

Vue基础

key, 计算属性, 过滤器, 侦听器



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ Vue基础_更新监测, key作用
- ◆ Vue基础_过滤器
- ◆ Vue基础_计算属性
- ◆ Vue基础_侦听器

学习目标

Learning Objectives

1. 能够了解更新监测, key作用, 虚拟DOM, diff算法
2. 能够掌握设置动态样式
3. 能够掌握过滤器, 计算属性, 侦听器
4. 能够完成品牌管理案例



目录

Contents

- ◆ Vue基础_更新监测, key作用
- ◆ Vue基础_过滤器
- ◆ Vue基础_计算属性
- ◆ Vue基础_侦听器



Vue基础_更新监测和key

A decorative graphic featuring a central red-outlined hexagon with the Chinese characters '思考' (Thinking) inside. Surrounding this central hexagon are several other hexagons: a light gray one at the top left, a dark red one at the top right, a dark gray one at the bottom left, and a red-outlined one at the bottom right. A dashed line connects the central hexagon to the dark gray one at the bottom left.

思考

问题1: v-for数组变化, 会更新页面吗?

问题2: 哪些数组方法会造成v-for更新, 哪些不能?

1.0_v-for更新监测

目标：目标结构变化, 触发v-for的更新

- 情况1: 数组翻转
- 情况2: 数组截取
- 情况3: 更新值
- 口诀:
 - 数组变更方法, 就会导致v-for更新, 页面更新
 - 数组非变更方法, 返回新数组, 就不会导致v-for更新, 可采用覆盖数组或this.\$set()

push()
pop()
shift()
unshift()
splice()
sort()
reverse()

```
<template>
  <div>
    <ul>
      <li v-for="(item, index) in arr">{{ item }}</li>
    </ul>
    <button @click="reBtn">翻转</button>
    <button @click="jBtn">截取前3个</button>
    <button @click="upBtn">点击改掉第一个元素的值</button>
  </div>
</template>
```

filter() 、 concat() 和 slice()



小结

1. 哪些数组方法会导致v-for更新页面?

可以改变原数组的方法

2. 有的数组方法不导致v-for更新页面, 如何处理?

拿返回的新数组, 直接替换旧数组

this.\$set()方法更新某个值



思考

问题1: JS原生数组变化, 如何更新页面?

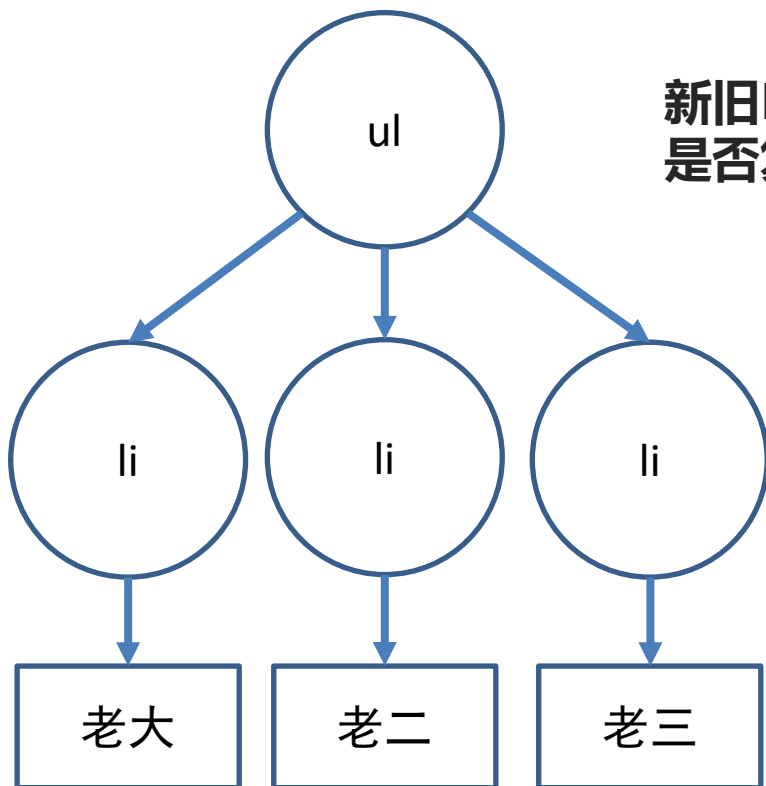
问题2: 重新渲染, 会重绘和重排性能不高吧?

问题3: v-for是如何更新DOM呢?

1.1_v-for就地更新

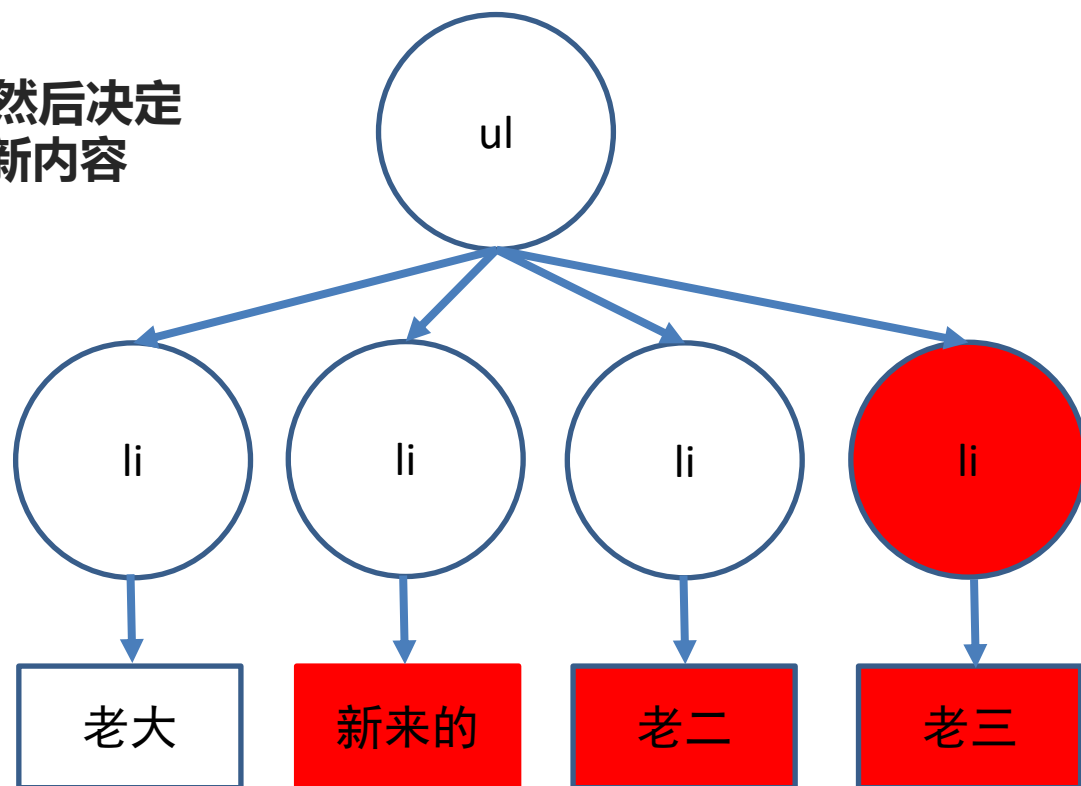
目标：当数组改变后是如何更新的

旧-虚拟DOM结构



新旧DOM产生后对比, 然后决定是否复用真实DOM/更新内容

新-虚拟DOM结构



```
arr: ["老大", "老二", "老三"]
```

```
arr: ["老大", "新来的", "老二", "老三"]
```



v-for更新时, 是如何操作DOM的?

循环出新的虚拟DOM结构, 和旧的虚拟DOM
结构对比, 尝试复用标签就地更新内容



思考

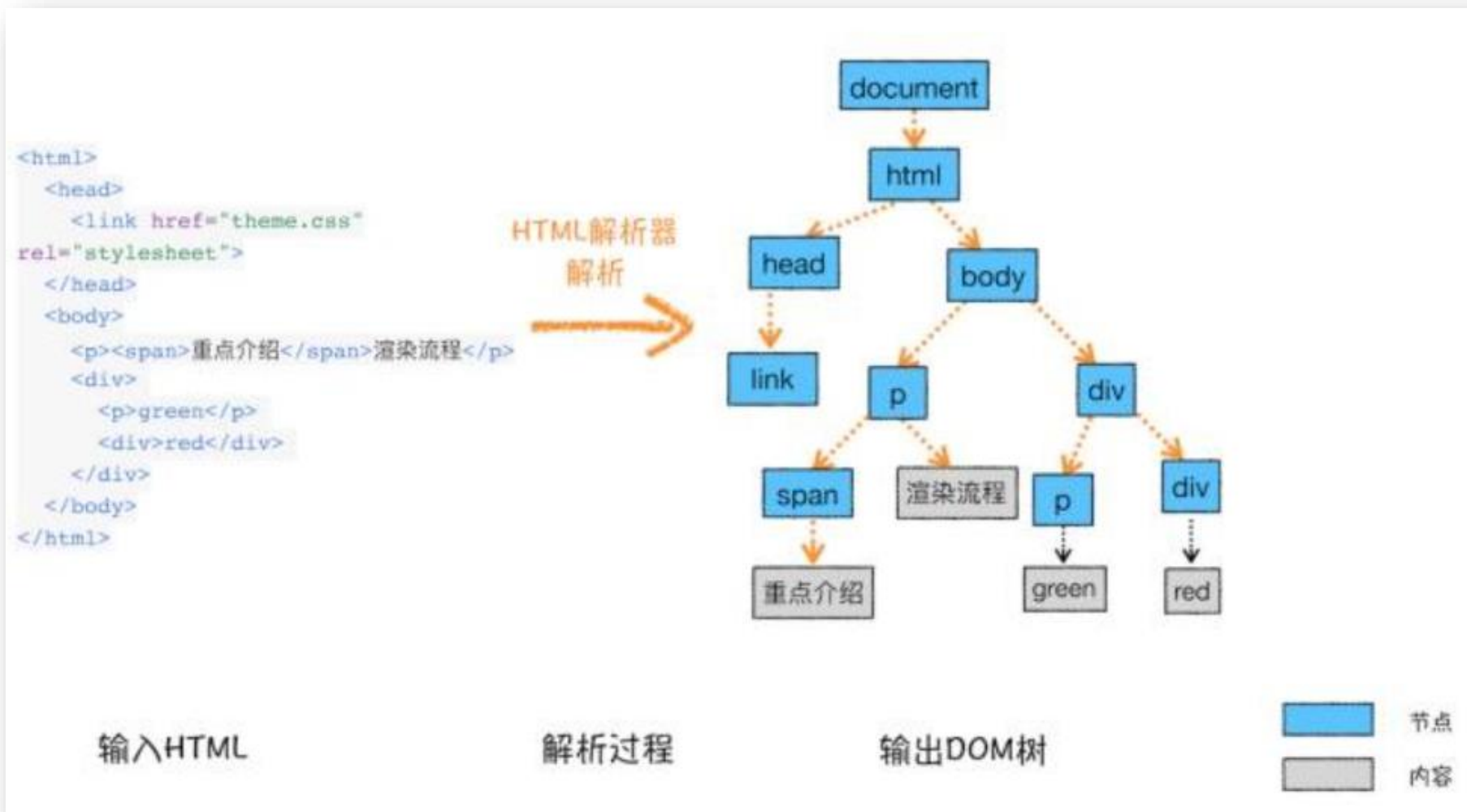
问题1: DOM树是什么? 它的结构什么样?

问题2: 频繁操作DOM会引发什么呢?

问题3: 如何提高性能呢?

1.2_真实DOM

目标：在document对象上, 渲染到浏览器上显示的标签

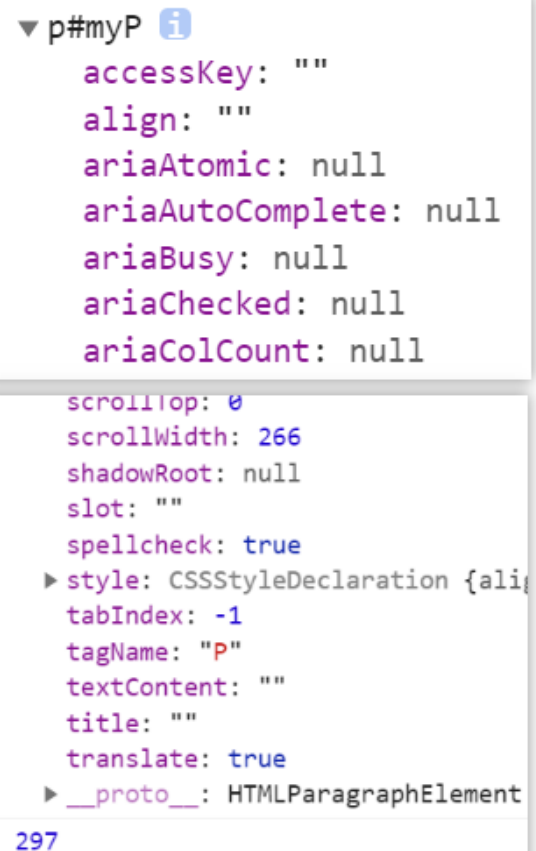


1.2_虚拟DOM

目标：本质是保存节点信息, 属性和内容的一个JS对象

```
<template>
  <div id="box">
    <p class="my_p">123</p>
  </div>
</template>
```

```
const dom = {
  type: 'div',
  attributes: [{id: 'box'}],
  children: {
    type: 'p',
    attributes: [{class: 'my_p'}],
    text: '123'
  }
}
```



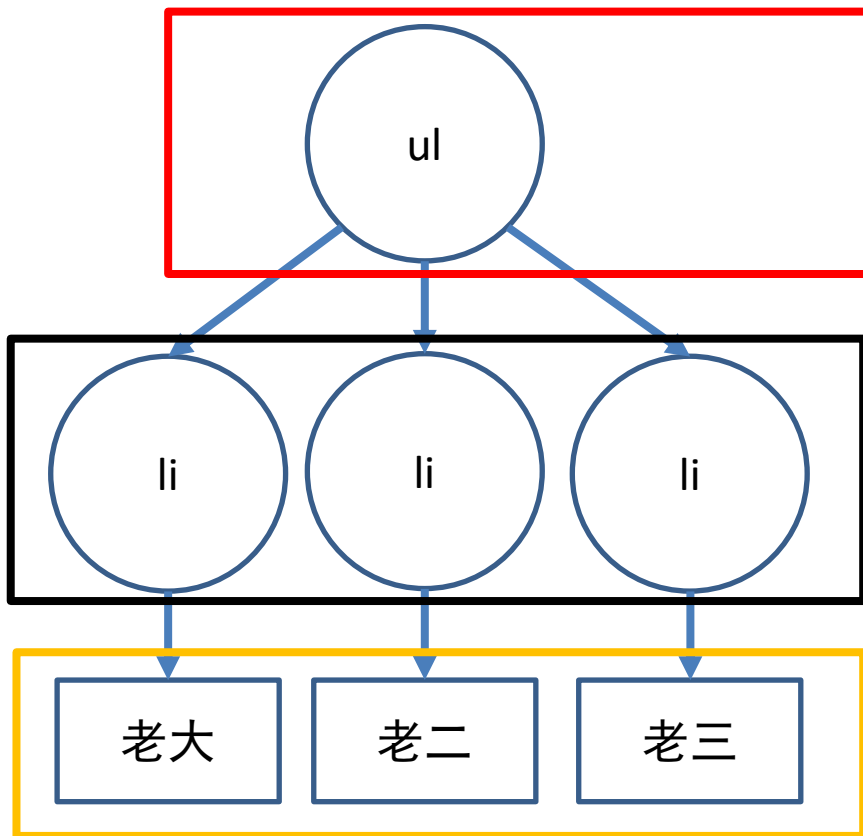
```
▼ p#myP ⓘ
  accessKey: ""
  align: ""
  ariaAtomic: null
  ariaAutoComplete: null
  ariaBusy: null
  ariaChecked: null
  ariaColCount: null
  scrollTop: 0
  scrollWidth: 266
  shadowRoot: null
  slot: ""
  spellcheck: true
  ▶ style: CSSStyleDeclaration {ali
  tabIndex: -1
  tagName: "P"
  textContent: ""
  title: ""
  translate: true
  ▶ __proto__: HTMLParagraphElement
297
```

真实DOM属性过多, 遍历耗时

1.2_虚拟DOM

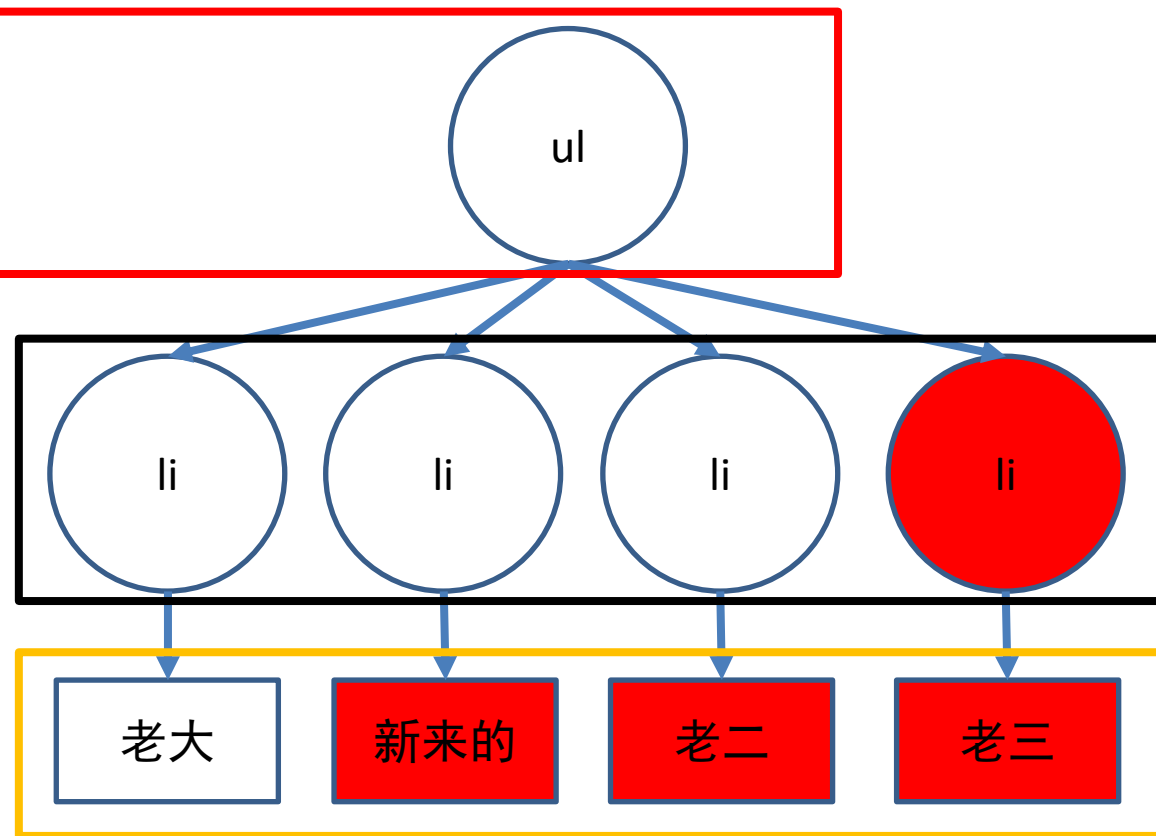
目标：在内存中比较变化部分，然后给真实DOM打补丁(更新)

旧-虚拟DOM结构

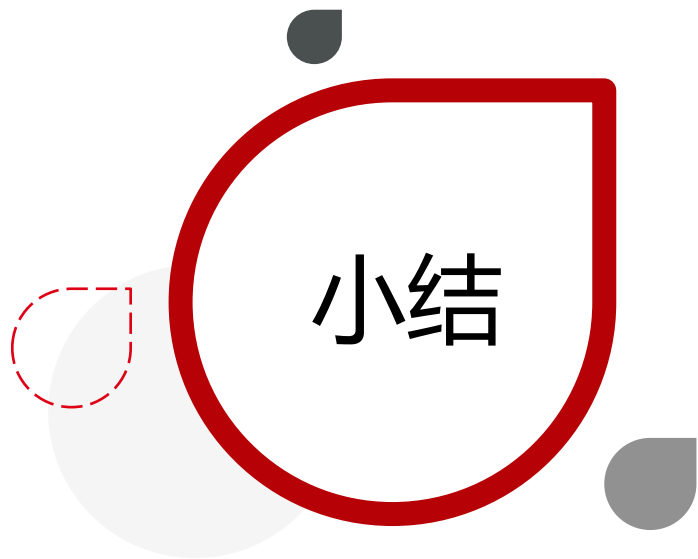


```
arr: ["老大", "老二", "老三"]
```

新-虚拟DOM结构



```
arr: ["老大", "新来的", "老二", "老三"]
```



1. 虚拟DOM是什么?

本质就是一个JS对象, 保存DOM关键信息

2. 虚拟DOM好处?

提高DOM更新的性能, 不频繁操作真实DOM,
在内存中找到变化部分, 再更新真实DOM(打补丁)



思考

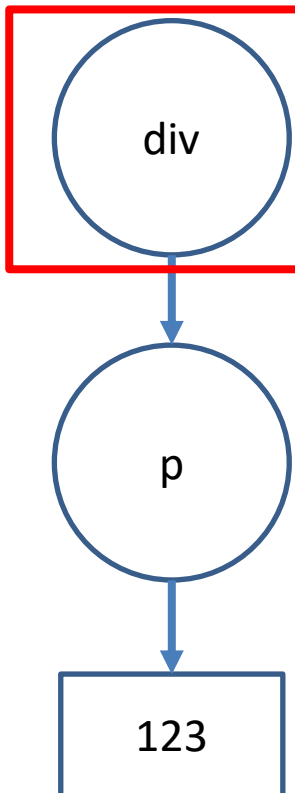
问题1: 新的虚拟DOM根元素, 或者属性变了呢? 如何更新?

问题2: 具体如何比较新旧虚拟DOM呢?

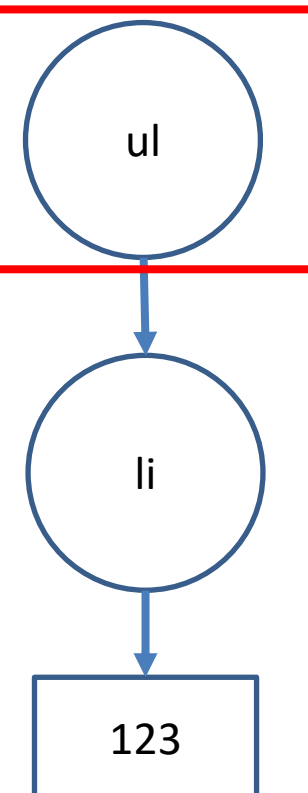
1.3_diff算法

目标：同级比较-根元素变化-整个dom树删除重建

旧-虚拟DOM结构



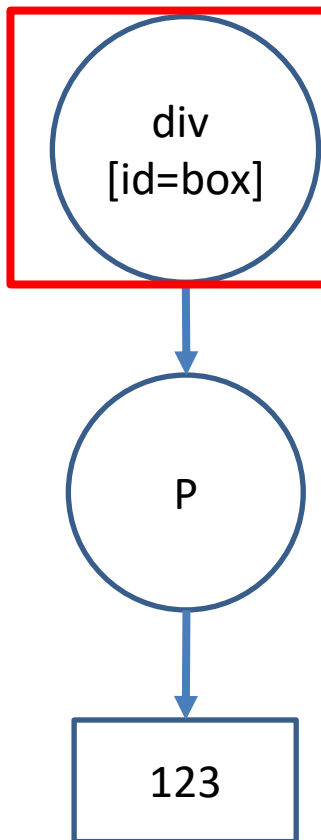
新-虚拟DOM结构



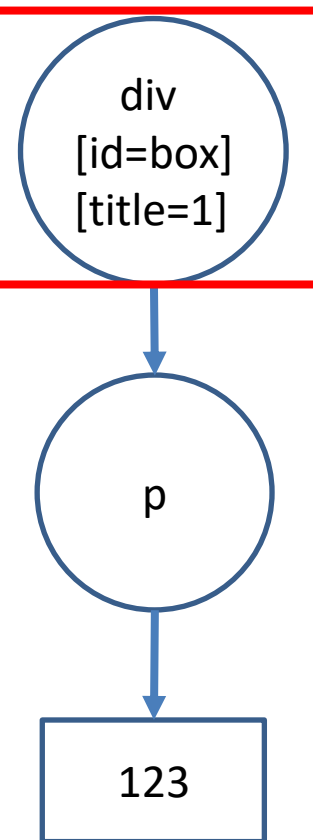
1.3_diff算法

目标：同级比较-根元素不变-属性改变更新属性

旧-虚拟DOM结构



新-虚拟DOM结构





1. diff算法如何比较新旧虚拟DOM?

同级比较

2. 根元素变化?

删除重新建立整个DOM树

3. 根元素未变, 属性改变?

DOM复用, 只更新属性



问题: 如果标签内子标签/内容改变, diff算法是如何对比的?

1.4_无key

- 从第二个往后更新内容 – 性能不高

```
<ul id="myUL">
  <li v-for="str in arr">
    {{ str }}
    <input type="text">
  </li>
</ul>
<button @click="addFn">下标为1的位置新增一个</button>
```

```
<button @click="splice" >上标为1的位置新增一个</button>
```

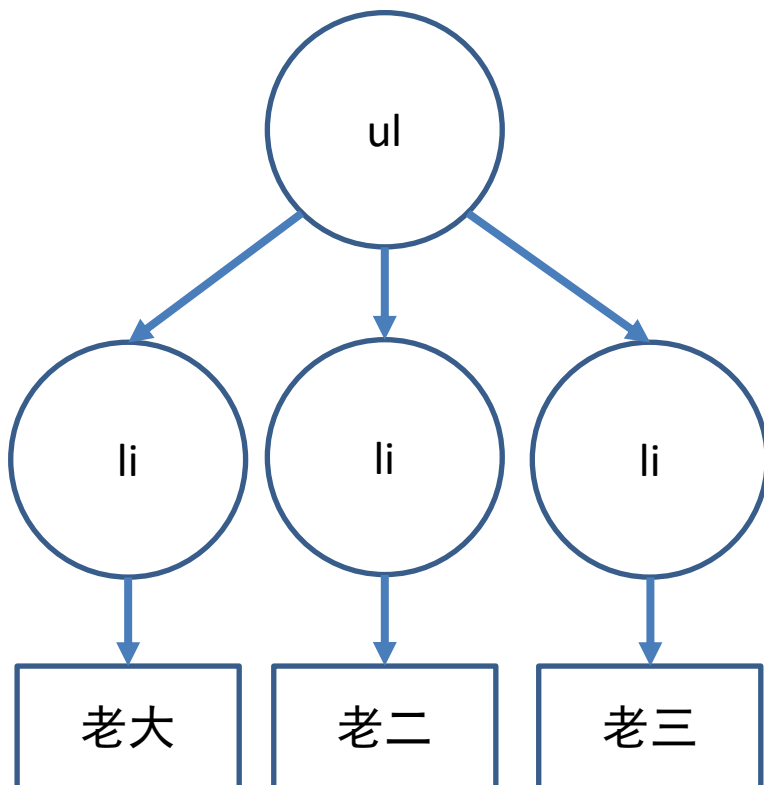
- 老大
- 老二
- 老三

下标为1的位置新增一个

1.4_无key

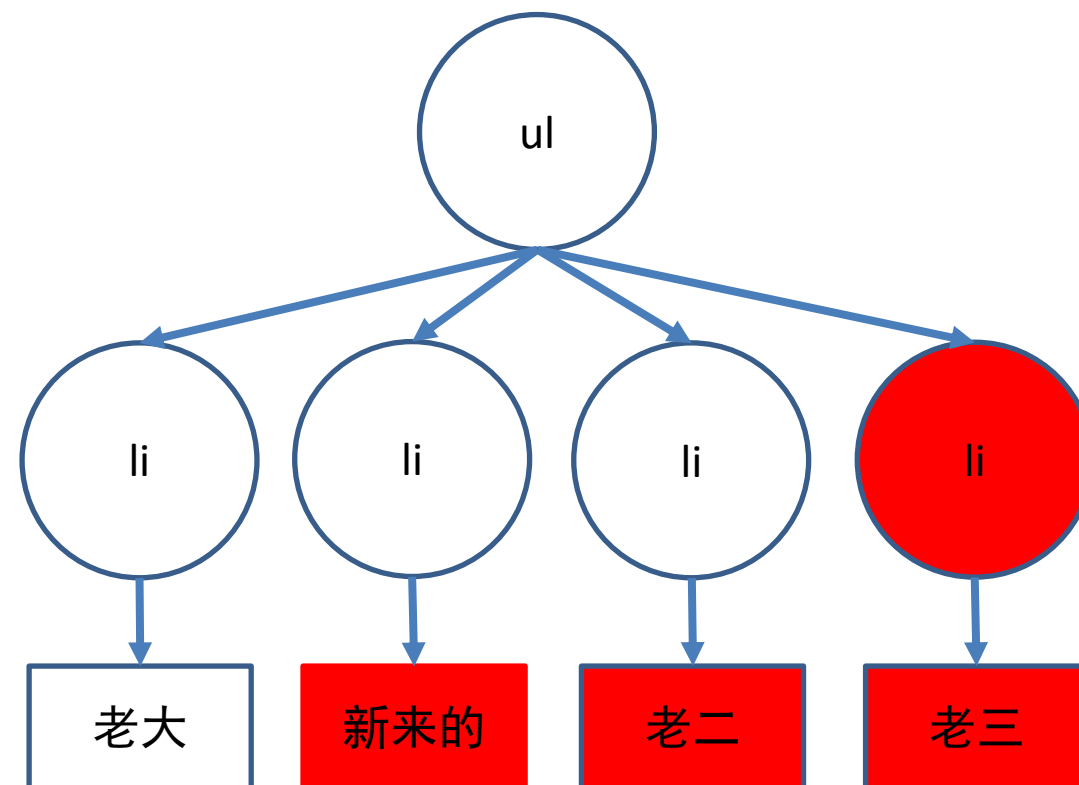
目标：最大限度尝试就地修改/复用相同类型元素

旧-虚拟DOM结构



```
arr: ["老大", "老二", "老三"]
```

新-虚拟DOM结构



```
arr: ["老大", "新来的", "老二", "老三"]
```

1.4_有key, 值为索引

目标: 有key属性, 基于key的来比较新旧虚拟DOM, 移除key不存在元素

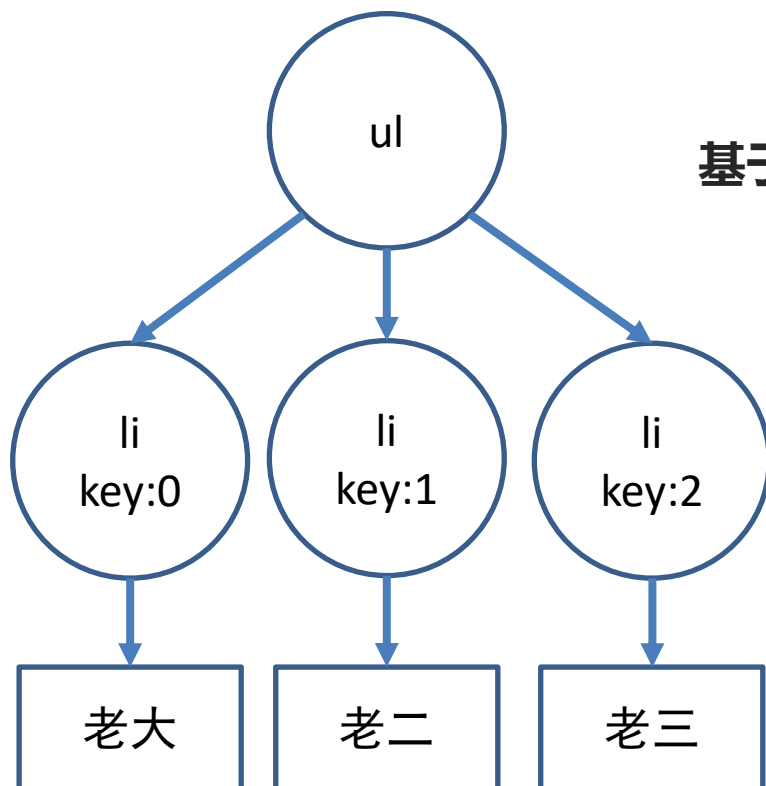
```
<ul id="myUL">
  <li v-for="(str, index) in arr" :key="index">
    {{ str }}
    <input type="text">
  </li>
</ul>
<button @click="addFn">下标为1的位置新增一个</button>
```

```
<button @click="addFn">上标为1的位置删除一个</button>
</ul>
```


1.4_有key, 值为索引

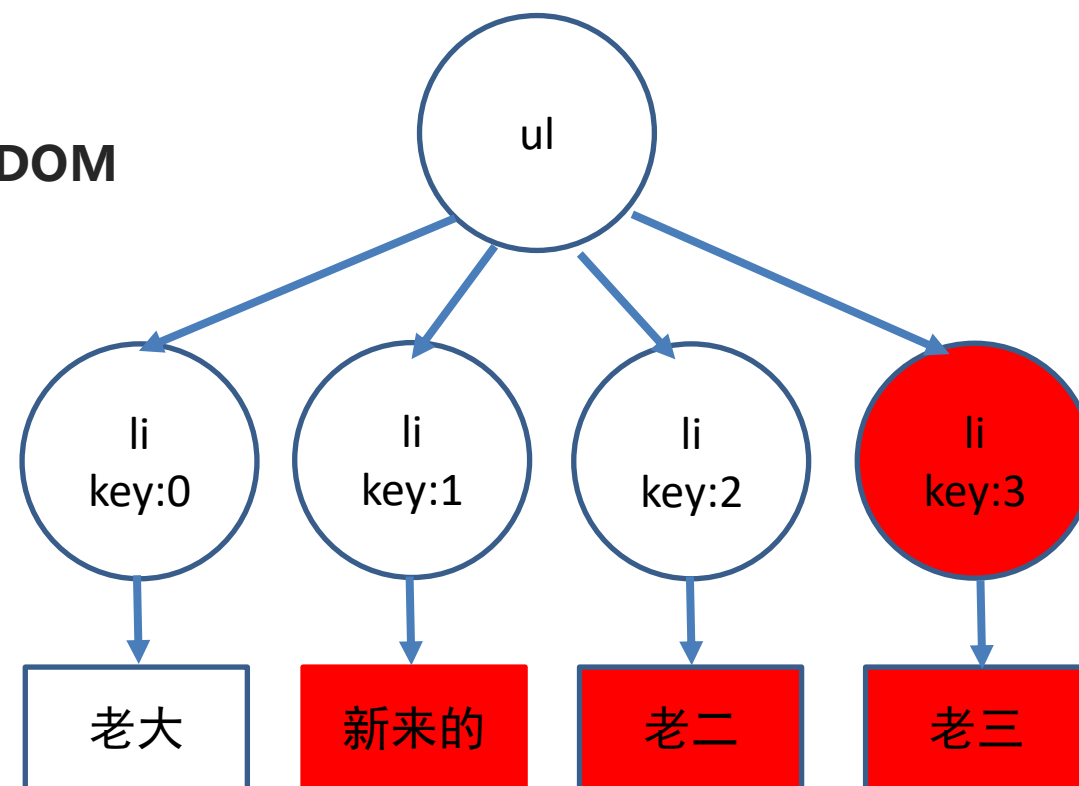
目标：先产生新旧虚拟DOM, 根据key比较, 还是就地更新

旧-虚拟DOM结构



基于key来比较新旧DOM

新-虚拟DOM结构



```
arr: ["老大", "老二", "老三"]
```

```
arr: ["老大", "新来的", "老二", "老三"]
```

1.4_有key, 值唯一不重复的字符串或数字

目标: 有key属性, 基于key的来比较新旧虚拟DOM, 移除key不存在元素

- 给每个数据换成对象, 准备id, 把id的值作为key

```
<ul id="myUL">
  <li v-for="obj in brr" :key="obj.id">
    {{ obj.name }}
    <input type="text">
  </li>
</ul>
```

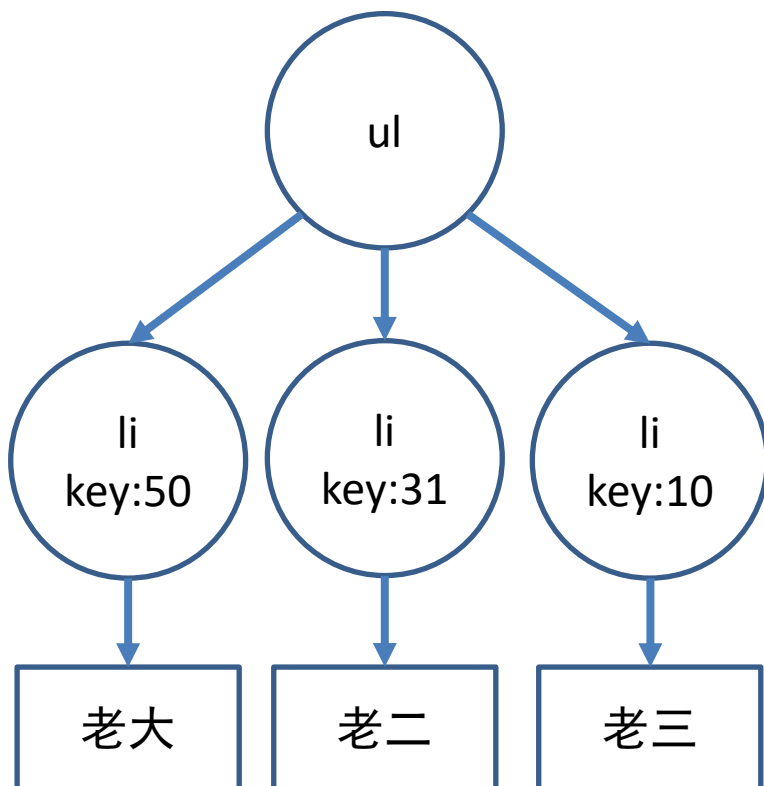
```
<\n>
```

```
<\n>
```

1.4_有key, 值为id

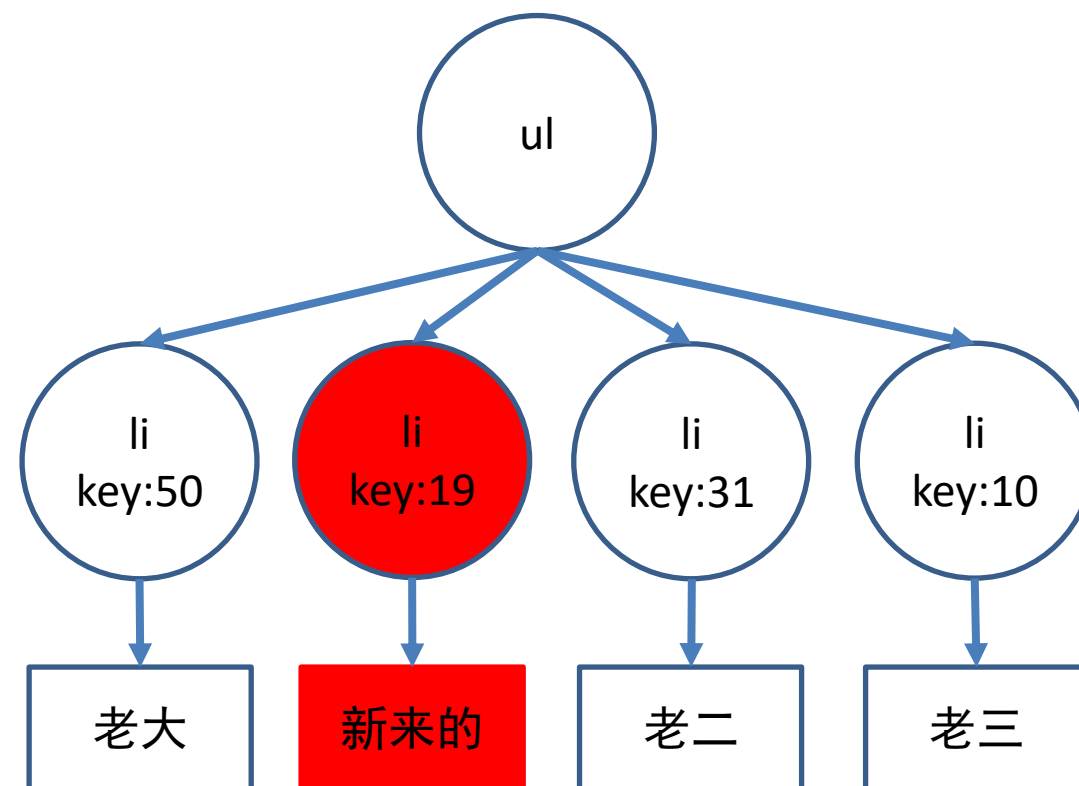
目标：先产生新旧虚拟DOM, 根据key比较

旧-虚拟DOM结构



arr: ["老大", "老二", "老三"]

新-虚拟DOM结构



arr: ["老大", "新来的", "老二", "老三"]



1. 子元素或者内容改变会分diff哪2种情况比较?

无key, 就地更新

有key, 按照key比较

2. key值要求是?

唯一不重复的字符串或者数值

3. key应该怎么用?

有id用id, 无id用索引

4. key的好处?

可以配合虚拟DOM提高更新的性能



总结

v-for什么时候会更新页面呢?

数组采用更新方法, 才导致v-for更新页面

vue是如何提高更新性能的?

采用虚拟DOM+diff算法提高更新性能

虚拟DOM是什么?

本质是保存dom关键信息的JS对象

diff算法如何比较新旧虚拟DOM?

根元素改变 – 删除当前DOM树重新建

根元素未变, 属性改变 – 更新属性

根元素未变, 子元素/内容改变

无key – 就地更新 / 有key – 按key比较



问题1: JS判断标签是否应该使用某个类名?

问题2: vue能不能直接在标签里判断?

1.6_动态class

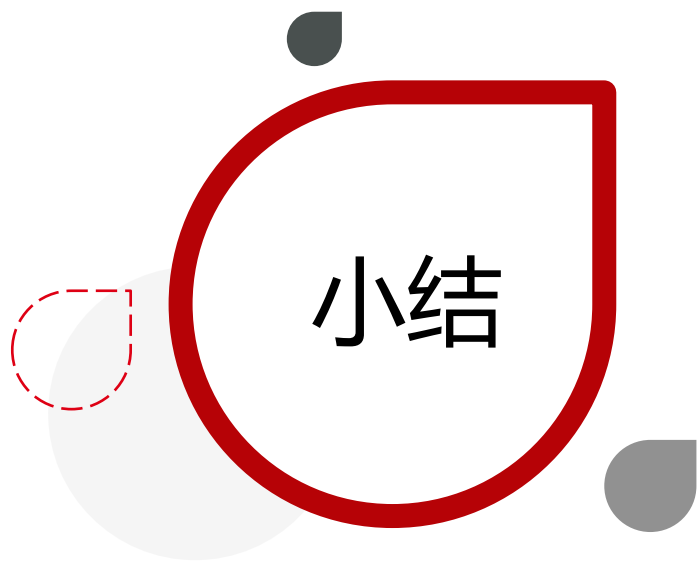
目标: 用v-bind给标签class设置动态的值

- 语法 :class="{类名: 布尔值}"

```
<div>
  <p :class="{redStr: bool}">值为true, key作为类名生效</p>
</div>
```

```
<style scoped>
.redStr{
  color: red;
}
</style>
```

```
data() {
  return {
    bool: true
  }
}
```



如何给标签class属性动态赋值?

:class= "{类名: 布尔值}" , true使用, false不用



问题: 能否给style属性赋vue变量呢?

1.7_动态style

目标: 给标签动态设置style的值

- 语法 :style="{css属性名: 值}"

```
<div>
  <p :style="{color: colorStr}">value变量的值, 将被赋予给css属性的key生效</p>
</div>
```

```
</div>
data() {
  return {
    colorStr: 'red'
  }
}
```

```
}
```



给style赋值和class区别是？

:class="{类名: 布尔值}" , true使用, false不用

:style="{css属性名: 值}"

1.8_案例-品牌管理(铺)



案例

品牌数据铺设

需求1: 把默认数据显示到表格上

需求2: 注意资产超过100的, 都用红色字体标记出来

效果如下

localhost:3000/#/品牌管理

编号	资产名称	价格	创建时间	操作
100	外套	199	Thu Aug 12 2010 08:00:00 GMT+0800 (中国标准时间)	删除
101	裤子	34	Sun Sep 01 2013 08:00:00 GMT+0800 (中国标准时间)	删除
102	鞋	25.4	Thu Nov 22 2018 08:00:00 GMT+0800 (中国标准时间)	删除
103	头发	19900	Sat Dec 12 2020 08:00:00 GMT+0800 (中国标准时间)	删除

资产名称

0

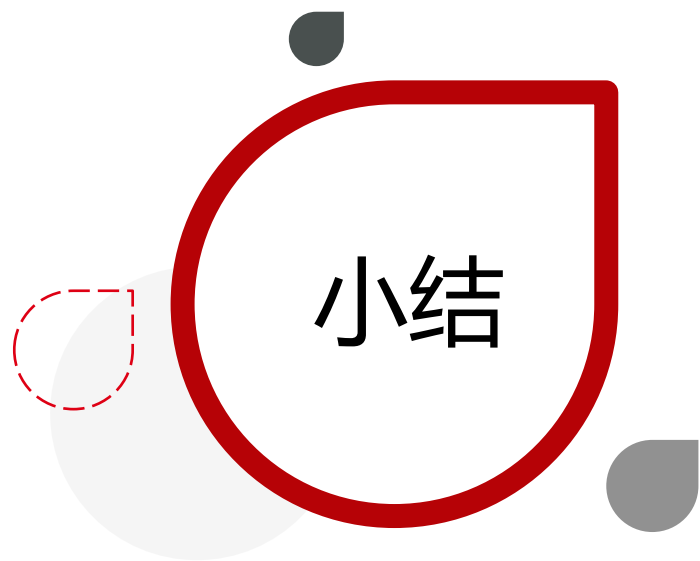
添加资产



品牌数据铺设

分析

- ① 先铺设静态页面 --- 去.md文档里, 复制数据和标签模板
- ② 此案例使用bootstrap, 需要下载, 并导入到工程main.js中
- ③ 用v-for配合默认数据, 把数据默认铺设到表格上显示
- ④ 直接在标签上, 大于100价格, 动态设置red类名



用到了哪些技术点？

布局用bootstrap, yarn下载引入使用

v-for把数据渲染到表格里

:class动态判断价格给红色类名

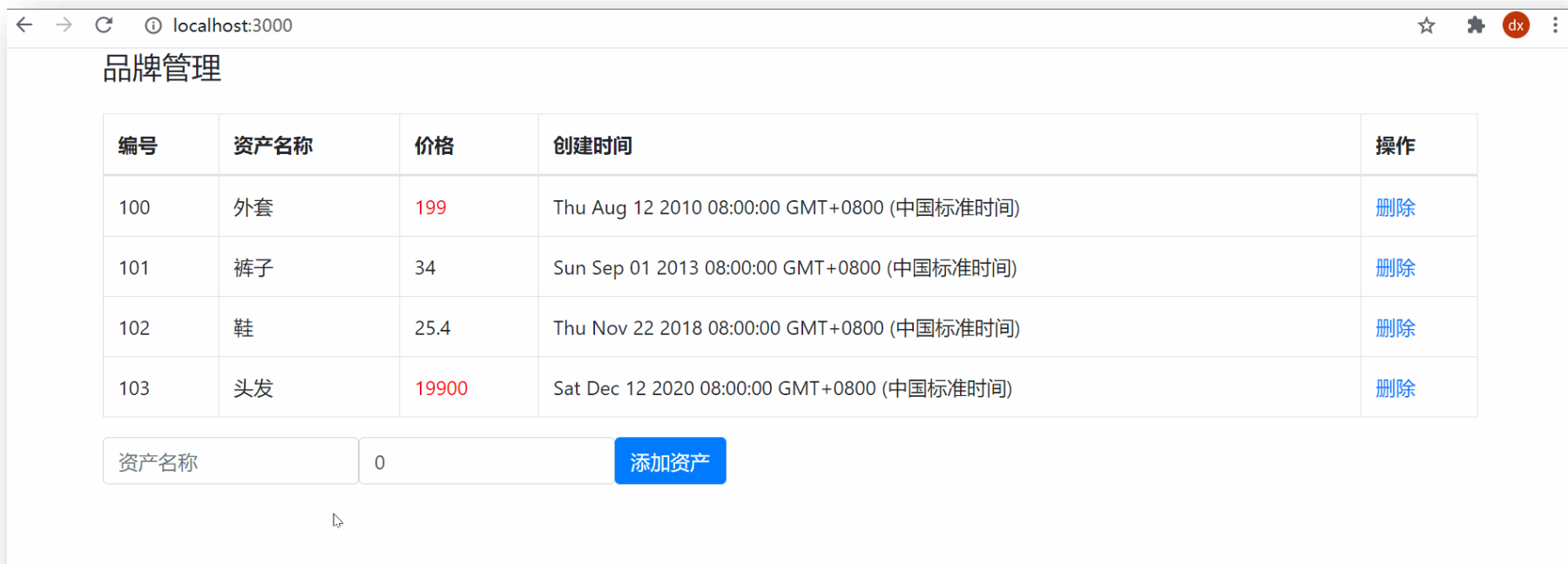


品牌数据增加, 功能实现

需求1: 实现表单数据新增进表格功能

需求2: 判断用户输入是否为空给提示

效果如下



编号	资产名称	价格	创建时间	操作
100	外套	199	Thu Aug 12 2010 08:00:00 GMT+0800 (中国标准时间)	删除
101	裤子	34	Sun Sep 01 2013 08:00:00 GMT+0800 (中国标准时间)	删除
102	鞋	25.4	Thu Nov 22 2018 08:00:00 GMT+0800 (中国标准时间)	删除
103	头发	19900	Sat Dec 12 2020 08:00:00 GMT+0800 (中国标准时间)	删除

资产名称 [添加资产](#)



品牌数据增加, 功能实现

分析

- ① 添加资产按钮 – 绑定点击事件
- ② 给表单v-model绑定vue变量收集用户输入内容
- ③ 添加数组到数组中
- ④ 判断用户内容是否符合规定



涉及到了哪些技术点?

@绑定事件

v-model收集表单数据

.prevent阻止按钮提交表单刷新页面

v-model修饰符

数组取最后一个元素

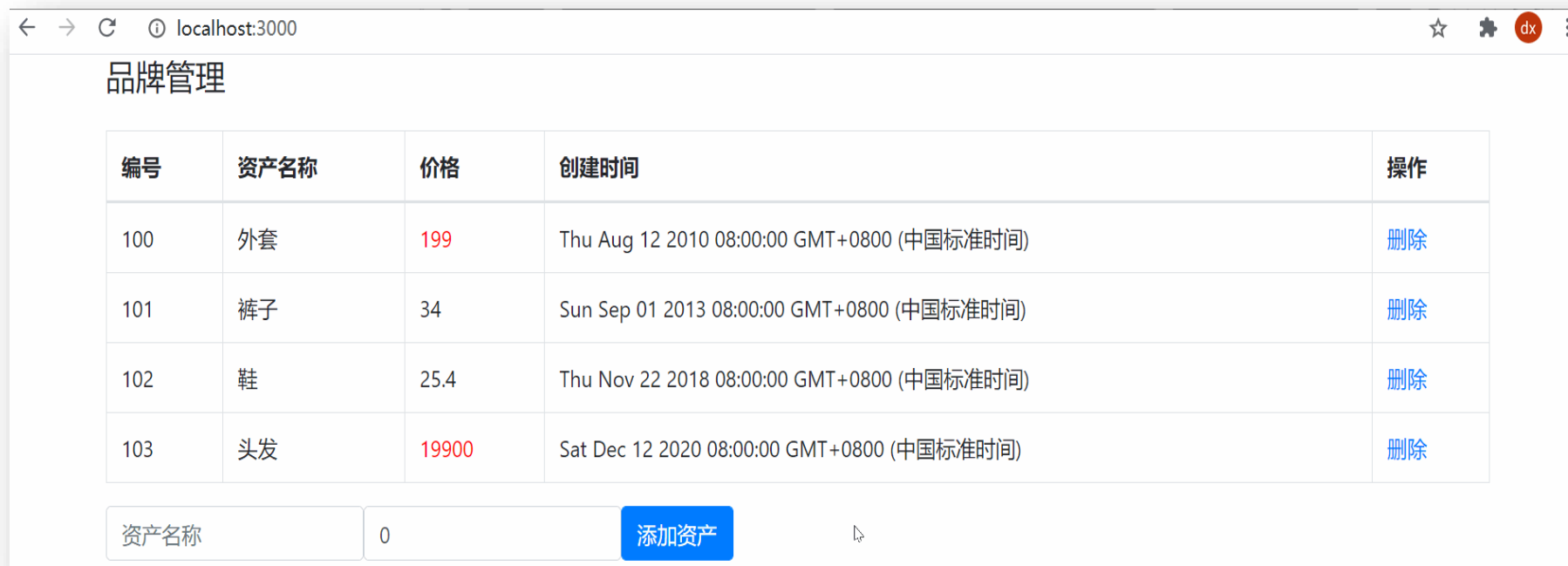
案例

品牌数据删除, 功能实现

需求1: 点击删除的a标签, 删除数据

需求2: 删除没数据了要提示暂无数据的tfoot

效果如下



编号	资产名称	价格	创建时间	操作
100	外套	199	Thu Aug 12 2010 08:00:00 GMT+0800 (中国标准时间)	删除
101	裤子	34	Sun Sep 01 2013 08:00:00 GMT+0800 (中国标准时间)	删除
102	鞋	25.4	Thu Nov 22 2018 08:00:00 GMT+0800 (中国标准时间)	删除
103	头发	19900	Sat Dec 12 2020 08:00:00 GMT+0800 (中国标准时间)	删除

添加资产



品牌数据删除, 功能实现

分析

- ① a标签绑定点击事件
- ② 给事件方法传id
- ③ 通过id, 找到对应数据删除
- ④ 删除光了要让tfoot显示
- ⑤ 删除光了再新增, 有bug(id值问题)需要修复



涉及到了哪些技术点?

事件传id

删除数组元素方法

id值判断问题



目录

Contents

- ◆ Vue基础_更新监测, key作用
- ◆ Vue基础_过滤器
- ◆ Vue基础_计算属性
- ◆ Vue基础_侦听器



Vue基础_过滤器



思考

问题1: JS原生如何封装一个字母翻转的函数?

问题2: 生活当中的过滤器是什么? Vue中如何实现过滤器?

2.0_过滤器_定义使用

目标: 转换格式, 过滤器就是一个函数, 传入值返回处理后的值

- 过滤器只能用在, 插值表达式和v-bind动态属性里
- Vue中的过滤器场景
 - 字符串翻转, "输入hello, world", 输出"dlrow ,olleh"
 - 字母转大写, 输入"hello", 输出"HELLO"
- 语法:
 - Vue.filter("过滤器名", (值) => {return "返回处理后的值"})
 - filters: {过滤器名字: (值) => {return "返回处理后的值"}}

```
<template>
  <div>
    {{ msg }}
    <!-- 1. 过滤器格式: vue变量 | 过滤器名 -->
    <!-- 翻转过滤器reverse -->
    <p>{{ msg | reverse }}</p>
    <!-- 首字母过滤器first -->
    <p :title="msg | toUp">鼠标长时间停留</p>
  </div>
</template>
```

```
<\reverse>
<\first>
<b :title="msg | toUp">鼠标长时间停留</b>
```




小结

1. Vue中过滤器作用?

传入一个值, 处理后返回另外一个值

2. 如何定义过滤器?

main.js – Vue.filter('过滤器名字', 函数体)

某.vue文件 – filters: {'过滤器名字': 函数体}

3. 如何使用过滤器?

Vue变量 | 过滤器名字

4. 过滤器注意事项?

插值表达式 / 动态属性



问题1: 如何给过滤器传参?

问题2: 如何使用多个过滤器?

2.1_过滤器_传参和多过滤器

目标: 可同时使用多个过滤器, 或者给过滤器传参

- 语法:
 - 过滤器传参: vue变量 | 过滤器(实参)
 - 多个过滤器: vue变量 | 过滤器1 | 过滤器2

```
<template>
  <div>
    {{ msg }}
    <!-- 可以给过滤器传参 -->
    <p>{{ msg | reverse("|") }}</p>
    <!-- 可以多个过滤器 -->
    <p :title="msg | toUp | reverse('|')">鼠标长时间停留</p>
  </div>
</template>
```

```
<\f6wbJ9f6>
```

```
<\q!^>
```

```
<:title=msg | toUp | reverse('|')>鼠标长时间停留</p>
```



1. 多个过滤器使用?

vue变量 | 过滤器1 | 过滤器2

2. 如何给过滤器传额外值

vue变量 | 过滤器(值)

2.2_案例-品牌管理(时间格式化)



案例

创建时间以YYYY-MM-DD格式显示

需求: 表格里时间用过滤器+moment模块, 格式化成YYYY-MM-DD 格式

效果如下:

localhost:3000/?#/

品牌管理

编号	资产名称	价格	创建时间	操作
100	外套	199	2010-08-12	删除
101	裤子	34	2013-09-01	删除
102	鞋	25.4	2018-11-22	删除
103	头发	19900	2020-12-12	删除

资产名称

0

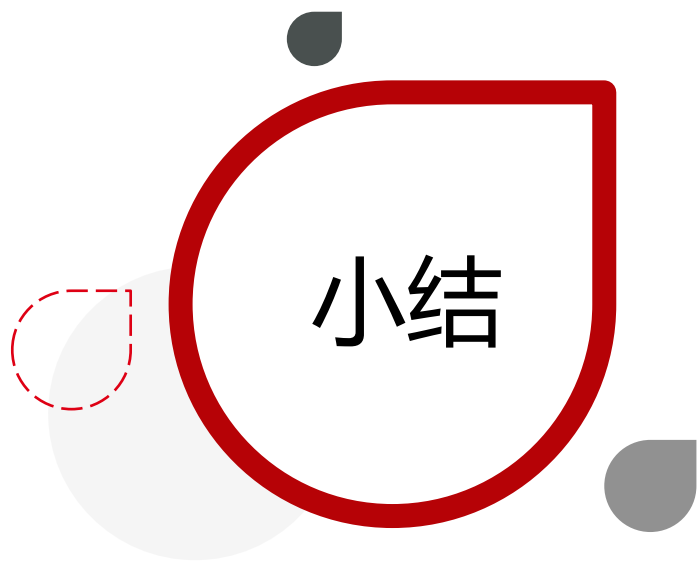
添加资产



创建时间以YYYY-MM-DD格式显示

分析

- ① 下载moment模块到当前工程
- ② 定义过滤器, 传入时间, 用moment模块格式化, 返回我们想要的格式



涉及到了哪些技术点?

yarn下包

过滤器定义和使用

moment复习



目录

Contents

- ◆ Vue基础_更新监测, key作用
- ◆ Vue基础_过滤器
- ◆ Vue基础_计算属性
- ◆ Vue基础_侦听器



vue基础_计算属性



思考

问题1: js原生变量num值, 来自a与b变量的和如何写?

问题2: 当a和b变量改变同步新的和给num变量, 还会吗?

3.0_计算属性_computed

目标: 一个变量的值, 依赖另外一些数据计算而来的结果

- 语法:

```
computed: {  
  "计算属性名" () {  
    return "值"  
  }  
}
```

```
<template>  
  <div>  
    <p>和为: {{ num }}</p>  
  </div>  
</template>
```

注意: 计算属性也是vue数据变量, 所以不要和data里重名, 用法和data相同



1. 计算属性使用场景?

当变量的值, 需要通过别人计算而得来

2. 计算属性特点?

函数内使用的变量改变, 重新计算结果返回

3. 计算属性注意事项?

计算属性名和data里名字不能重复



思考

问题1: 不用计算属性, 用函数调用能实现吗?

问题2: 计算属性优势在哪里?

3.1_计算属性_缓存

目标: 计算属性, 基于依赖项的值进行缓存, 依赖的变量不变, 都直接从缓存取结果

```
<template>
  <div>
    <p>{{ reverseMessage }}</p>
    <p>{{ reverseMessage }}</p>
    <p>{{ reverseMessage }}</p>
    <p>{{ getMessage() }}</p>
    <p>{{ getMessage() }}</p>
    <p>{{ getMessage() }}</p>
  </div>
</template>
```

```
<\template>
<\div>
  <b>{{ getMessage() }}<\b>
```

```
<body>
  <div id="app">
    <p>{{reverseMessage}}</p>
    <p>{{reverseMessage}}</p>
    <p>{{reverseMessage}}</p>
  </div>
</body>
<script>
  const vm = new Vue({
    el:'#app',
    data:{
      message:'hello vue'
    },
    computed:{
      reverseMessage(){
        return this.message
          .split('')
          .reverse()
          .join()
      }
    }
  })
</script>
```

渲染时第一次遇到

reverseMessage()

函数返回值存入缓存

再次遇到直接使用缓存值不再调用函数

缓存值



1. 计算属性好处是?

带缓存

依赖项不变, 直接从缓存取

依赖项改变, 函数自动执行并重新缓存

2. 计算属性使用场景?

当变量值, 依赖其他变量计算而得来才用

3.2_案例-品牌管理(总价和均价)



案例

完成总价和均价的显示结果

需求: 完成商品总价和均价的数据显示功能

效果如下:

localhost:3000/#/

品牌管理

编号	资产名称	价格	创建时间	操作
100	外套	199	2010-08-12	删除
101	裤子	34	2013-09-01	删除
102	鞋	25.4	2018-11-22	删除
103	头发	19900	2020-12-12	删除
统计:	总价钱为: 20158		平均价: 5039	

资产名称

0

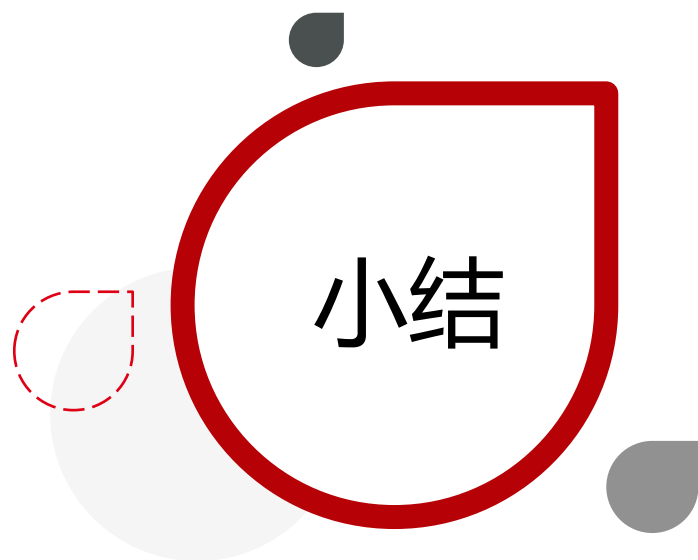
添加资产



完成总价和均价的显示结果

分析:

- ① tbody最后新增一行tr
- ② 定义总价和均价2个计算属性, 在函数里编写计算过程返回结果
- ③ 把计算属性用到相应标签处



涉及到的知识点?

computed定义和使用计算属性

数组reduce方法累积运算



思考

问题1: 计算属性和data有区别吗?

问题2: 计算属性能不能用在输入框上, 双向绑定呢?

3.3_计算属性_完整写法

目标: 计算属性也是变量

- 语法:

```
computed: {  
  "属性名": {  
    set(值){  
    },  
    get() {  
      return "值"  
    }  
  }  
}
```

```
<template>  
  <div>  
    <div>  
      <span>名字:</span>  
      <input type="text" v-model="full">  
    </div>  
  </div>  
</template>
```

- 演示: 给计算属性赋值问题



小结

1. 何时用计算属性完整写法?

给计算属性变量赋值时

2. set函数和get函数什么执行?

set接收要赋予的值

get里要返回给这个计算属性具体值

3.4_案例-小选影响全选

案例 小选影响全选

需求: 小选框都选中(手选), 全选自动选中

效果演示:

全选: ☐ 反选

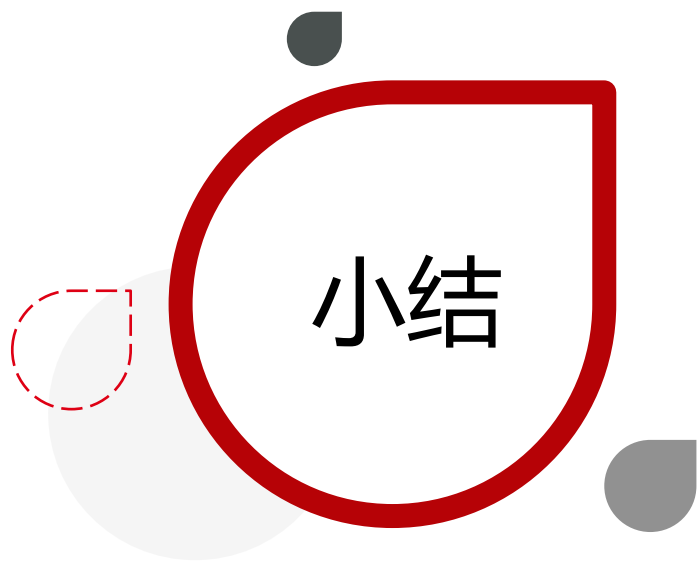
- ☐ 猪八戒
- ☐ 孙悟空
- ☐ 唐僧
- ☐ 白龙马



小选影响全选

分析:

- ① 先静态后动态, 从.md拿到静态标签和数据
- ② 循环生成复选框和文字, 对象的c属性和小选框的选中状态, 用v-model双向绑定
- ③ 定义isAll计算属性, 值通过小选框们统计c属性状态得来



小选框如何影响全选框选中状态的?

v-model给全选框绑定isAll计算属性

isAll计算属性里返回统计小选框选中状态结果



案例

全选影响小选

需求1: 获取到全选状态 – 改装isAll计算属性

需求2: 全选状态同步给所有小选框

效果演示:

全选: ☐ 反选

- ☐ 猪八戒
- ☐ 孙悟空
- ☐ 唐僧
- ☐ 白龙马



全选影响小选

分析:

- ①: isAll改成完整写法, set里获取到全选框, 勾选的状态值
- ②: 遍历数据数组, 赋给所有小选框v-model关联的属性



全选如何影响小选框选中状态的?

小选框v-model, 关联数组里对象的c属性

isAll计算属性的set方法, 拿到全选框状态

状态值赋给, 数组里每个对象的c属性

3.6_案例-反选

案例 反选

需求: 点击反选, 让所有小选框, 各自取相反勾选状态

效果演示:

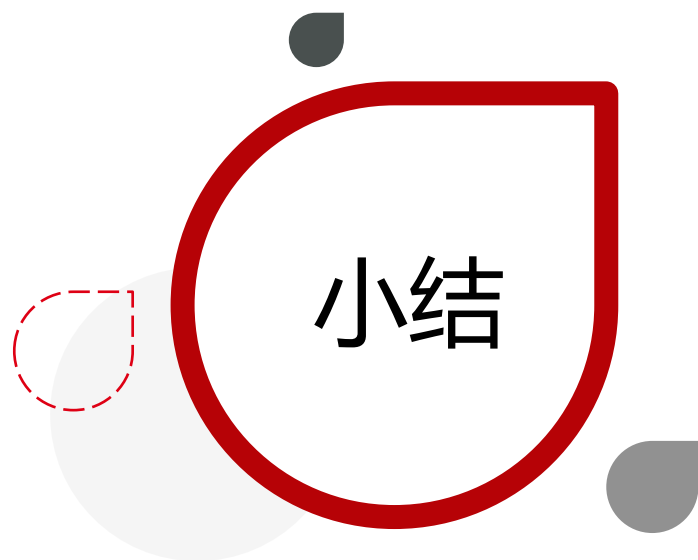
全选: ☐ 反选

- ☐ 猪八戒
- ☐ 孙悟空
- ☐ 唐僧
- ☐ 白龙马

案例 反选

分析:

- ①: 小选框的勾选状态, 在对象的c属性
- ②: 遍历所有对象, 把对象的c属性取相反值赋予回去即可



反选如何实现的?

遍历每个对象, 把c属性的值取反再赋予回去



目录

Contents

- ◆ Vue基础_更新监测, key作用
- ◆ Vue基础_过滤器
- ◆ Vue基础_计算属性
- ◆ Vue基础_侦听器



Vue基础_侦听器



问题: Vue能否侦听到某个变量的值改变了?

4.0_侦听器_watch

目标: 可以侦听data/computed属性值的改变

- 语法:

```
watch: {  
  "被侦听的属性名" (newVal, oldVal){  
  
  }  
}
```

```
<template>  
  <div>  
    <input type="text" v-model="name">  
  </div>  
</template>
```

```
</script>  
export default {  
  data(){  
    return {  
      name: ""  
    }  
  },  
  watch: {  
    name(newVal, oldVal){ // 当msg变量的值改变触发此函数  
      console.log(newVal, oldVal);  
    }  
  }  
}
```



如何侦听到某个变量值改变呢?

使用watch配置项, key是要侦听的data/计算属性名



问题1: 刚刚侦听的是基础类型?

问题2: 能否侦听对象?

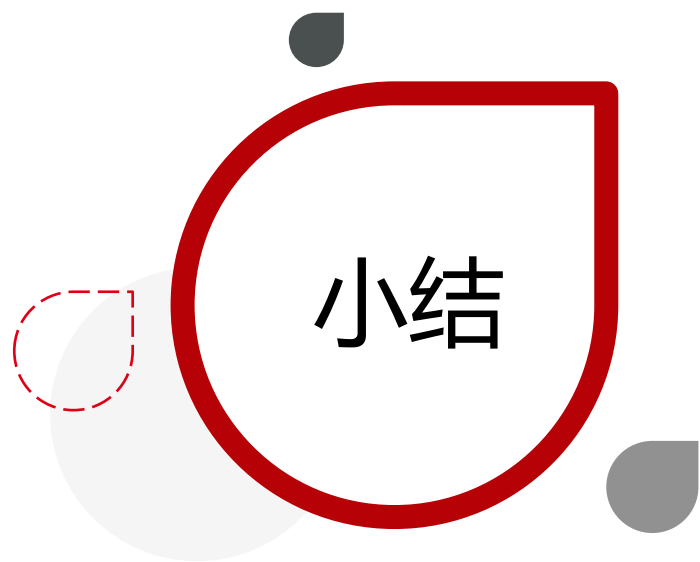
4.1_侦听器_深度侦听和立即执行

目标: 侦听复杂类型, 或者立即执行侦听函数

- 语法:

```
watch: {  
  "要侦听的属性名": {  
    immediate: true, // 立即执行  
    deep: true, // 深度侦听复杂类型内变化  
    handler (newVal, oldVal) {  
  
    }  
  }  
}
```

```
<template>  
  <div>  
    <input type="text" v-model="user.name">  
    <input type="text" v-model="user.age">  
  </div>  
</template>
```



1. 如何侦听一个对象/数组呢?

把侦听器写成对象形式, 给handler方法和deep:true

2. 侦听器函数马上执行?

immediate: true



案例

监听list变化, 同步到浏览器本地

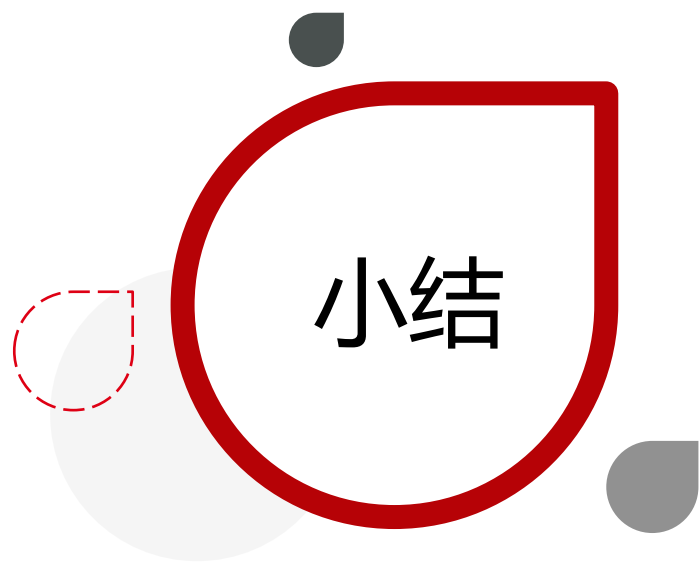
需求1: 把品牌管理的数据实时同步到本地缓存

分析:

- ① 在watch侦听list变化的时候, 把最新的数组list转成JSON字符串存入到localStorage本地
- ② data里默认把list变量从本地取值, 如果取不到给个默认的空数组

效果:

新增/删除 - 刷新页面 - 数据还在



涉及到了哪些知识点?

watch深度侦听数组, deep: true

localStorage本地缓存里只能存储字符串

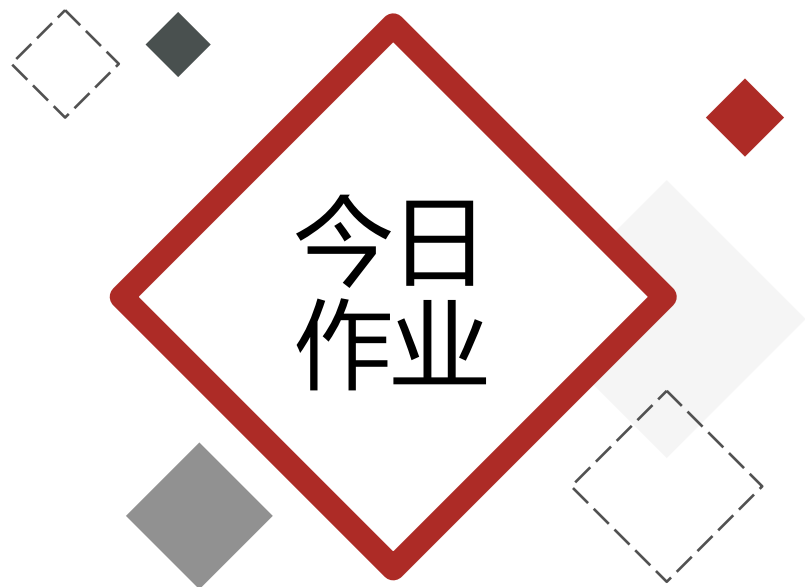
JSON.stringify和JSON.parse使用

逻辑或

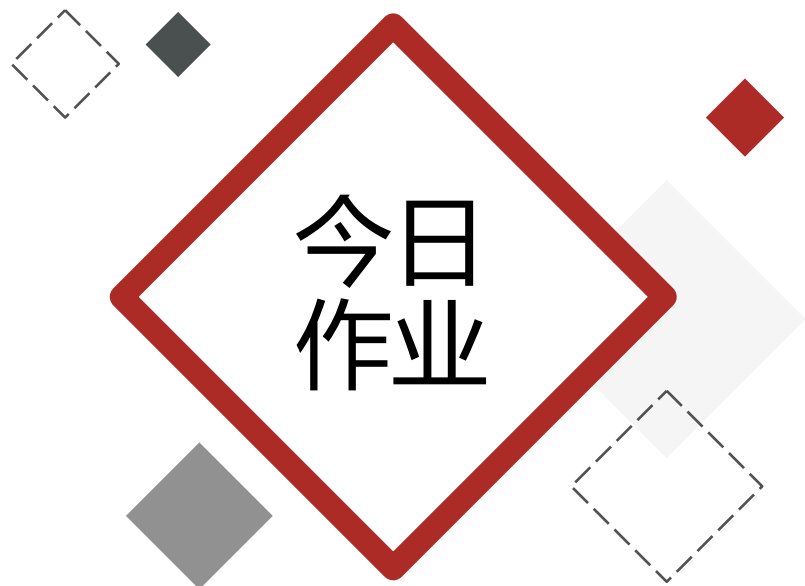


思考

1. v-for能监测到哪些数组方法改变, 更新页面
2. key作用是什么
3. 虚拟dom好处, diff结论
4. 动态设置class或style
5. Vue过滤器作用和分类
6. Vue计算属性作用
7. Vue侦听器的作用



在md文档最后, 还有一些练习, 练习后再挑战作业



1. 把今日课上的练习和案例来一遍(边看视频边写, 然后自己整理思路, 从0来一遍)
2. 移动端-导航切换效果(参考md里gif图)
3. 统计学生信息(参考md里gif图)



传智教育旗下高端IT教育品牌