

vue - watch 数组或者对象

作者: [maixiaojie](#)

原文链接: <https://ld246.com/article/1559096174414>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



很多时候我们需要监听数据，可能是路由，类型可能会是字符串，数组，对象，整理了下几种监听的方式。

普通的watch

```
data() {
  return {
    frontPoints: 0
  }
},
watch: {
  frontPoints(newValue, oldValue) {
    console.log(newValue)
  }
}
```

数组的watch

```
data() {
  return {
    winChips: new Array(11).fill(0)
  }
},
watch: {
  winChips: {
    handler(newValue, oldValue) {
      for (let i = 0; i < newValue.length; i++) {
        if (oldValue[i] !== newValue[i]) {
          console.log(newValue)
        }
      }
    }
  }
}
```

```
    }
  },
  deep: true
}
}
```

对象的watch

```
data() {
  return {
    bet: {
      pokerState: 53,
      pokerHistory: 'local'
    }
  }
},
watch: {
  bet: {
    handler(newValue, oldValue) {
      console.log(newValue)
    },
    deep: true
  }
}
```

只要bet中的属性发生变化（可被监测到的），便会执行handler函数；

如果想监测具体的属性变化，如pokerHistory变化时，才执行handler函数，则可以利用计算属性computed做中间层。

事例如下：

对象具体属性的watch [活用computed]

```
data() {
  return {
    bet: {
      pokerState: 53,
      pokerHistory: 'local'
    }
  }
},
computed: {
  pokerHistory() {
    return this.bet.pokerHistory
  }
},
watch: {
  pokerHistory(newValue, oldValue) {
    console.log(newValue)
  }
}
```