



土巴兔
Tubatu.com

省钱 · 省心 · 省时间

VUE 组件开发实战



1

什么是组件化开发？

2

JavaScript 概念梳理

3

Vue 组件开发常见问题

什么是组件化开发？





尤雨溪

Vue.js 作者，前 Google
Creative Lab 成员。

Vue

渐进式 JavaScript 框架



土巴兔
Tubatu.com

省钱 · 省心 · 省时间



渐进式 JavaScript 框架

起步

GITHUB

易用

已经会了HTML,CSS,JavaScript? 即刻阅读指南开始构建应用!

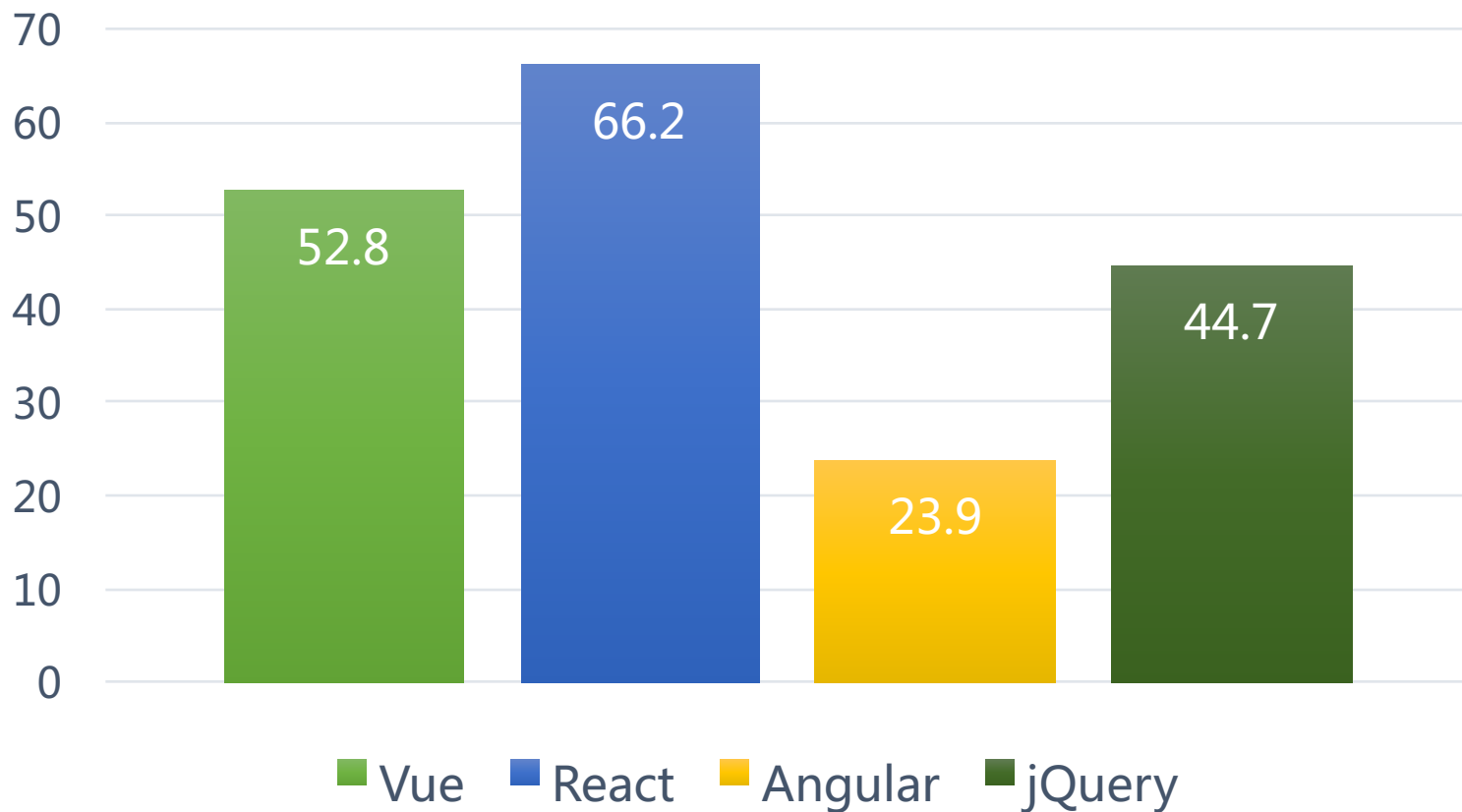
灵活

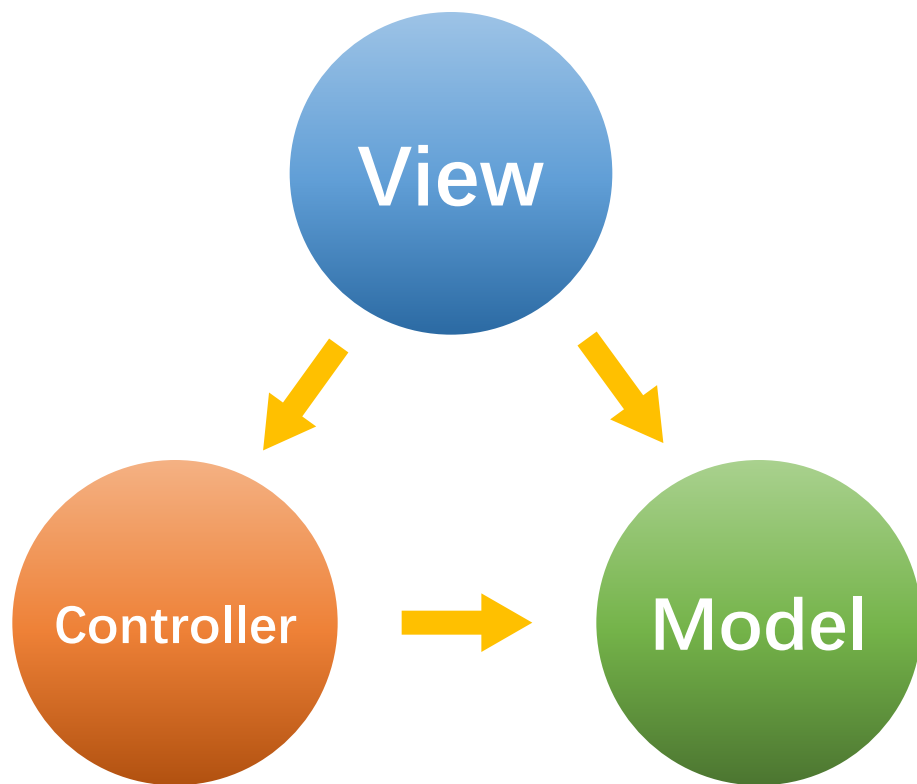
简单小巧的核心, 渐进式技术栈, 足以应付任何规模的应用。

性能

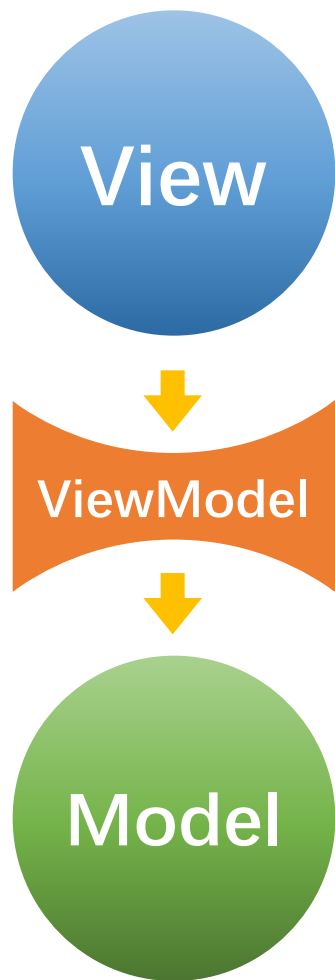
17kb min+gzip 运行大小
超快虚拟 DOM
最省心的优化

GitHub 上的主流前端框架





MVC



MVVM

- 工作台
- 供应链管理
- 系统管理
- 商家后台
- 流程中心
- 财务
 - 财务配置管理
 - 资金管理
 - 财务基础数据
 - 银行账号主数据
 - 应收核算
 - 应付核算
- 交付管理

财务 > 财务基础数据 > 银行账号

银行账号

更多查询

查询

重置

查看

导出

<input type="checkbox"/>	序号	银行	开户支行	银行账号	账户名称	状态
<input type="checkbox"/>	1	建设银行	建设银行科苑支行	635285965478652	金证科技	已审核
<input type="checkbox"/>	2	工商银行	广州大学城支行	6222023602037747722	黄端端	已审核
<input type="checkbox"/>	3	工商银行	公司科苑支行	5689526385963256458	泰山科技	待审核
<input type="checkbox"/>	4	建设银行	建设银行科苑支行	352385695236158	中潜股份	待审核
<input type="checkbox"/>	5	工商银行	工商银行	6532123546666666	华鹏有限公司	待审核
<input type="checkbox"/>	6	建设银行	ass	1111111113333	tomytest3	暂存
<input type="checkbox"/>	7	建设银行	一天一天	56542	trtr	待审核
<input type="checkbox"/>	8	建设银行	dsfdsf	5555556112	tomy2000	待审核
<input type="checkbox"/>	9	中国银行	gfsdgdg	555555611	tebu	待审核
<input type="checkbox"/>	10	中国银行	gfsdgdg	5555556	kapa	暂存
<input type="checkbox"/>	11	中国银行	fdasf	555555	vivo	已作废

20

K <

1

/ 5

> X

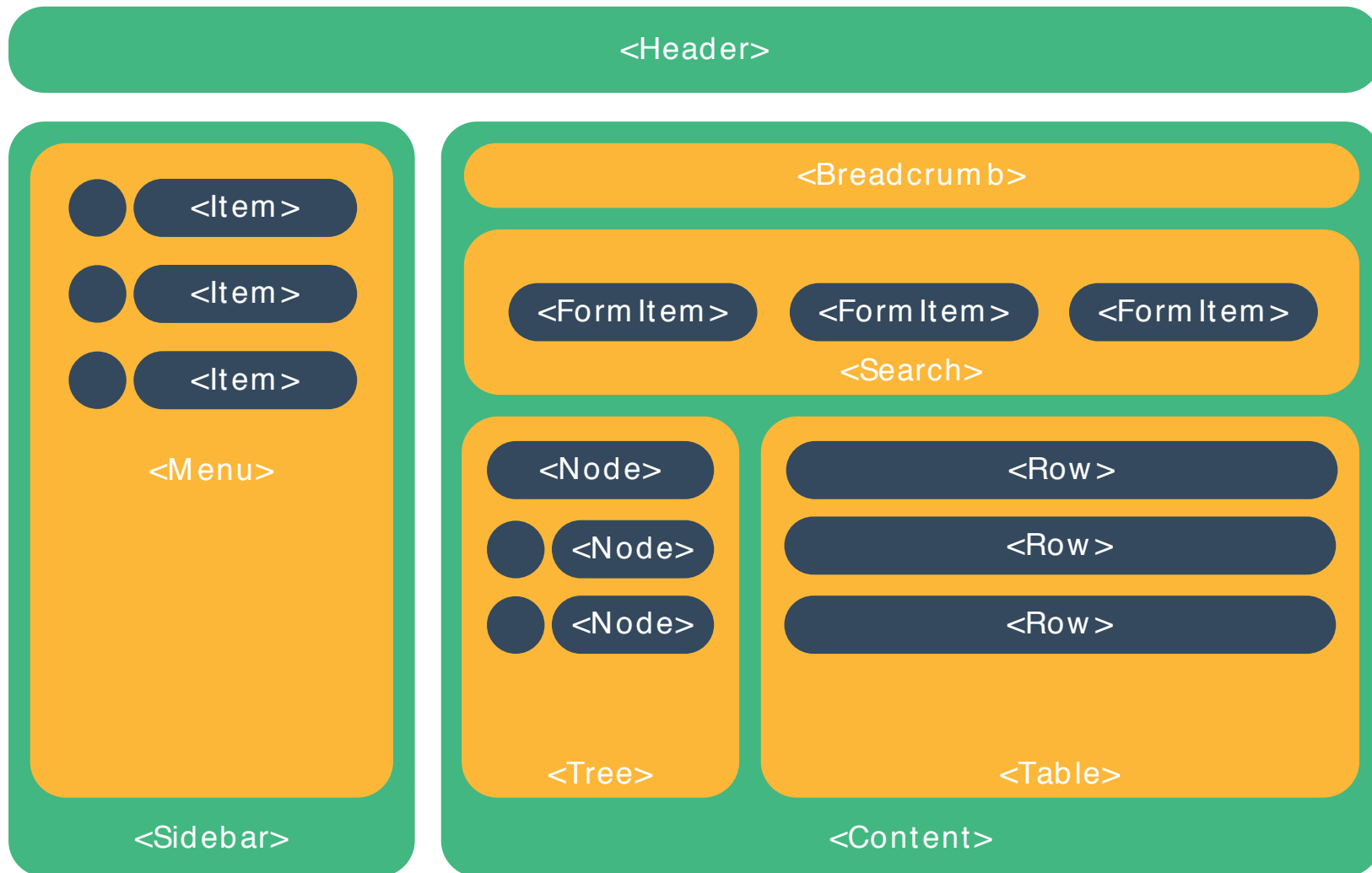
↺

显示从 1 到 20 条，总 89 条。每页显示 20 条。

组件的分类

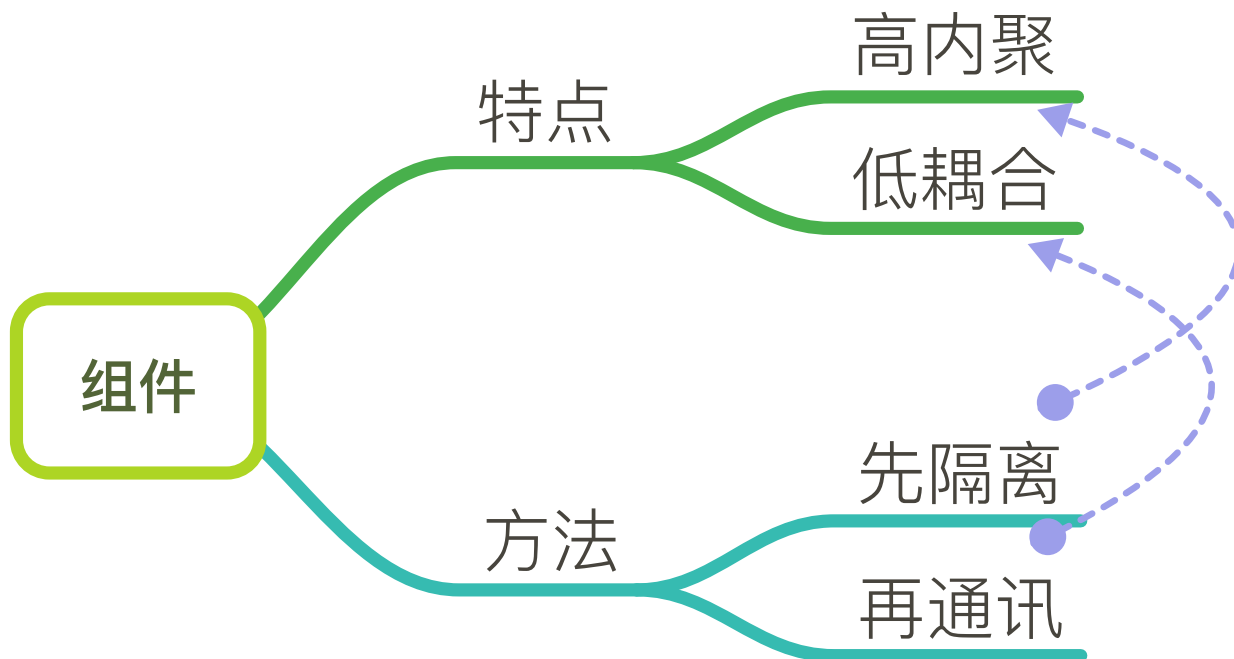


省钱 · 省心 · 省时间

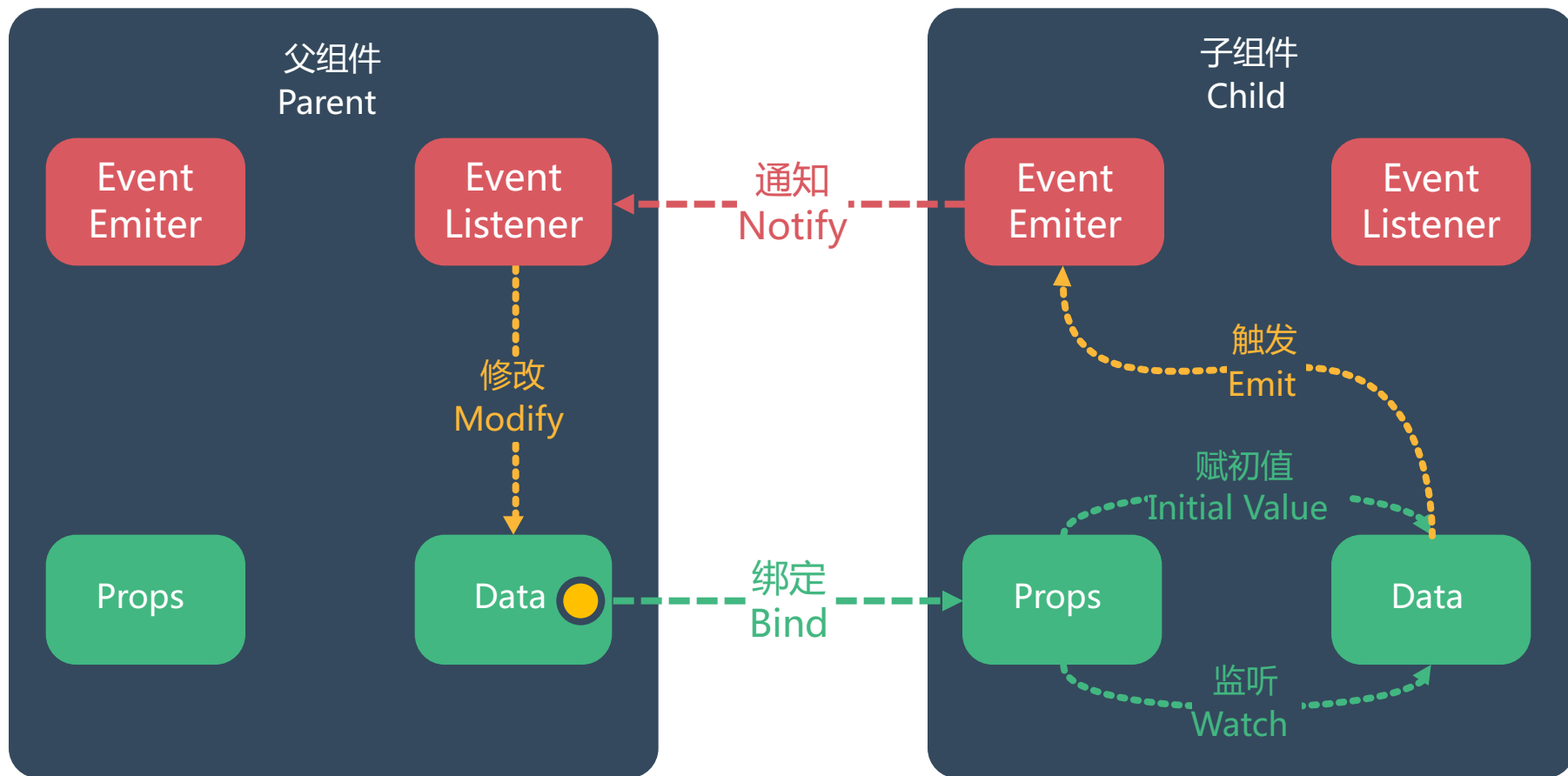


● 通用组件 ● 业务组件

怎样实现一个组件？



组件通讯：单向数据流

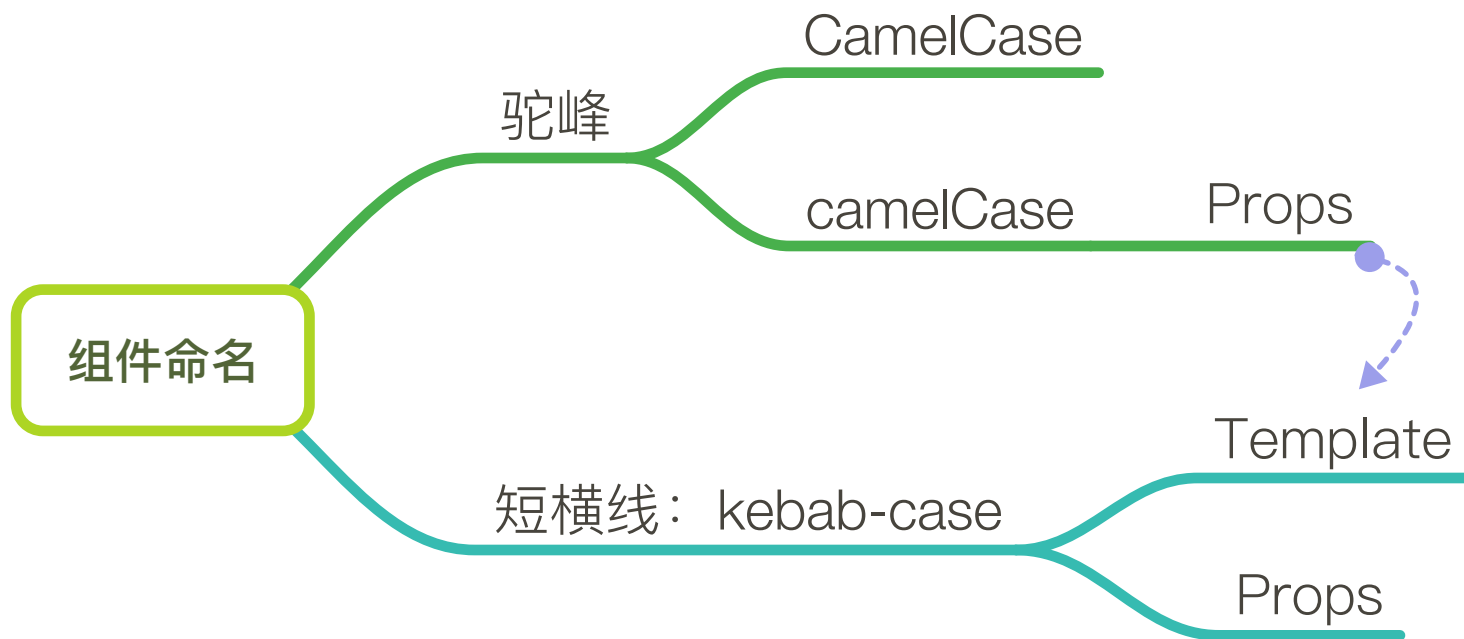


适当的嵌套

规范的命名

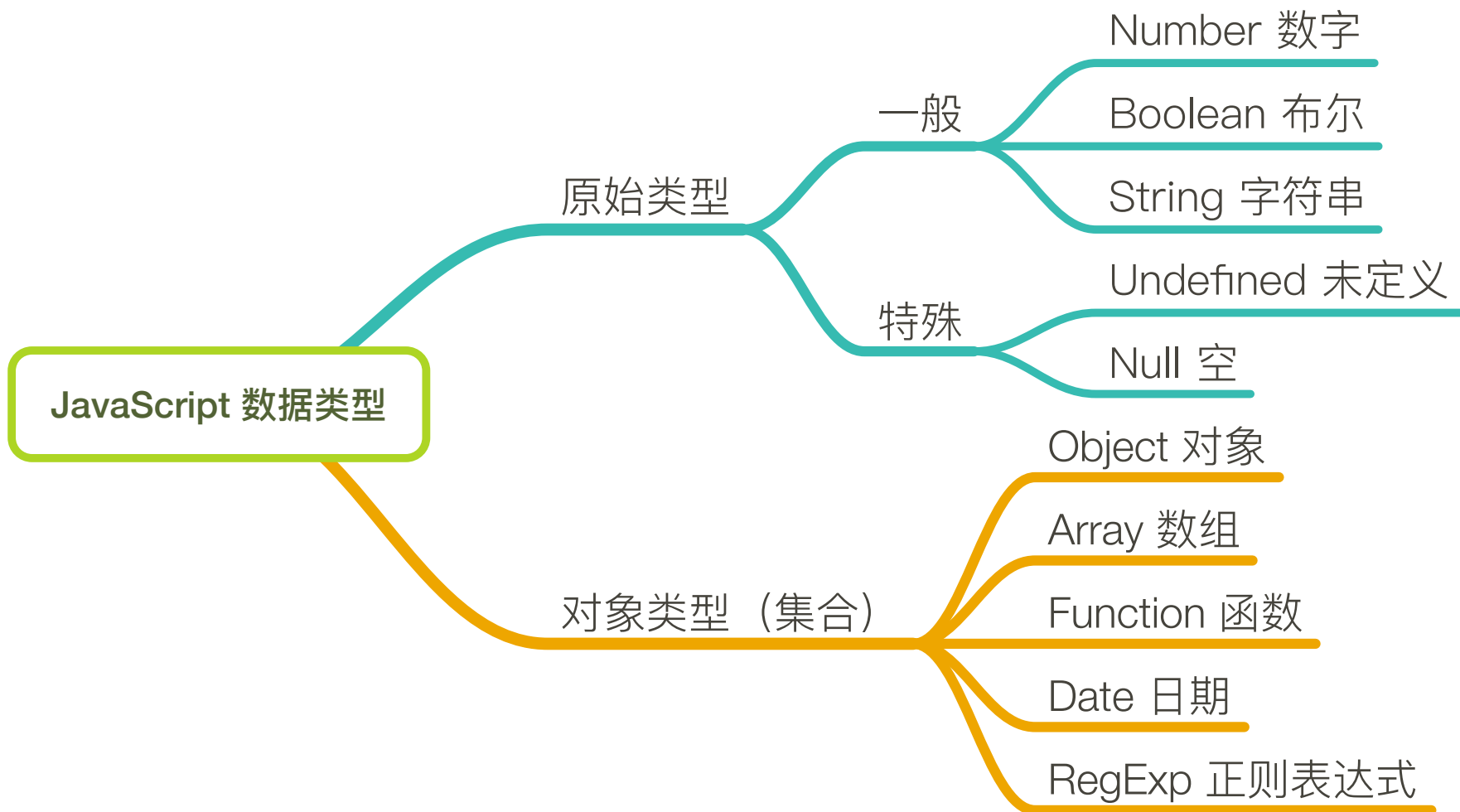
统一的接口

清晰的文件结构



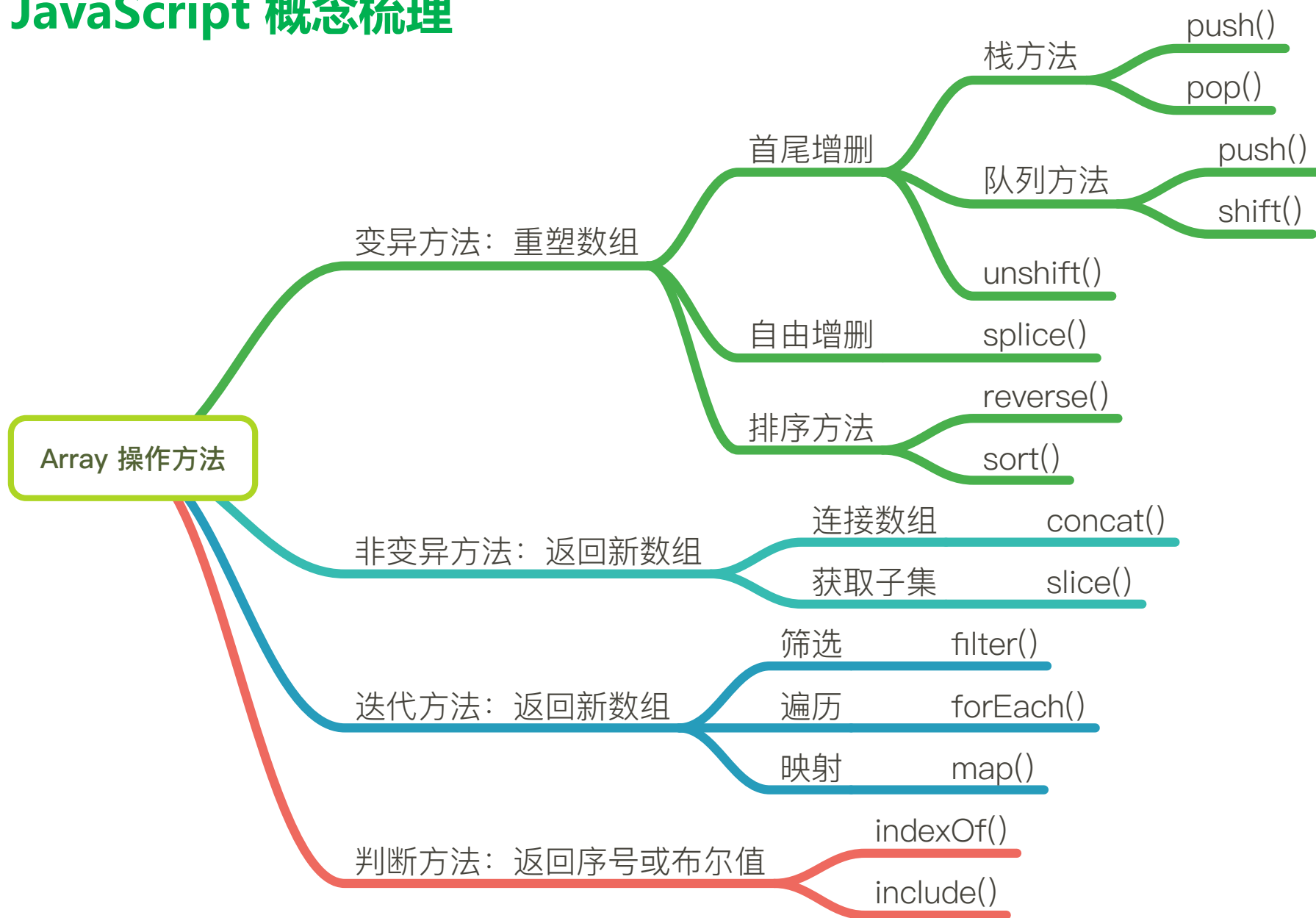
JavaScript 概念梳理







JavaScript 概念梳理



push

- 尾部增加

pop

- 尾部删除

unshift

- 头部增加

shift

- 头部删除

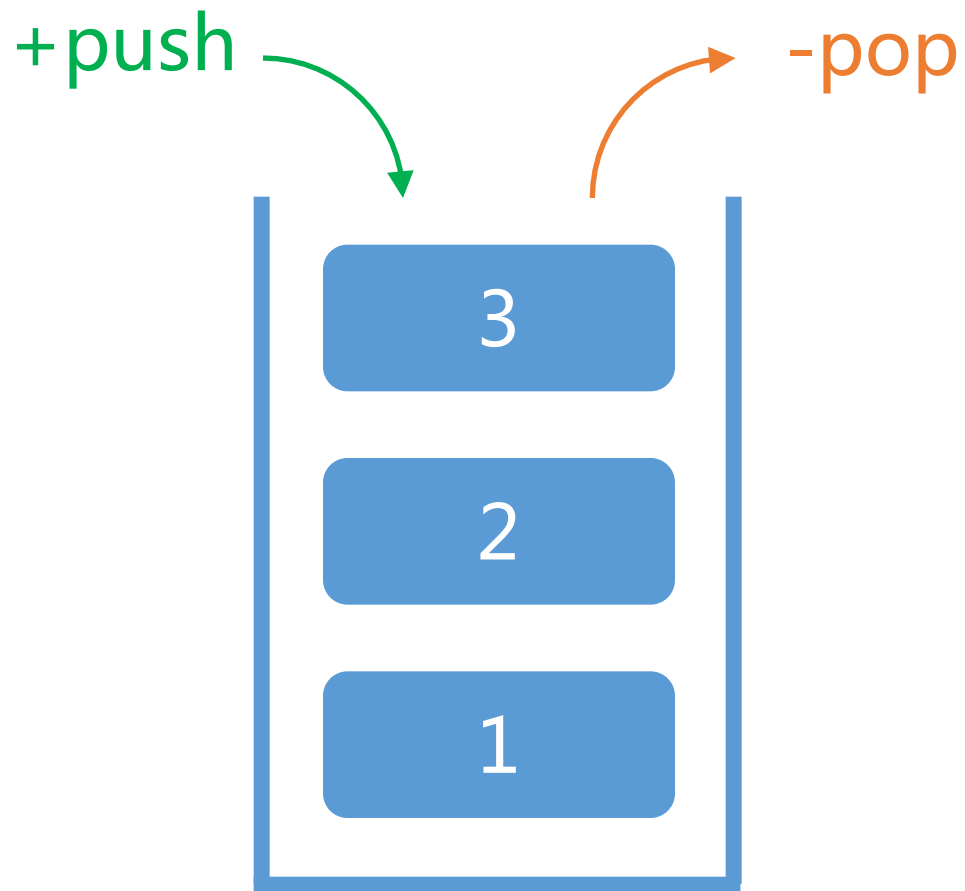
splice

- 自由增删

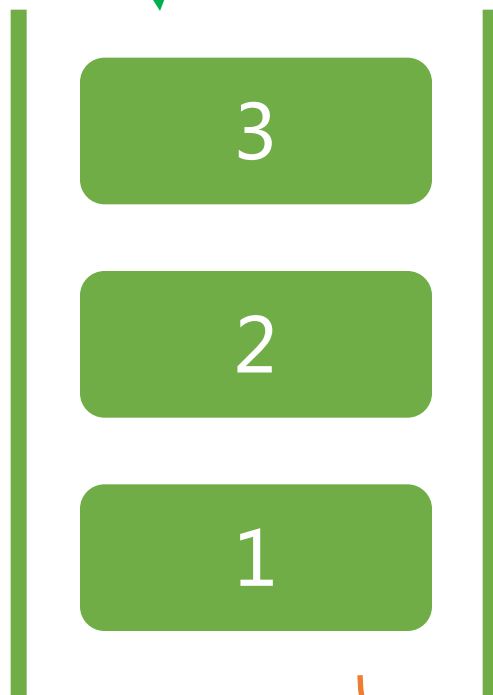
数组常用操作方法



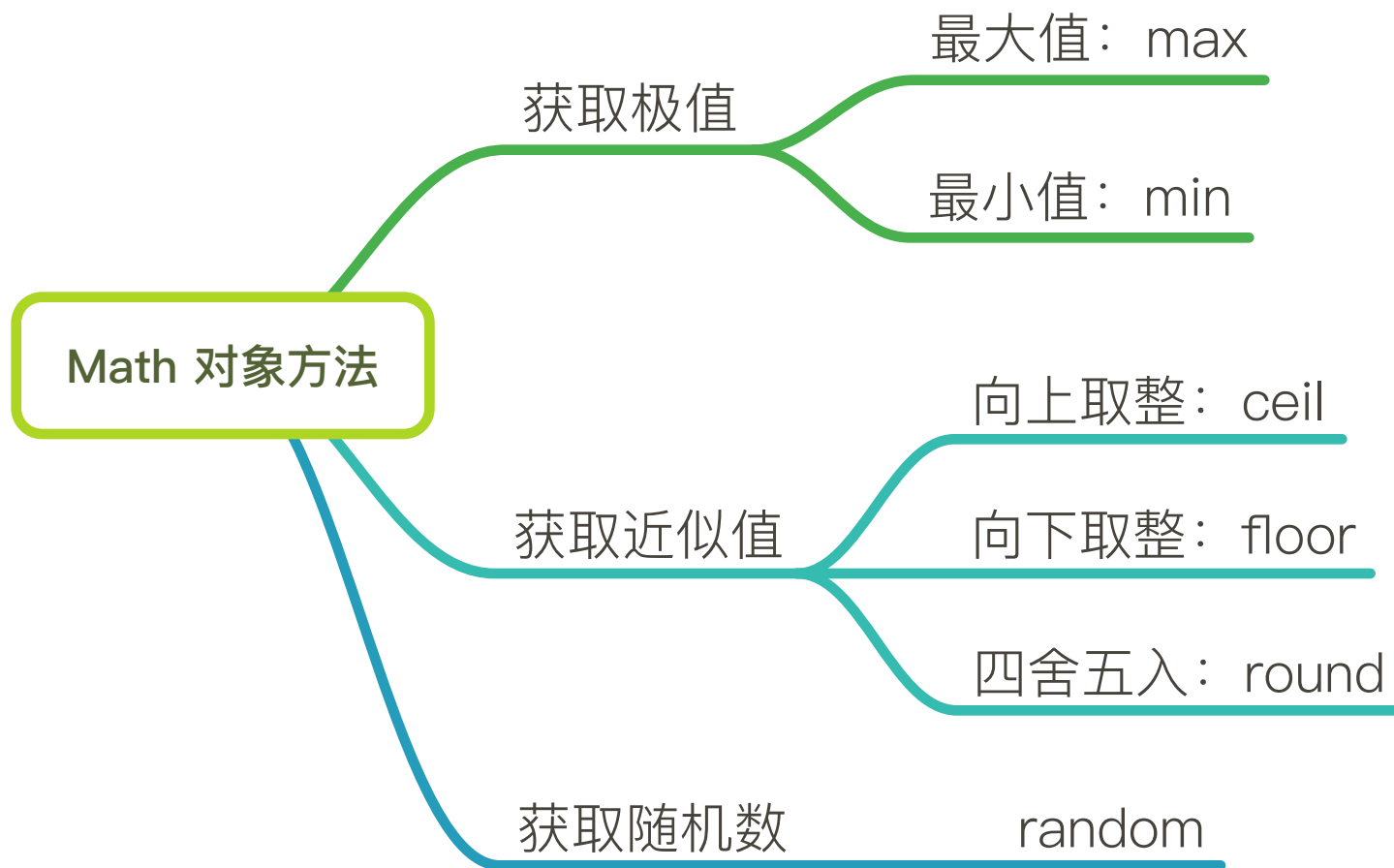
省钱 · 省心 · 省时间



+push



-shift





?

```
setTimeout(function() {  
    console.log("first")  
}, 0)  
console.log("second")
```



?

```
while(true){}  
setTimeout(function() {  
    console.log("first")  
}, 0)
```


Vue 组件开发常见问题



Template 只允许一个根元素

```
<template>
  <div class="m-finance">
    <t8t-search></t8t-search>
    <t8t-tree></t8t-tree>
    <t8t-table></t8t-table>
  </div>
</template>
```

Data 必须是函数

```
data() {  
  return {  
    currentValue: this.value  
  }  
},  
props: {  
  value: [Number, String]  
}
```

快速访问 Vue 实例

通过 Vue Devtools

```
$vm0
```

DOM 直接访问

```
$('el-menu').__vue__
```

使用局部样式

```
<style scoped>
  .m-finance .t8t-dialog {
    /* styles */
  }
</style>
```

```
<div data-v-hash>
  .t8t-dialog[data-v-hash]
```

如何判断一个对象？



省钱 · 省心 · 省时间

typeof

```
var isObject = function (obj) {  
    return obj !== null &&  
        typeof obj === 'object'  
}
```

如何判断一个数组？



省钱 · 省心 · 省时间

instanceof

```
var isArray = function (arr) {  
    return arr instanceof Array  
}
```

toLocaleString

```
computed: {  
  timeString: {  
    get () {  
      // Js默认以毫秒为单位，需要进行转换  
      let timeStamp = new Date(this.updateTime * 1000)  
      return timeStamp.toLocaleString()  
    },  
    set (val) {  
      let timeStamp = Math.round(Date.getTime() / 1000)  
      this.updateTime = val || timeStamp  
    }  
  }  
}
```


如何复制一个 JS 对象？



省钱 · 省心 · 省时间

```
var a = { prop: 'val' }  
var b = a  
a.prop = 'nothing'  
console.log(b.prop)
```

如何复制一个 JS 对象？



省钱 · 省心 · 省时间

浅拷贝

V.S.

深拷贝

// 浅拷贝

```
function lightCopy(obj) {  
    var copy = {}  
    for (var i in obj) {  
        copy[i] = obj[i]  
    }  
    return copy  
}
```

```
// 深拷贝
function deepCopy(obj) {
  let copy = {};
  Object.keys(obj).forEach(key => {
    if (obj[key] instanceof Array) {
      copy[key] = Array.from(obj[key]);
    }
    else if (typeof obj[key] === 'object') {
      copy[key] = deepCopy(obj[key]);
    }
    else {
      copy[key] = obj[key];
    }
  })
  return copy;
}
```

```
// utils.js  
export const lightCopy = function(obj) {...}  
export const deepCopy = function(obj) {...}
```

```
// page.js  
import { lightCopy, deepCopy } from utils.js  
  
var a = {...}  
var b = lightCopy(a)
```

如何复制一个 JS 对象？



省钱 · 省心 · 省时间

```
// 浅拷贝  
Var copy = Object.assign({}, obj)  
  
// 深拷贝  
Var copy = JSON.parse(JSON.stringify(obj))
```

如何判断项包含在数组中？



省钱 · 省心 · 省时间

indexOf

```
var isSelected = function (row) {  
    return (selection || []).indexOf(row) > -1  
}
```

// 通过或运算赋初始值

```
var text = this.value || ''
```

// 三元操作符

```
<span> {{ status ? '有效' : '无效' }} </span>
```

// 数字类型转换

```
var numString = '123'
```

```
var num = +numString
```


如何筛选一个数组？



省钱 · 省心 · 省时间

filter

```
var users = allUsers.filter(  
  user => user.logged === true)
```

Object.keys

```
if (Object.keys(search).length !== 0) {  
    args = Object.assign({}, args, search)  
}
```

如何循环删除数组的项？



省钱 · 省心 · 省时间

for + splice

```
for (let i = 0,; i < delIds.length - 1; i++) {  
    allItems.splice(delIds[i], 1)  
}
```

```
for (let i = delIds.length - 1; i > 0; i--)  
{  
    allItems.splice(delIds[i], 1)  
}
```

如何循环删除数组的项？



省钱 · 省心 · 省时间

forEach + indexOf + splice

```
delItems.forEach(item => {  
    allItems.splice(allItems.indexOf(item))  
})
```

THE END 😊

组件开发就用 **Vue**

