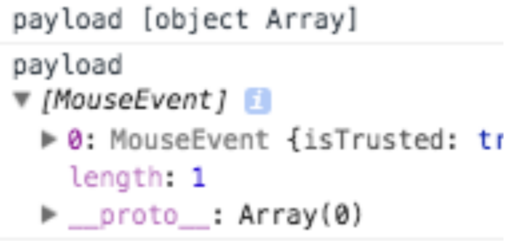
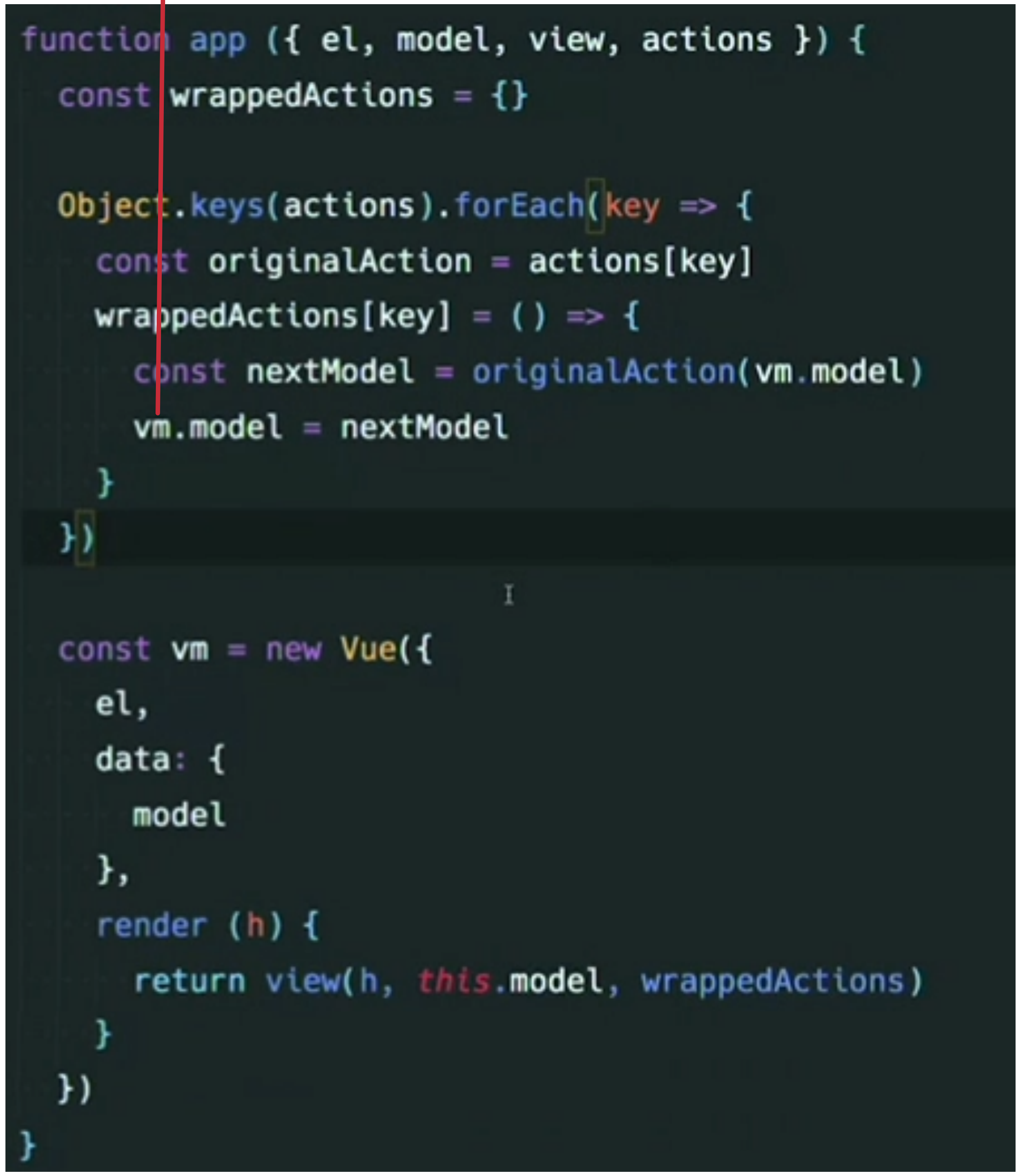
Although vm is defined below, it exist here
because wrappedActions get invoke when
user clicked



```
payload [object Array]
payload
▼ [MouseEvent] i
  ▶ 0: MouseEvent {isTrusted: tr
    length: 1
  ▶ __proto__: Array(0)
```

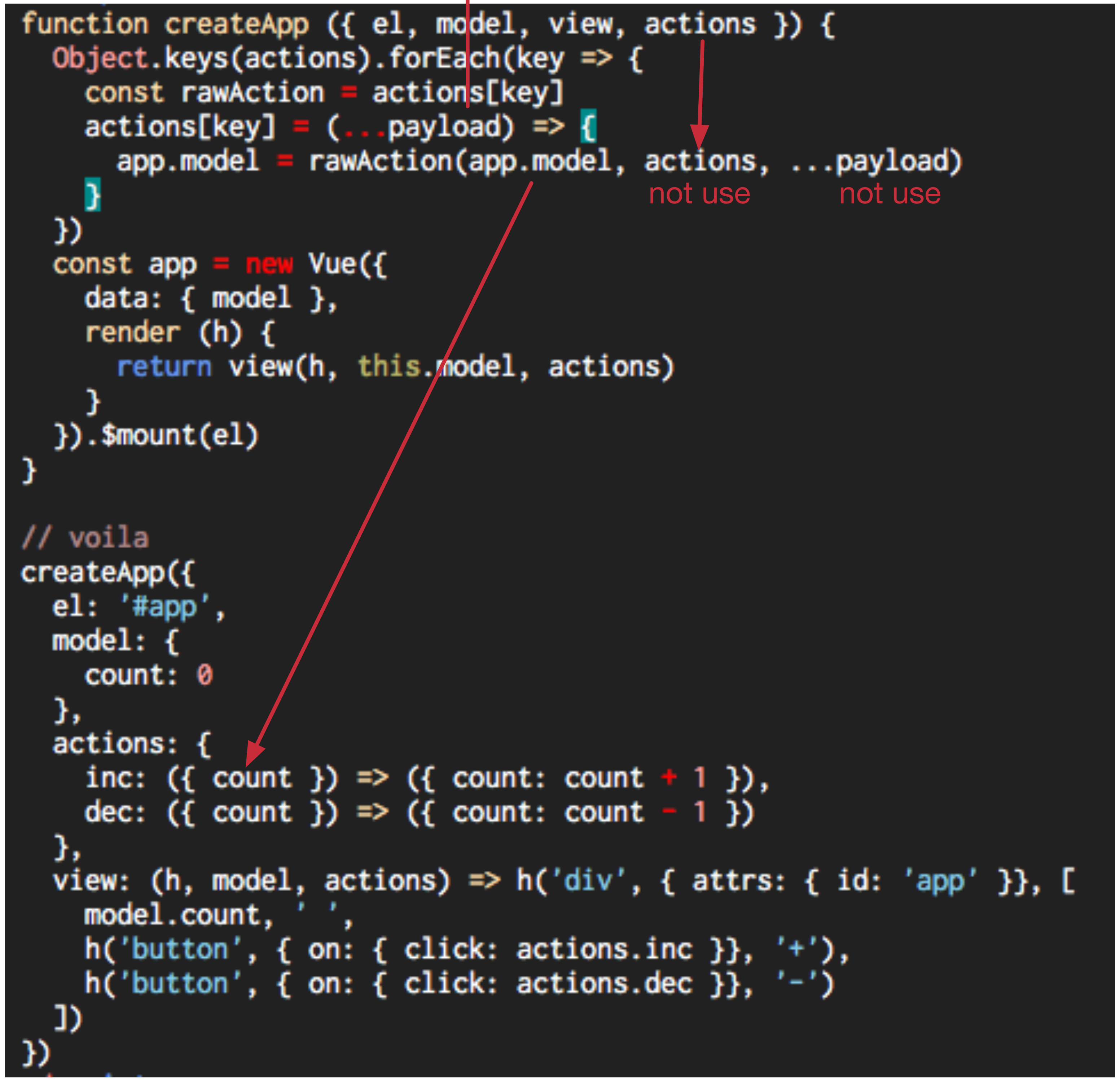


```
function app ({ el, model, view, actions }) {
  const wrappedActions = {}

  Object.keys(actions).forEach(key => {
    const originalAction = actions[key]
    wrappedActions[key] = () => {
      const nextModel = originalAction(vm.model)
      vm.model = nextModel
    }
  })

  const vm = new Vue({
    el,
    data: {
      model
    },
    render (h) {
      return view(h, this.model, wrappedActions)
    }
  })
}
```

Above code in video does not exist
in downloaded lesson's repo



```
function createApp ({ el, model, view, actions }) {
  Object.keys(actions).forEach(key => {
    const rawAction = actions[key]
    actions[key] = (...payload) => {
      app.model = rawAction(app.model, actions, ...payload)
    }
  })
  const app = new Vue({
    data: { model },
    render (h) {
      return view(h, this.model, actions)
    }
  }).$mount(el)
}

// voila
createApp({
  el: '#app',
  model: {
    count: 0
  },
  actions: {
    inc: ({ count }) => ({ count: count + 1 }),
    dec: ({ count }) => ({ count: count - 1 })
  },
  view: (h, model, actions) => h('div', { attrs: { id: 'app' }}, [
    model.count, ' ',
    h('button', { on: { click: actions.inc }}, '+'),
    h('button', { on: { click: actions.dec }}, '-')
  ])
})
```

not use

not use