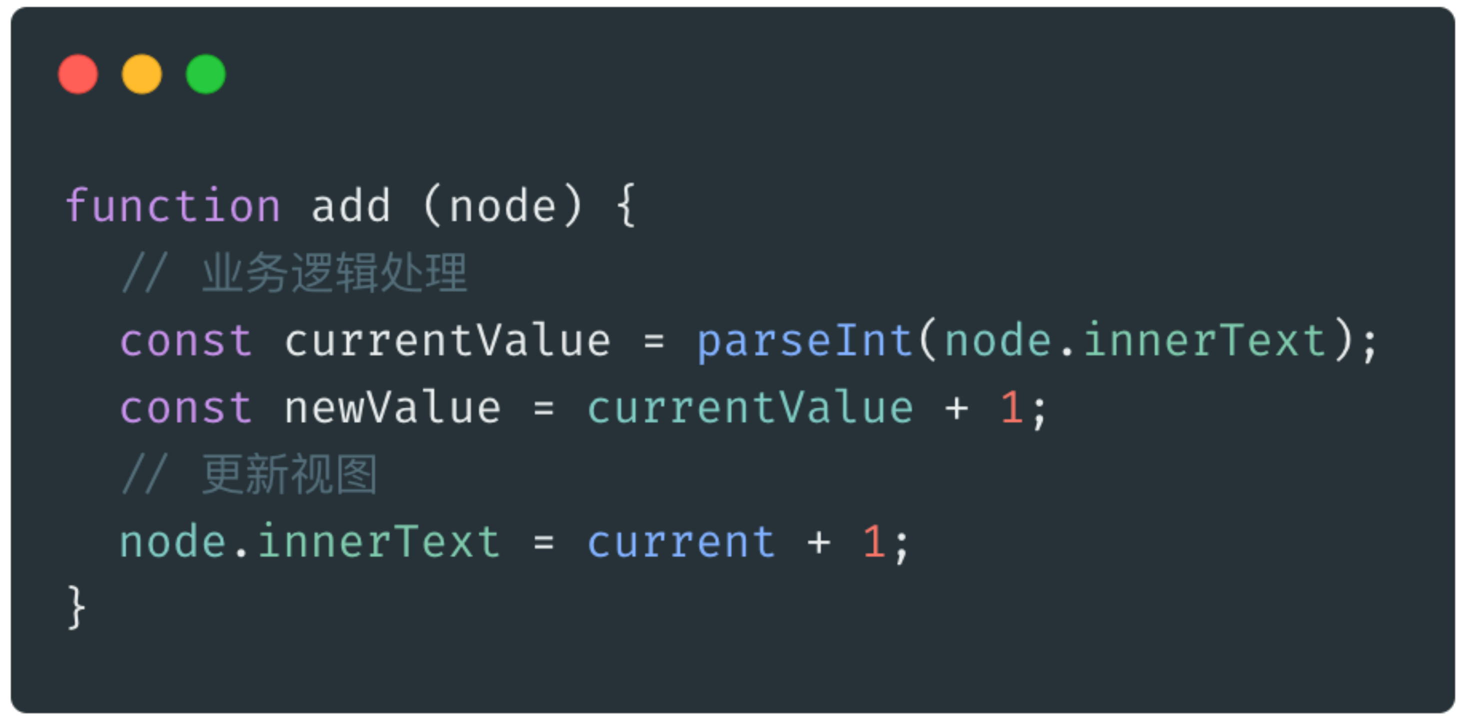


```html
<div>
  <span id="container">0</span>
  <button id="btn">+</button>
</div>
```



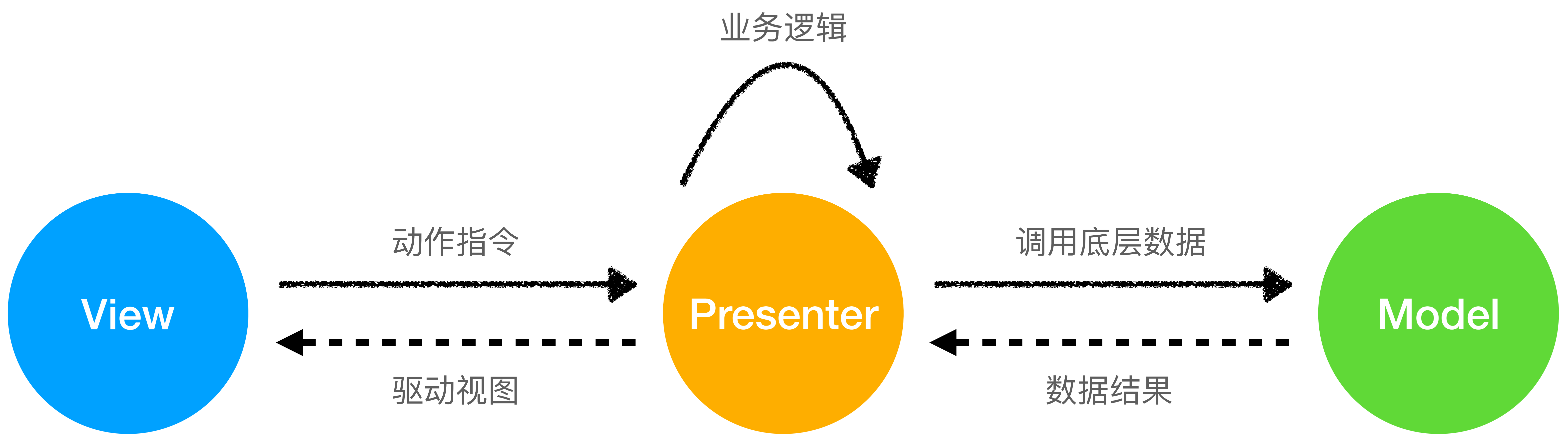


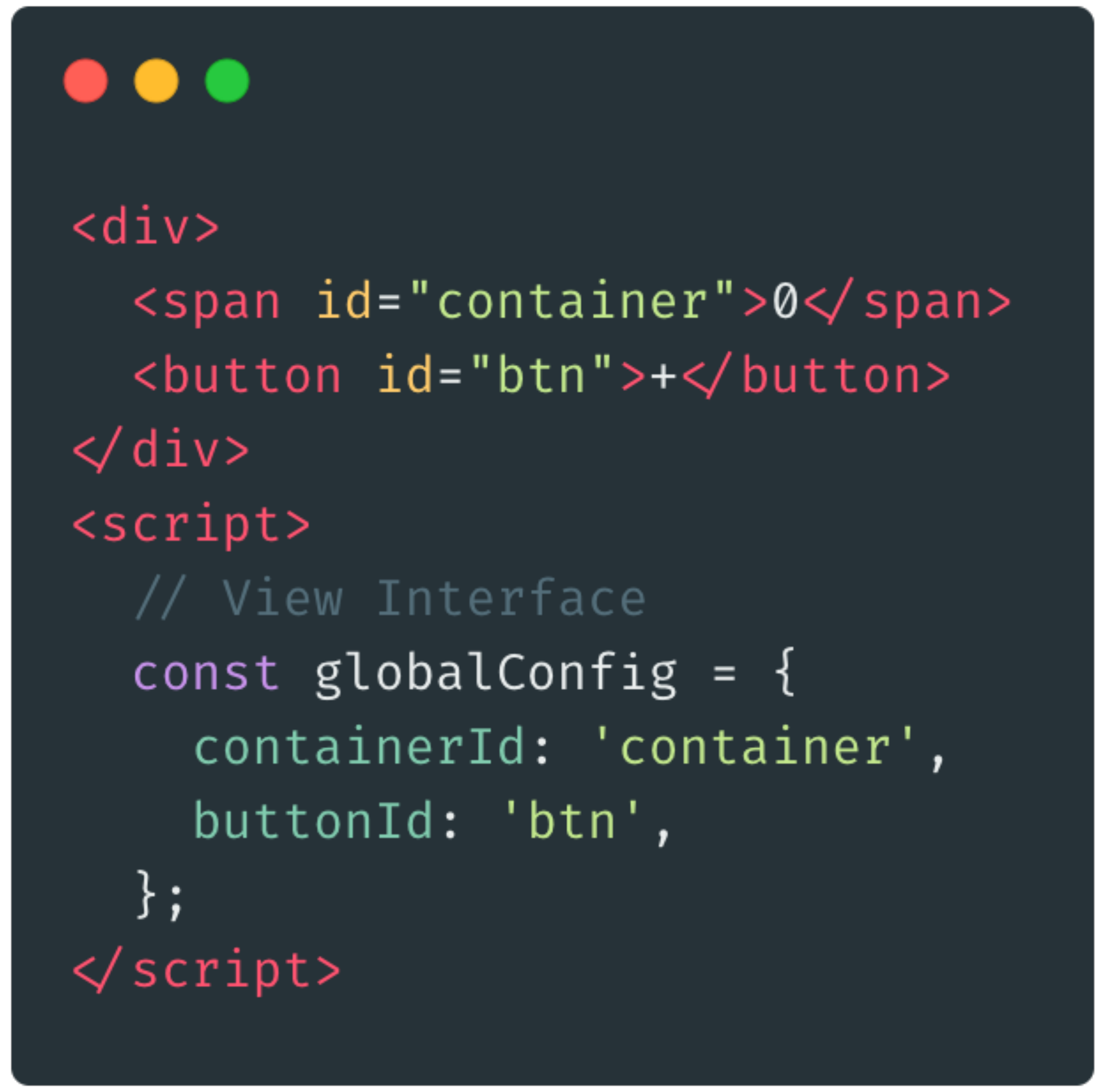
```javascript
const button = document.getElementById('btn');
// 响应视图指令
button.addEventListener('click', () => {
  const container = document.getElementById('container');
  // 调用模型
  add(container);
}, false);
```



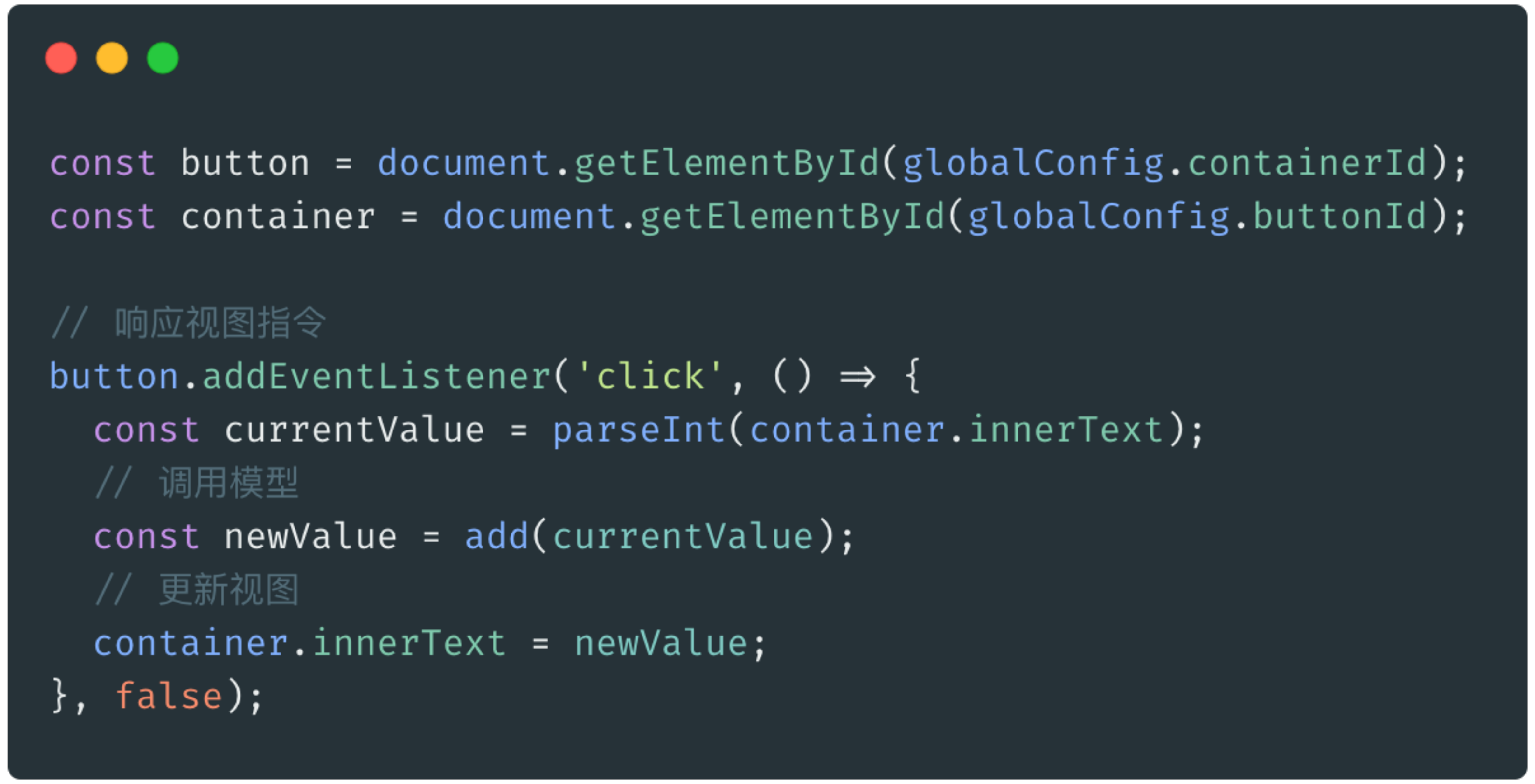
```javascript
function add (node) {
  // 业务逻辑处理
  const currentValue = parseInt(node.innerText);
  const newValue = currentValue + 1;
  // 更新视图
  node.innerText = current + 1;
}
```

MPV框架

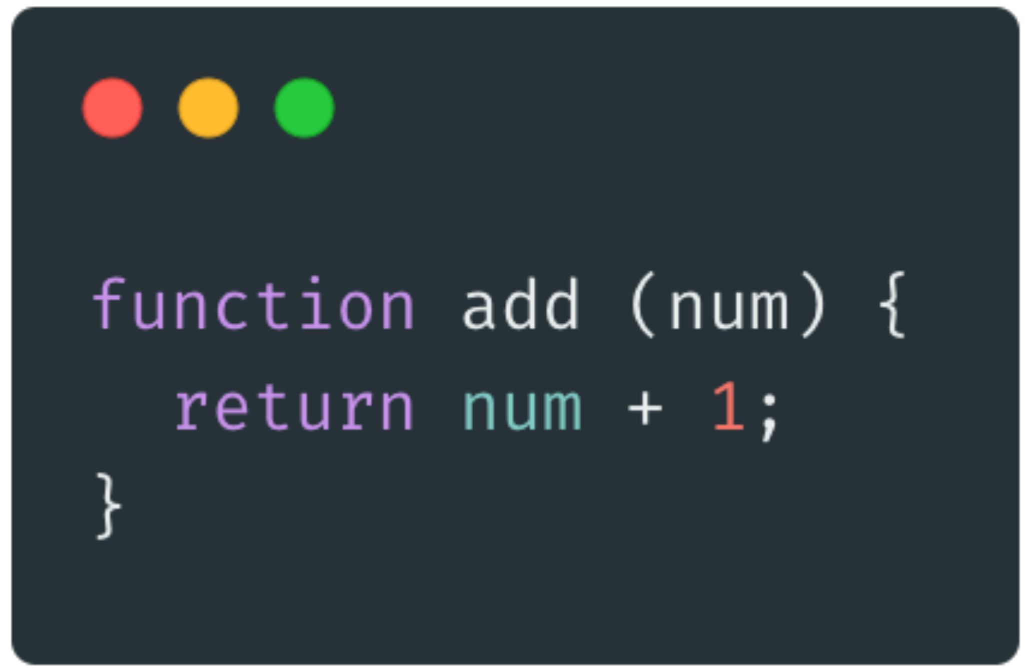




```
<div>
  <span id="container">0</span>
  <button id="btn">+</button>
</div>
<script>
  // View Interface
  const globalConfig = {
    containerId: 'container',
    buttonId: 'btn',
  };
</script>
```

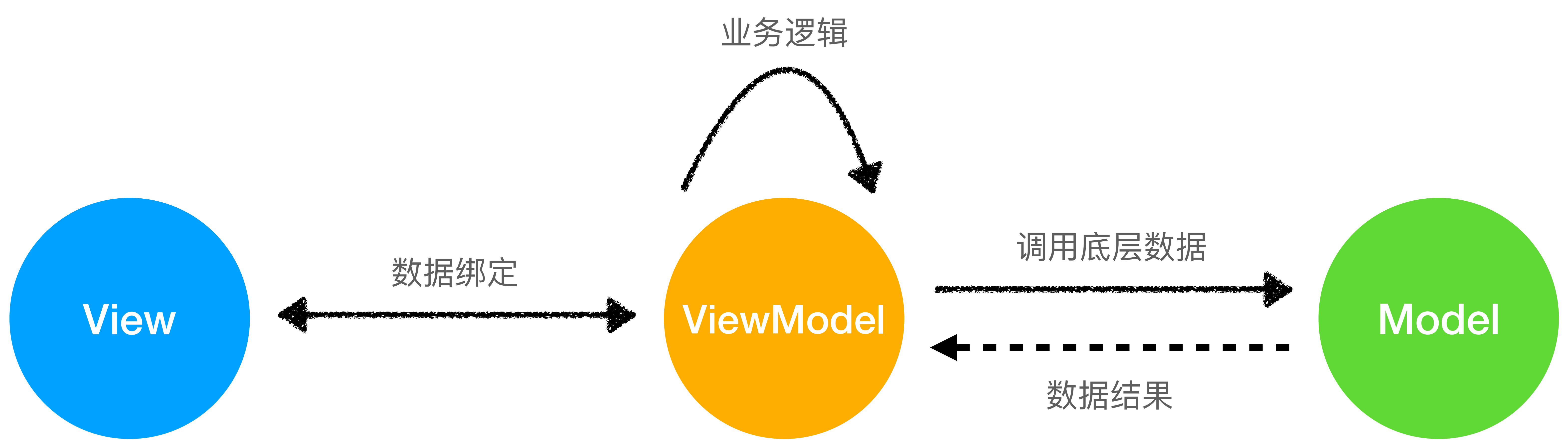


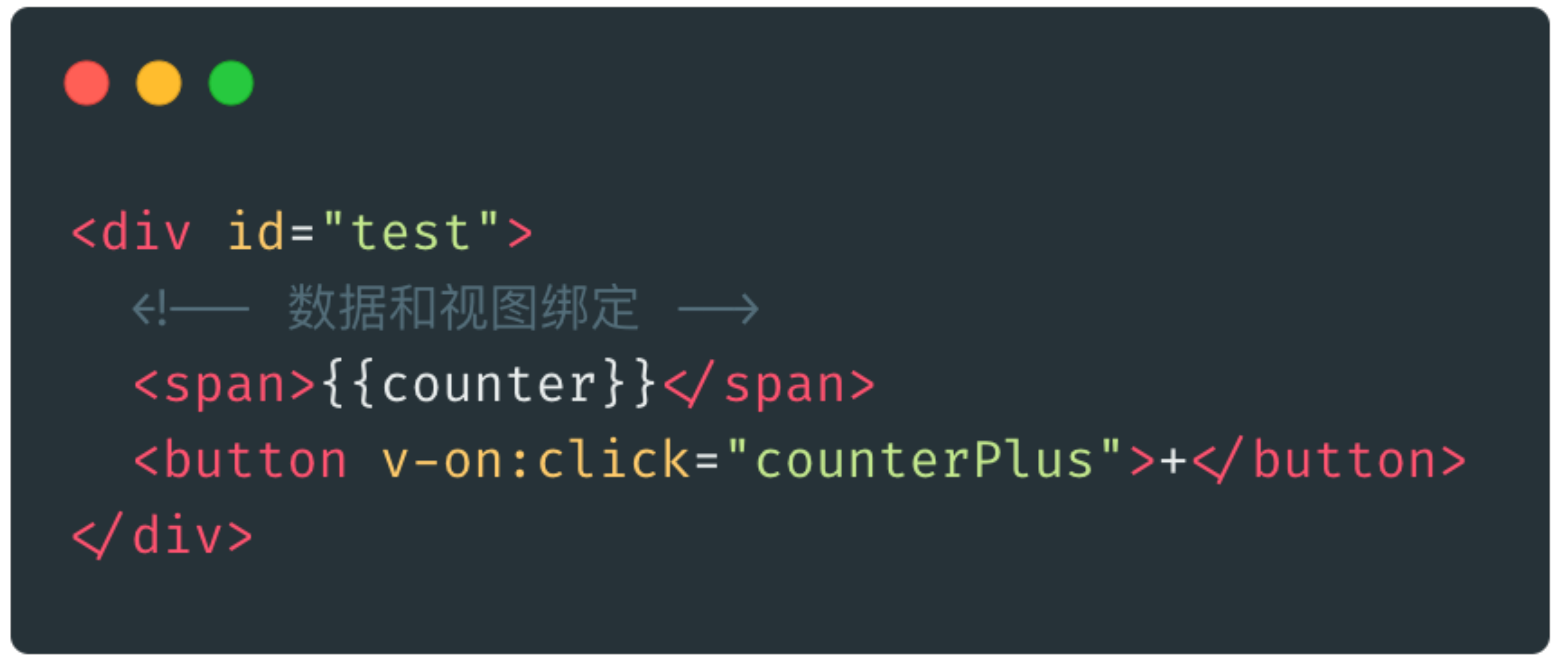
```
const button = document.getElementById(globalConfig.containerId);
const container = document.getElementById(globalConfig.buttonId);

// 响应视图指令
button.addEventListener('click', () ⇒ {
  const currentValue = parseInt(container.innerText);
  // 调用模型
  const newValue = add(currentValue);
  // 更新视图
  container.innerText = newValue;
}, false);
```



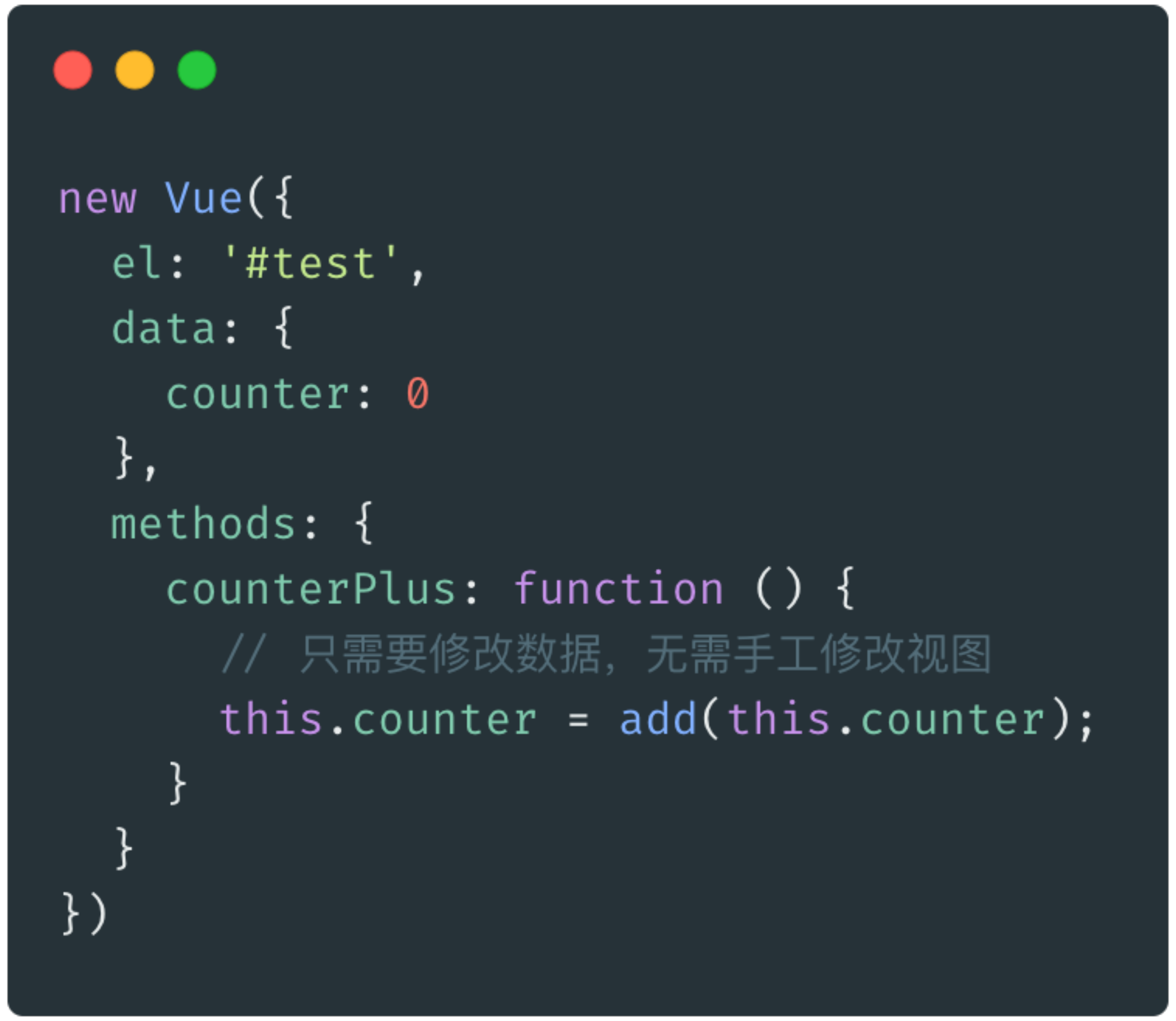
```
function add (num) {
  return num + 1;
}
```

MVVM框架

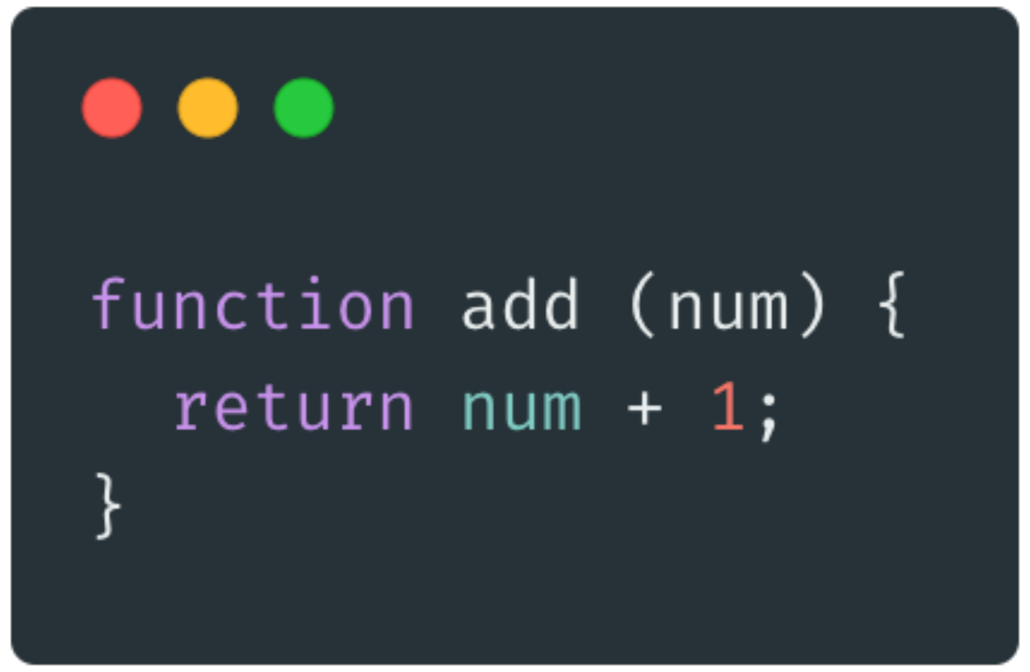




```
<div id="test">
  <!-- 数据和视图绑定 -->
  <span>{{counter}}</span>
  <button v-on:click="counterPlus">+</button>
</div>
```

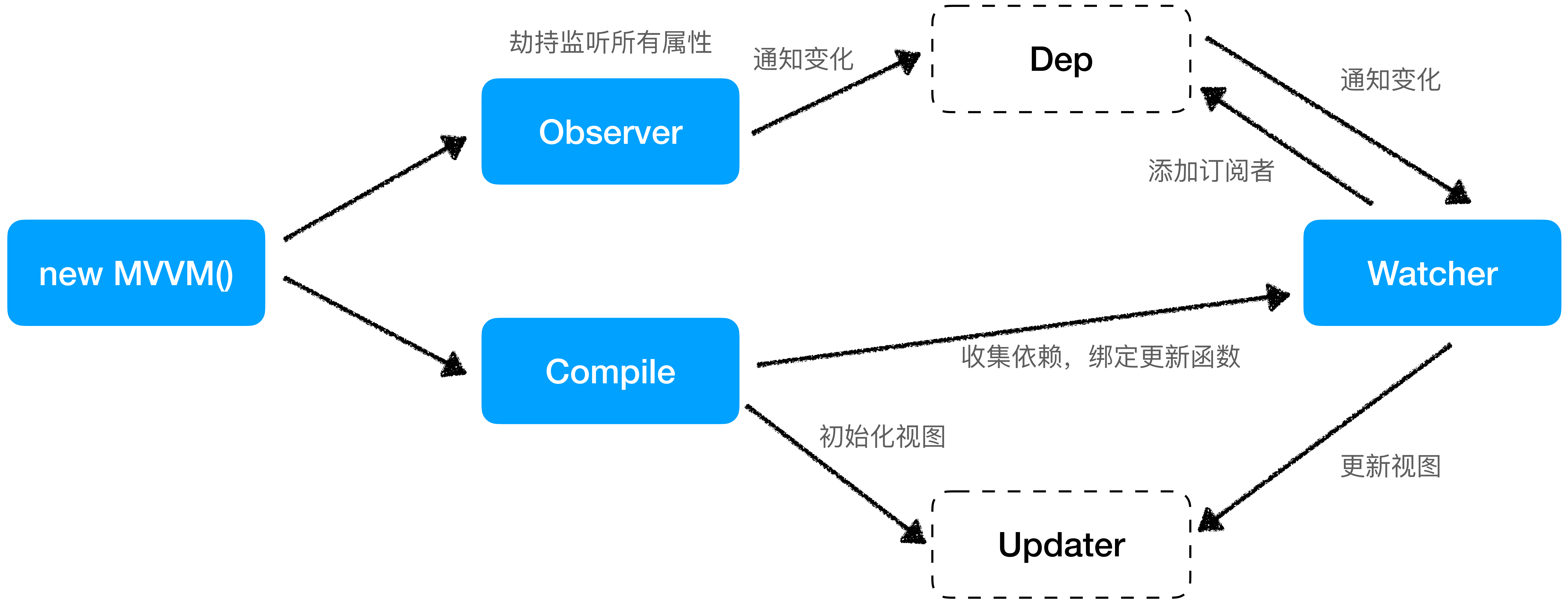


```
new Vue({
  el: '#test',
  data: {
    counter: 0
  },
  methods: {
    counterPlus: function () {
      // 只需要修改数据，无需手工修改视图
      this.counter = add(this.counter);
    }
  }
})
```



```
function add (num) {
  return num + 1;
}
```

MVVM框架ViewModel层运行机制


劫持监听所有属性
通知变化
Dep
通知变化
Observer
添加订阅者
new MVVM()
Watcher
Compile
收集依赖，绑定更新函数
初始化视图
更新视图
Updater