

Vue训练营2

@大圣
8点正式开始



奖品: 前端知识图谱鼠标垫 ×10 份
06月11日 22:10 自动开奖



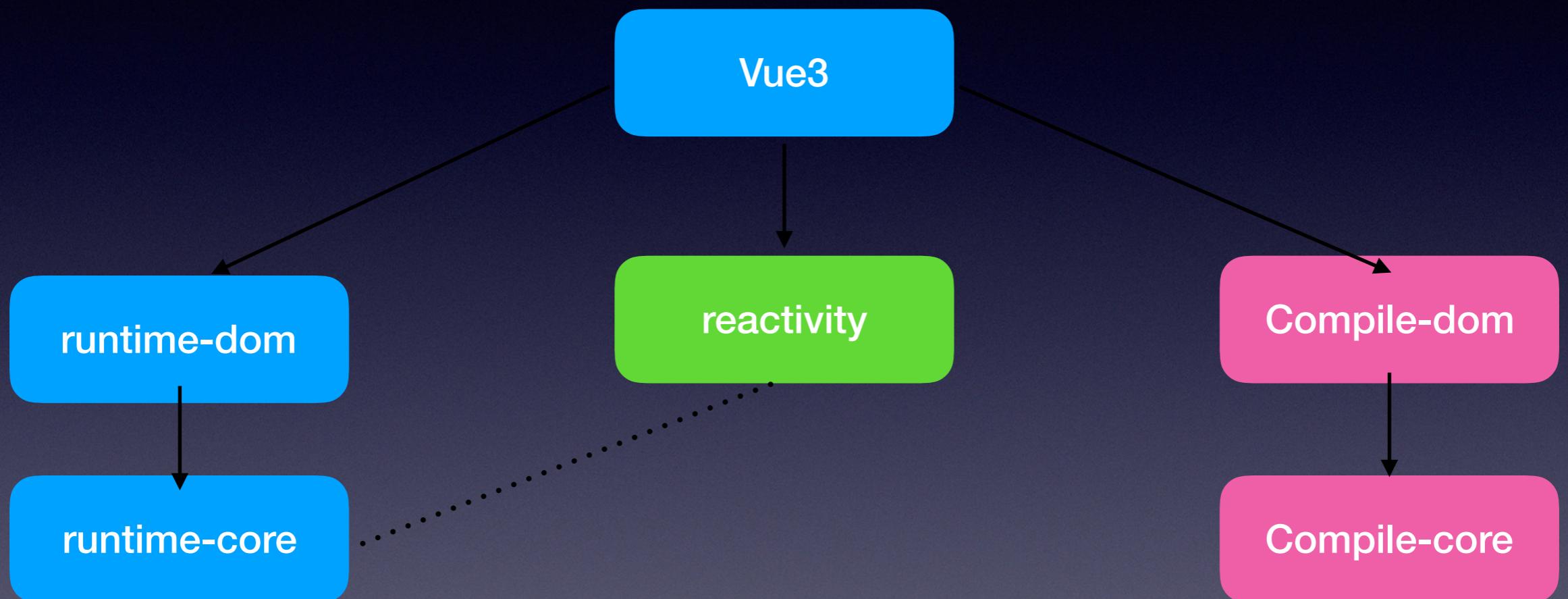
长按识别小程序，参与抽奖

抽奖说明: 抽奖结果微信通知，今晚 22 点 10 分
公布中奖结果，中奖学员按要求填写收货信息即
可。

今日任务

- 性能对比
- compiler
- 静态标记
- ssr对比
- 自定义renderer
- 贡献代码心得

Vue3模块





尤小右



2018-4-18 21:27 来自 微博国际版



☆ 收藏

42

40

154

Compiler

- compiler目的只有一个， 我们写的template 实际上是js， 返回vdom
- 追本溯源， 为什么vue/react需要编译

倔强青铜

- jquery dom查找修改 没啥模板的概念
- \$(xx).html(yy)

秩序白银

- Underscore bdtemplate 字符串模板
- <%= 标记变量
- <% 标记js语法

```
<!-- <script type="text/template" id="tpl"> -->
<script src="underscore.js"></script>
<script type="text/template" id="tpl">
  <ul class="list">
    <% _.each(obj, function(e, i, a){ %>
      <% if (i === 0) %>
        <li><%= e.name %>
      <% else if (i === a.length - 1) %>
        <li class="last-item"><%= e.name %></li>
      <% else %>
        <li><%= e.name %></li>
      <% } ) %>
    </ul>
  </script>

<script>
  var data = [{name: "开课吧大圣"}, {name: "最强王者"}];

  var compiled = _.template(document.getElementById("tpl").innerHTML);
  var html = compiled(data);
  // console.log(html)
  document.querySelector("div").innerHTML = html;
```

秩序白银

- 有数据变化，全量重新渲染即可
- 原理就是基本的字符串编译，拼接成js函数，然后 new Function即可

```
<script>
var tpl = `<% for(var i = 0; i < this.posts.length; i++) {
    var post = this.posts[i];
    if(!post.expert){
        <p>post is null</p>
    } else {
        <p><a href="#"><% post.expert %> at <% post.time %></a></p>
    }
}>`
```

```
function anonymous() {
    var r = [];
    for (var i = 0; i < this.posts.length; i++) {
        var post = this.posts[i];
        r.push("      ");
        if (!post.expert) {
            r.push("      <p>post is null</p>      ");
        } else {
            r.push("      <p><a href="#">");
            r.push(post.expert);
            r.push(" at ");
            r.push(post.time);
            r.push("</a></p>      ");
        }
        r.push("      ");
    }
    return r.join("");
}
</script>
```

荣耀王者黄金

- 每次重新render成本太高
- dom是万恶之源
- 按需更新

```
<body>
```

```
<div id="app">
  开课吧
</div>

<script>
  let ele = document.getElementById('app')
  let arr = []
  for(let k in ele){
    arr.push(k)
  }
  console.dir(arr.length,arr.join(' '))
  // 一个简单的dom节点，竟然又200+属性，怪不的慢
</script>
</body>
```

```
// 其实用简单的对象就可以描述
let vdom = {
  type:"div",
  props:{ id:'app' },
  children:['开课吧']
}
```

align title lang translate dir dataset hidden tabIndex accessKey draggable spellcheck autocapitalize contentEditable isContentEditable inputMode offsetParent offsetTop offsetLeft offsetWidth offsetHeight style innerText outerText oncopy oncut onpaste onabort onblur oncancel oncanplay oncanplaythrough onchange onclick onclose oncontextmenu oncuechange ondblclick ondrag ondragend ondragenter ondragleave ondragover ondragstart ondrop ondurationchange onemptied onended onerror onfocus oninput oninvalid onkeydown onkeypress onkeyup onload onloadeddata onloadedmetadata onloadstart onmousedown onmouseenter onmouseleave onmousemove onmouseout onmouseover onmouseup onmousewheel onpause onplay onplaying onprogress onratechange onreset onresize onscroll onseeked onseeking onselect onstalled onsubmit onsuspend ontimeupdate ontoggle onvolumechange onwaiting onwheel onauxclick ongotpointercapture onlostpointercapture onpointerdown onpointermove onpointerup onpointercancel onpointerover onpointerout onpointerenter onpointerleave onselectstart onselectionchange nonce click focus blur namespaceURI prefix localName tagName id className classList slot attributes shadowRoot assignedSlot innerHTML outerHTML scrollTop scrollLeft scrollWidth scrollHeight clientTop clientLeft clientWidth clientHeight attributeStyleMap onbeforecopy onbeforecut onbeforepaste onsearch previousElementSibling nextElementSibling children firstElementChild lastElementChild childElementCount onfullscreenchange onfullscreenerror onwebkitfullscreenchange onwebkitfullscreenerror setPointerCapture releasePointerCapture hasPointerCapture hasAttributes getAttributeNames getAttribute getAttributeNS setAttribute setAttributeNS removeAttribute removeAttributeNS hasAttribute hasAttributeNS toggleAttribute getAttributeNode getAttributeNodeNS setAttributeNode setAttributeNodeNS removeAttributeNode closest matches webkitMatchesSelector attachShadow getElementsByTagName getElementsByTagNameNS getElementsByClassName insertAdjacentElement insertAdjacentText insertAdjacentHTML requestPointerLock getClientRects getBoundingClientRect scrollIntoView scrollTo scrollBy scrollIntoViewIfNeeded animate computedStyleMap before after replaceWith remove prepend append querySelector querySelectorAll requestFullscreen webkitRequestFullScreen webkitRequestFullscreen part createShadowRoot getDestinationInsertionPoints ELEMENT_NODE ATTRIBUTE_NODE TEXT_NODE CDATA_SECTION_NODE ENTITY_REFERENCE_NODE ENTITY_NODE PROCESSING_INSTRUCTION_NODE COMMENT_NODE DOCUMENT_NODE DOCUMENT_TYPE_NODE DOCUMENT_FRAGMENT_NODE NOTATION_NODE DOCUMENT_POSITION_DISCONNECTED DOCUMENT_POSITION_PRECEDING DOCUMENT_POSITION_FOLLOWING DOCUMENT_POSITION_CONTAINS DOCUMENT_POSITION_CONTAINED_BY DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC nodeType nodeName baseURI isConnected ownerDocument parentNode parentElement childNodes firstChild lastChild previousSibling nextSibling nodeValue textContent hasChildNodes getRootNode normalize cloneNode isEqualNode isSameNode compareDocumentPosition contains lookupPrefix lookupNamespaceURI isDefaultNamespace insertBefore appendChild replaceChild removeChild addEventListener removeEventListener dispatchEvent

精细化分析

- 使用js对象描述标签
- 数据修改，不再全量覆盖，而是js先进行一次 dom diff，然后算出需要修改最小的部分
- 相当于在template和浏览器之间，加了一层缓存

```
// 创建虚拟DOM
function createElement(type, props, children){
  return { type, props, children }
}

function render(dom) {
  let el = document.createElement(dom.type)
  for (let key in dom.props) {
    el.setAttribute(key, dom.props[key])
  }
  dom.children.forEach(child => {
    child = (typeof child=='object') ? render(child) : document.createTextNode(child)
    el.appendChild(child)
  })
  return el
}

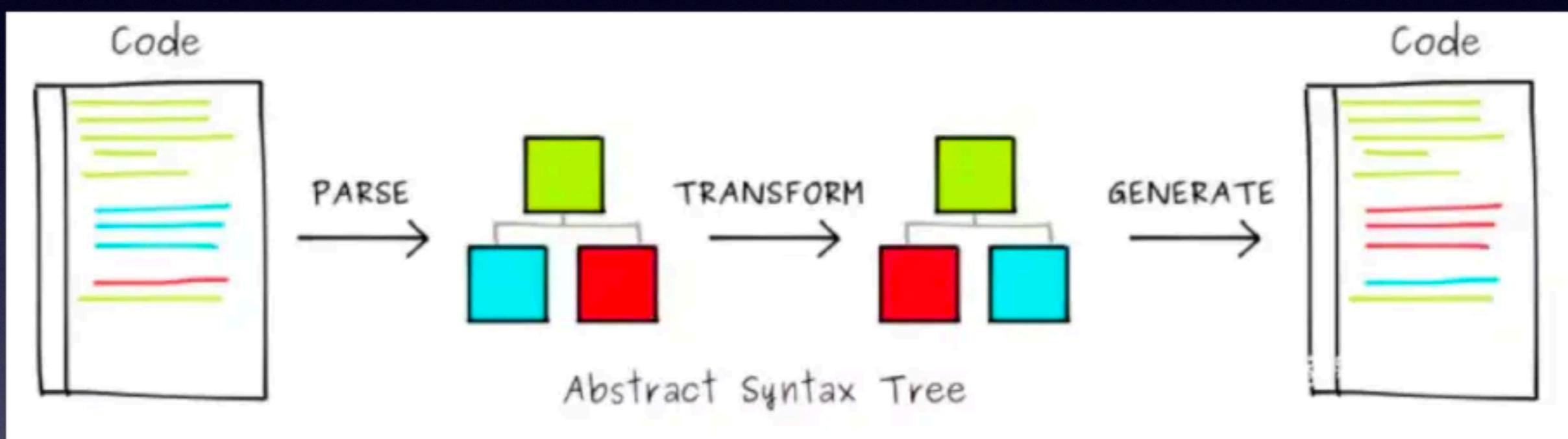
let vdom = createElement(['ul', {class:'list'}, [
  createElement('li', {class:'item'}, ['item1']),
  createElement('li', {class:'item'}, ['item2']),
  createElement('li', {class:'item'}, ['item3'])
]])
var el = render(vdom)
document.body.appendChild(el)
```

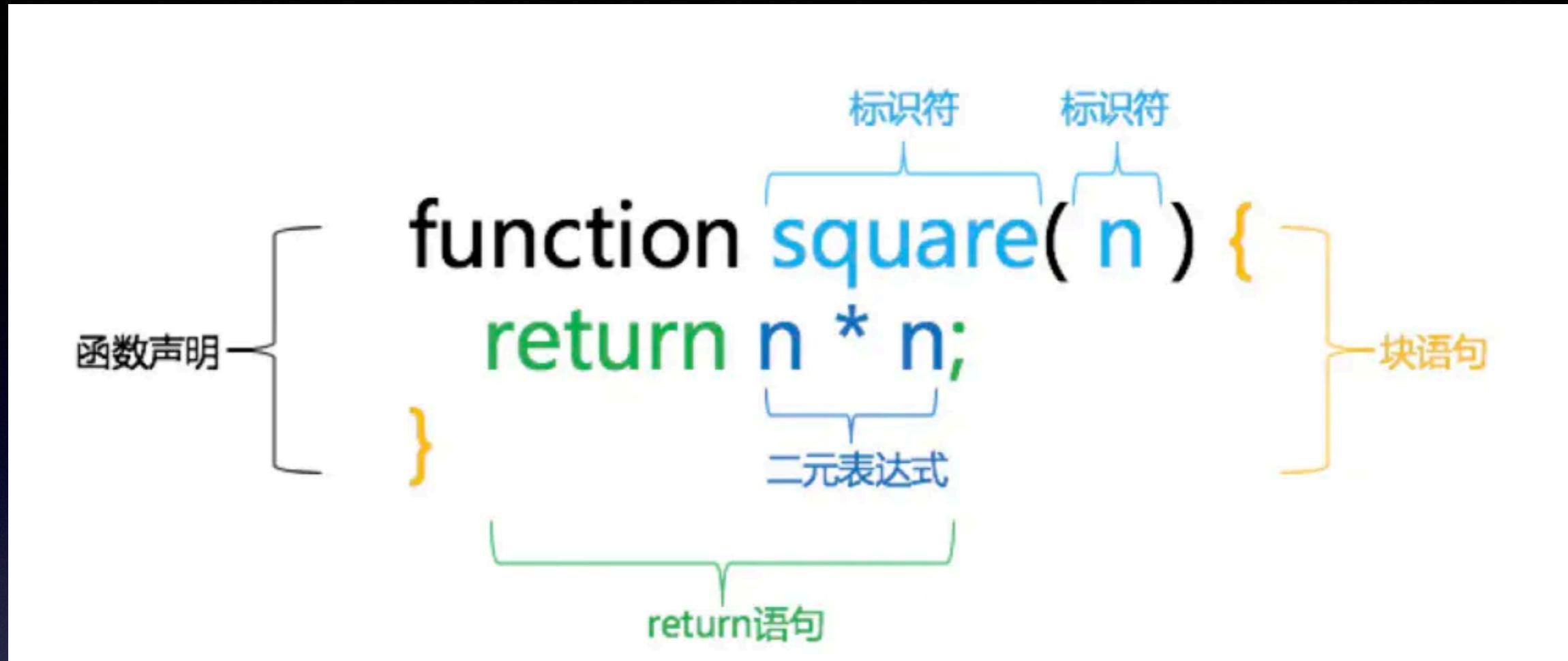
問題

- 用createElement创建元素 写起来太痛苦了，所以 template=> vdom机制，应运而生
- <https://template-explorer.vuejs.org/>
- Vue2解析结果

编译

- 写的依然是html，实际执行的是javascript
- template=> render函数，经典编译原理
- ast=> transform=>generate





```
<div> 开始  
  <input v-model="name" /> 自闭和  
  <h1>天气不错</h1> 静态标签  
开始<p @click="handleChild"> {{name}} </p> 结束  
  </div> 事件  
结束  
  
</template>
```

```
const { code } = compile(template, {
  hoistStatic: true,
  onError(err: CompilerError) {
    if (__DEV__) {
      const message = `Template compilation error: ${err.message}`
      const codeFrame =
        err.loc &&
        generateCodeFrame(
          template as string,
          err.loc.start.offset,
          err.loc.end.offset
        )
      warn(codeFrame ? `${message}\n${codeFrame}` : message)
    } else {
      throw err
    }
  },
  ...options
})
```

```
// The wildcard import results in a huge object with every export
// with keys that cannot be mangled, and can be quite heavy size-wise.
// In the global build we know `Vue` is available globally so we can avoid
// the wildcard object.
const render = (__GLOBAL__
  ? new Function(code)()
  : new Function('Vue', code)(runtimeDom)) as RenderFunction
return (compileCache[key] = render)
```

Compile

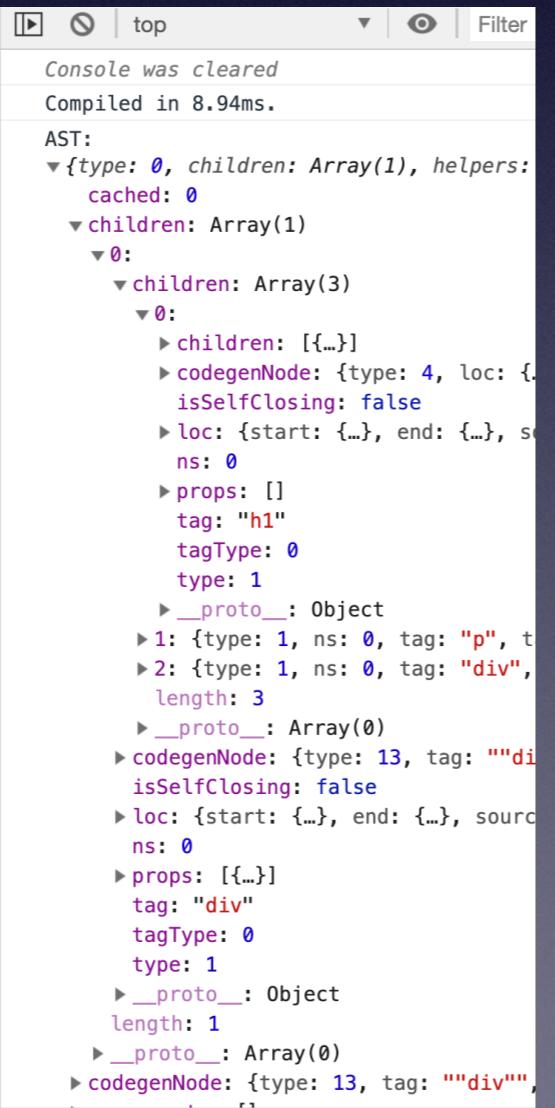
baseCompile

baseParse

Template

```
<div id="app">
  <h1>技术摸鱼</h1>
  <p>今天天气真不错</p>
  <div :id="userid">{{name}}</div>
</div>
```

AST



The screenshot shows the Vue DevTools interface with the 'AST' tab selected. The console output indicates the template was compiled in 8.94ms. The AST tree is displayed, showing the hierarchical structure of the components and their properties. The root node is a component with children: a h1 node, a p node, and a div node with a dynamic id and a template expression.

```
Console was cleared
Compiled in 8.94ms.

AST:
▼ {type: 0, children: Array(1), helpers: 
  cached: 0
  ▼ children: Array(1)
    ▼ 0:
      ▼ children: Array(3)
        ▼ 0:
          ▶ children: [{}]
          ▶ codegenNode: {type: 4, loc: {}, 
            isSelfClosing: false
            loc: {start: {...}, end: {...}, source: "...", ns: 0}
            props: []
            tag: "h1"
            tagType: 0
            type: 1
            ▶ __proto__: Object
          ▶ 1: {type: 1, ns: 0, tag: "p", type: 1, 
            isSelfClosing: false
            loc: {start: {...}, end: {...}, source: "...", ns: 0}
            props: []
            tag: "p"
            tagType: 0
            type: 1
            ▶ __proto__: Object
          ▶ 2: {type: 1, ns: 0, tag: "div", type: 1, 
            isSelfClosing: false
            loc: {start: {...}, end: {...}, source: "...", ns: 0}
            props: []
            tag: "div"
            tagType: 0
            type: 1
            ▶ __proto__: Object
          ▶ __proto__: Array(0)
        ▶ codegenNode: {type: 13, tag: ""div"", 
          isSelfClosing: false
          loc: {start: {...}, end: {...}, source: "...", ns: 0}
          props: []
          tag: "div"
          tagType: 0
          type: 1
          ▶ __proto__: Object
        ▶ length: 1
        ▶ __proto__: Array(0)
      ▶ codegenNode: {type: 13, tag: ""div""}, 
        isSelfClosing: false
        loc: {start: {...}, end: {...}, source: "...", ns: 0}
        props: []
        tag: "div"
        tagType: 0
        type: 1
        ▶ __proto__: Object
      ▶ __proto__: Array(0)
    ▶ codegenNode: {type: 13, tag: ""div""}, 
      isSelfClosing: false
      loc: {start: {...}, end: {...}, source: "...", ns: 0}
      props: []
      tag: "div"
      tagType: 0
      type: 1
      ▶ __proto__: Object
    ▶ __proto__: Array(0)
  ▶ codegenNode: {type: 13, tag: ""div""}, 
    isSelfClosing: false
    loc: {start: {...}, end: {...}, source: "...", ns: 0}
    props: []
    tag: "div"
    tagType: 0
    type: 1
    ▶ __proto__: Object
  ▶ __proto__: Array(0)
}
```

v-model

V-on

V-for

静态提升

Transform

Generate

```
import { createVNode as _createVNode, toDisplayString as 
_toDisplayString, openBlock as _openBlock, createBlock as 
_createBlock } from "vue"

const _hoisted_1 = { id: "app" }
const _hoisted_2 = /*#__PURE__*/_createVNode("h1", null, "技术摸鱼", -1 /* HOISTED */)
const _hoisted_3 = /*#__PURE__*/_createVNode("p", null, "今天天气真不错", -1 /* HOISTED */)

export function render(_ctx, _cache) {
  return (_openBlock(), _createBlock("div", _hoisted_1, [
    _hoisted_2,
    _hoisted_3,
    _createVNode("div", { id: _ctx.userid }, _toDisplayString(_ctx.name), 9 /* TEXT, PROPS */, ["id"])
  ]))
}

// Check the console for the AST
```

```
const ast = isString(template) ? baseParse(template, options) : template
const [nodeTransforms, directiveTransforms] = getBaseTransformPreset(
  prefixIdentifiers
)
transform(ast, {
  ...options,
  prefixIdentifiers,
  nodeTransforms: [
    ...nodeTransforms,
    ...(options.nodeTransforms || []) // user transforms
  ],
  directiveTransforms: {
    ...directiveTransforms,
    ...(options.directiveTransforms || {}) // user transforms
  }
})

return generate(ast, {
  ...options,
  prefixIdentifiers
})
```

parse

baseParser

parseChildren

动态文本

截止标签

开始标签

普通文本

标签属性

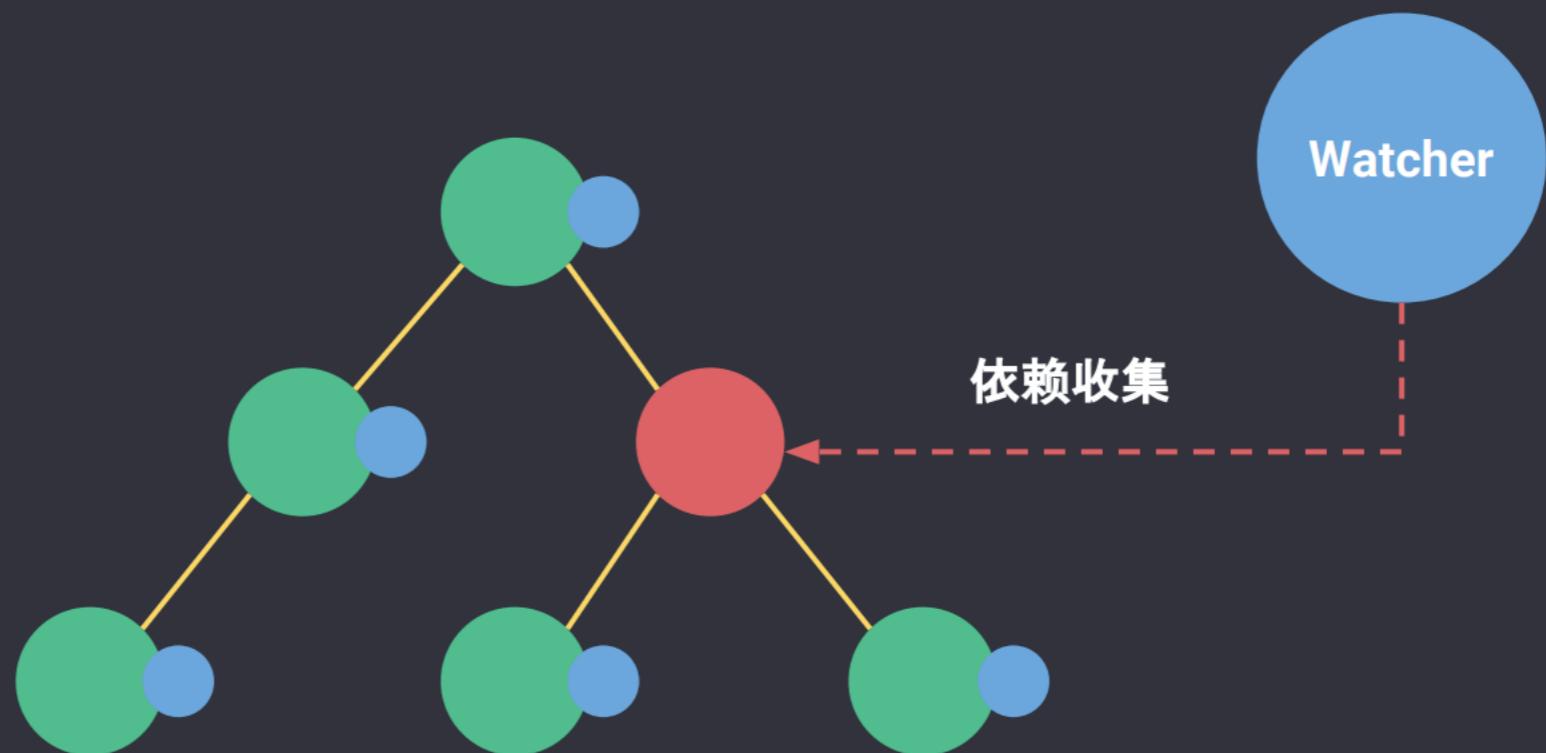
```
node = parseInterpolation(context, mode)
} else if (mode === TextModes.DATA && s[0] === '<') {
    // https://html.spec.whatwg.org/multipage/parsing.html#tag-open-state
    if (s.length === 1) {
        emitError(context, ErrorCode.EOF_BEFORE_TAG_NAME, 1)
    } else if (s[1] === '!') {
        // https://html.spec.whatwg.org/multipage/parsing.html#markup-declaration-open-state
        if (startsWith(s, '<!--')) {
            node = parseComment(context)
        } else if (startsWith(s, '<!DOCTYPE')) {
            // Ignore DOCTYPE by a limitation.
            node = parseBogusComment(context)
        } else if (startsWith(s, '<![CDATA[')) {
            if (ns !== Namespaces.HTML) {
                node = parseCDATA(context, ancestors)
            } else {
                emitError(context, ErrorCode.CDATA_IN_HTML_CONTENT)
                node = parseBogusComment(context)
            }
        } else {
            emitError(context, ErrorCode.INCORRECTLY_OPENED_COMMENT)
            node = parseBogusComment(context)
        }
    } else if (s[1] === '/') {
        // https://html.spec.whatwg.org/multipage/parsing.html#end-tag-open-state
        if (s.length === 2) {
            emitError(context, ErrorCode.EOF_BEFORE_TAG_NAME, 2)
        } else if (s[2] === '>') {
            emitError(context, ErrorCode.MISSING_END_TAG_NAME, 2)
            advanceBy(context, 3)
            continue
        } else if (/^[a-z]/i.test(s[2])) {
            emitError(context, ErrorCode.X_INVALID_END_TAG)
            parseTag(context, TagType.End, parent)
            continue
        } else {
            emitError(
                context,
                ErrorCode.INVALID_FIRST_CHARACTER_OF_TAG_NAME,
                2
            )
            node = parseBogusComment(context)
        }
    } else if (/^[a-z]/i.test(s[1])) {
        node = parseElement(context, ancestors)
    } else if (s[1] === '?') {
        emitError(
            context,
            ErrorCode.UNEXPECTED_QUESTION_MARK_INSTEAD_OF_TAG_NAME,
            1
        )
    }
}
```

Vdom diff

- Dom diff 目的是算出最小的修改
- 举个栗子

Vue3 vdom

传统 vdom 的性能瓶颈



- 虽然 Vue 能够保证触发更新的组件最小化, 但在单个组件内部依然需要遍历该组件的整个 vdom 树

传统 vdom 的性能瓶颈

```
<template>
  <div id="content">
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">{{ message }}</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
  </div>
</template>
```

- Diff <div>
 - Diff props of <div>
 - Diff children of <div>
 - Diff <p>
 - Diff props of <p>
 - Diff children of <p>
 - Repeat n times...
- 传统 vdom 的性能跟模版大小正相关，跟动态节点的数量无关。在一些组件整个模版内只有少量动态节点的情况下，这些遍历都是性能的浪费

传统 vdom 的性能瓶颈

```
function render() {
  const children = []
  for (let i = 0; i < 5; i++) {
    children.push(h('p', {
      class: 'text'
    }, i === 2 ? this.message : 'Lorum ipsum'))
  }
  return h('div', { id: 'content' }, children)
}
```

- 根本原因:JSX 和手写的 render function 是完全动态的, 过度的灵活性导致运行时可以用于优化的信息不足

最简单的情况

```
<template>
  <div id="content">
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">{{ message }}</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
  </div>
</template>
```

- Diff <p> textContent

- 节点结构完全不会改变
- 只有一个动态节点

节点结构变化:v-if

```
<template>
  <div>
    <p class="text">Lorem ipsum</p>
    <p v-if="ok">
      <span>Lorem ipsum</span>
      <span>{{ message }}</span>
    </p>
  </div>
</template>
```

- Check <p v-if="ok">
 - Diff textContent

- v-if 外部:只有 v-if 是动态节点
- v-if 内部:只有 {{ message }} 是动态节点

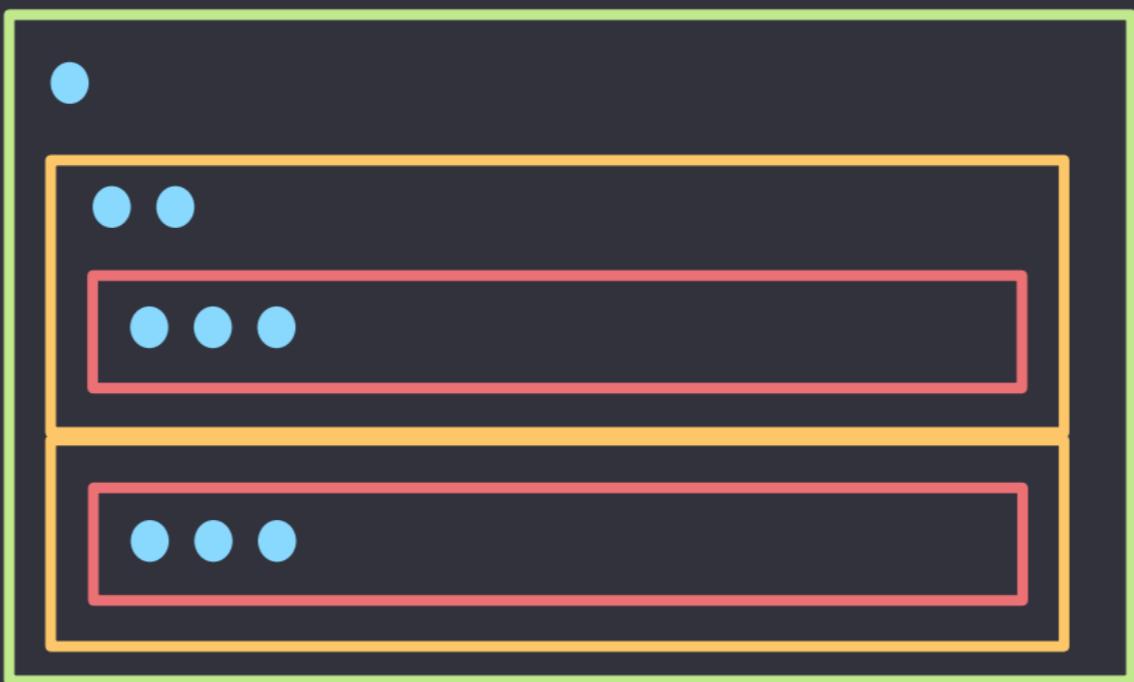
节点结构变化:v-for

```
<template>
  <div>
    <p class="text">Lorem ipsum</p>
    <p v-for="item of list">
      <span>Lorem ipsum</span>
      <span>{{ item.message }}</span>
    </p>
  </div>
</template>
```

- Diff <p v-for> children
 - Diff textContent
 - Repeat n times...

- v-for 外部: 只有 v-for 是动态节点 (fragment)
- 每个 v-for 循环内部: 只有 {{ item.message }} 是动态节点

Block tree “区块树”



- 将模版基于动态节点指令切割为嵌套的区块
- 每个区块内部的节点结构是固定的
- 每个区块只需要以一个 Array 追踪自身包含的动态节点

Before

```
<template>
  <div id="content">
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">{{ message }}</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
  </div>
</template>
```

- Diff <div>
 - Diff props of <div>
 - Diff children of <div>
 - Diff <p>
 - Diff props of <p>
 - Diff children of <p>
 - Repeat n times...

After

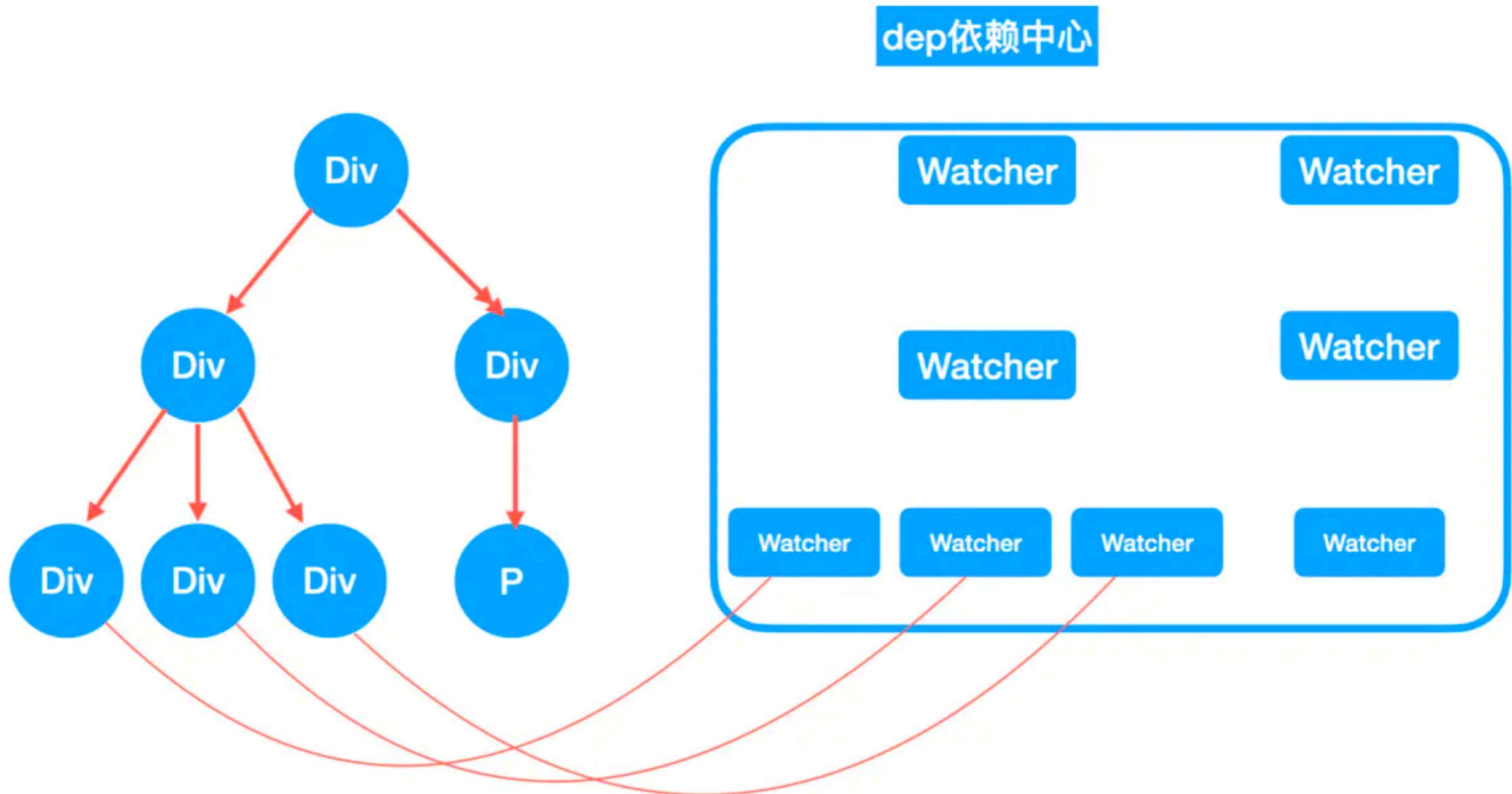
```
<template>
  <div id="content">
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">{{ message }}</p>
    <p class="text">Lorem ipsum</p>
    <p class="text">Lorem ipsum</p>
  </div>
</template>
```

- Diff <p> textContent

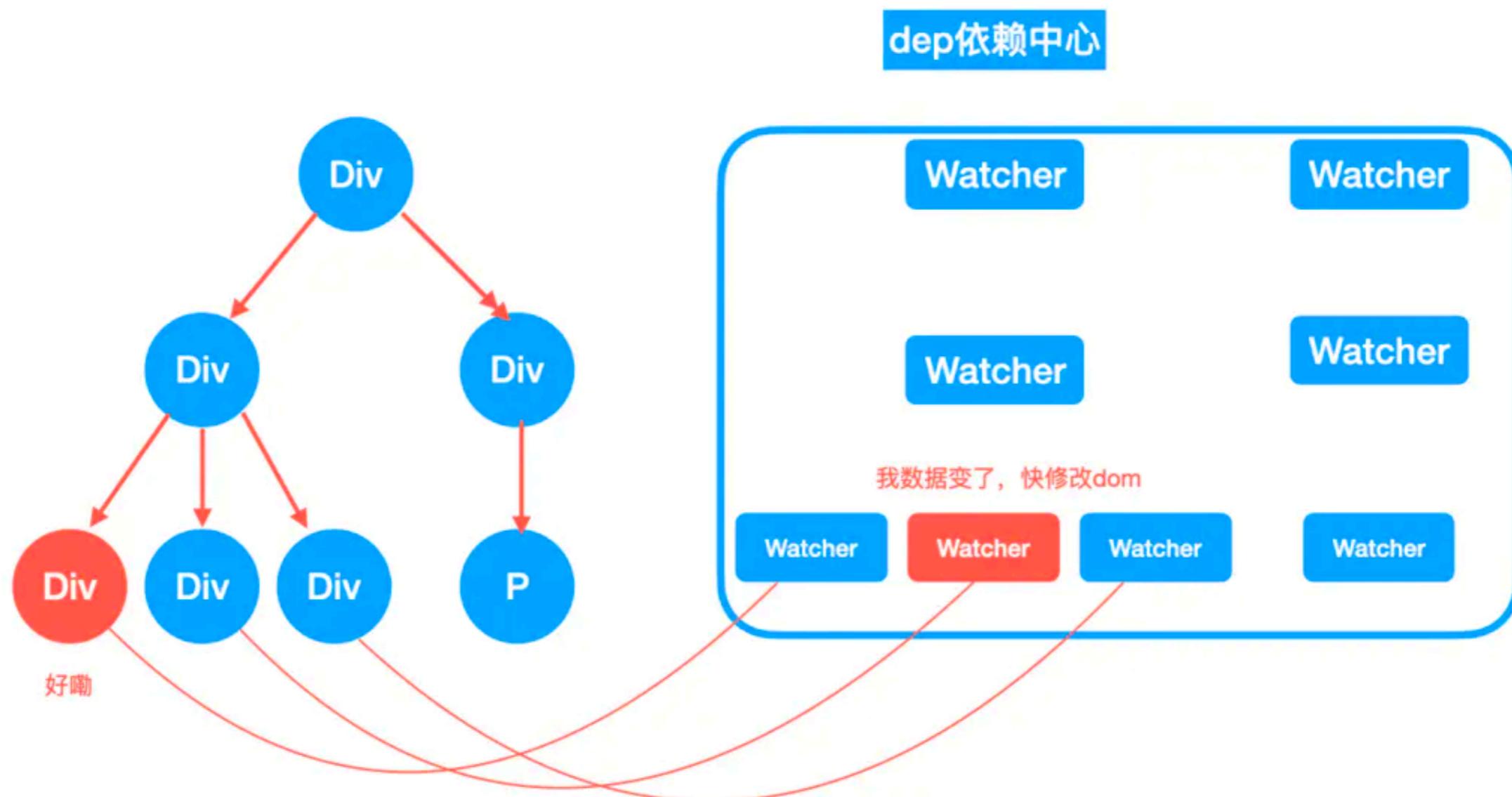
Vdom静态标记

- 编译时优化（灵感来源prepack.io）
- 静态标记
- vdom发展史（Vue3为啥快）
- 在vue2静态标记的基础上，动态标签内部的静态标签，使用block来标记

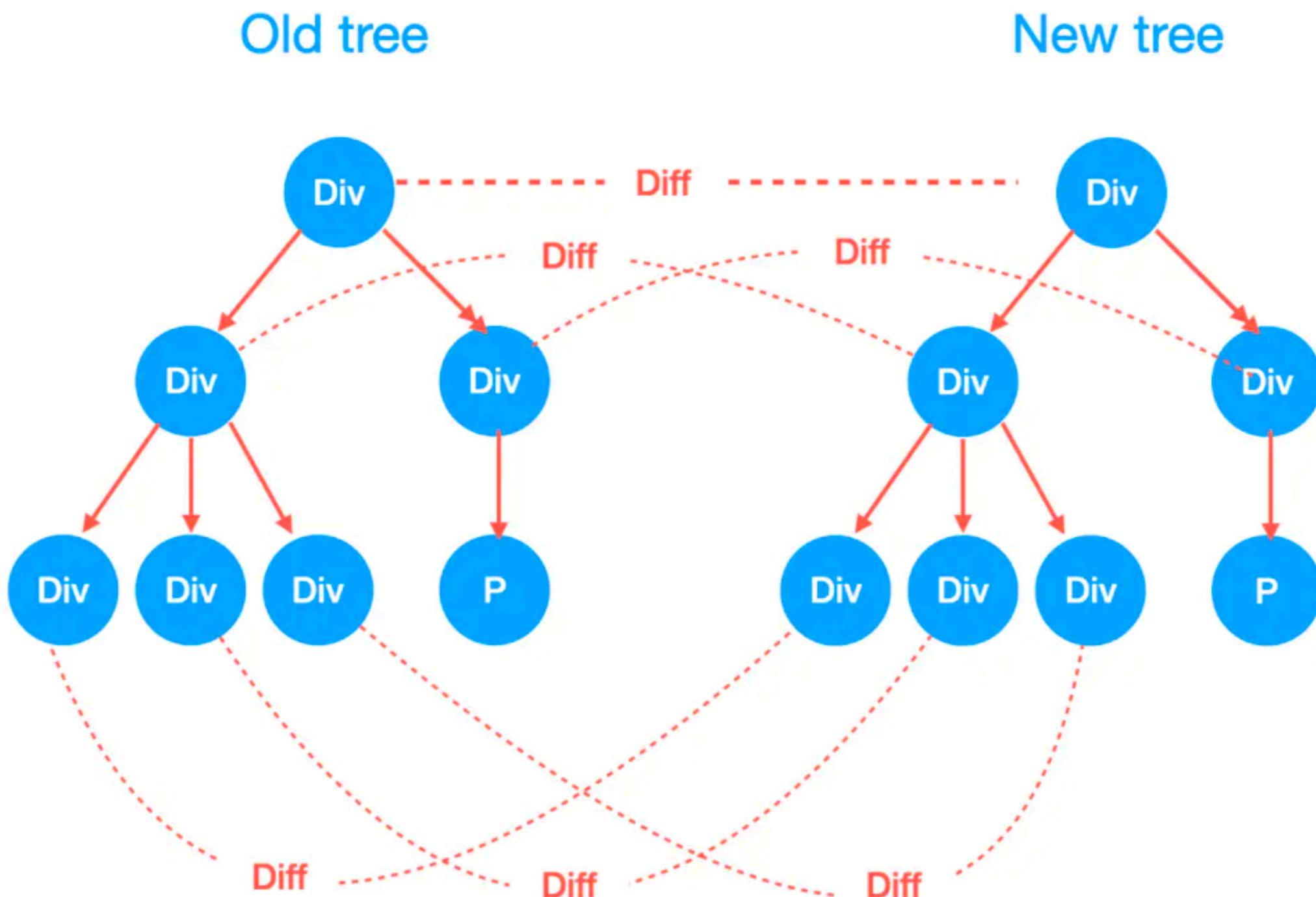
Vue1.x



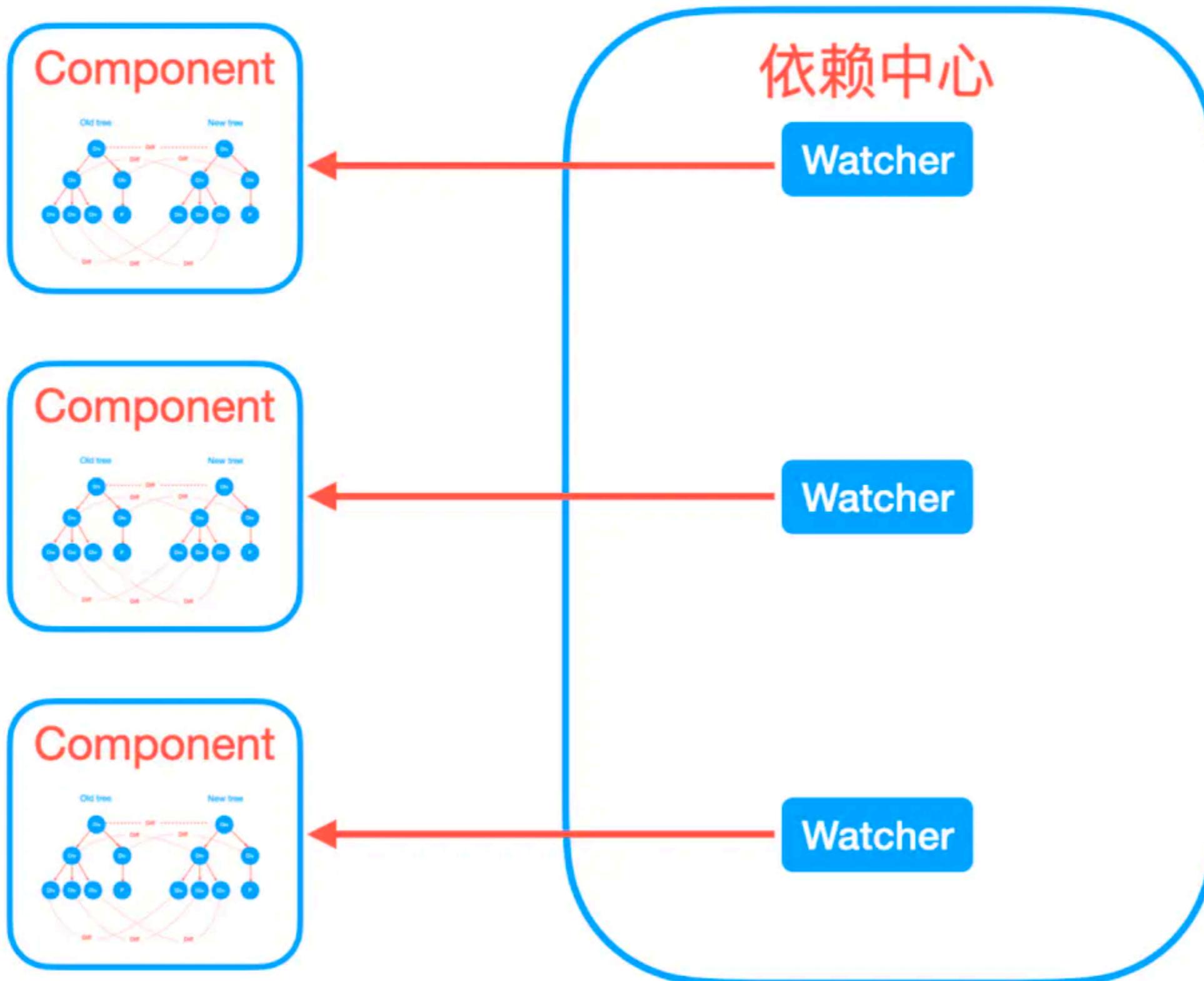
Vue1.x



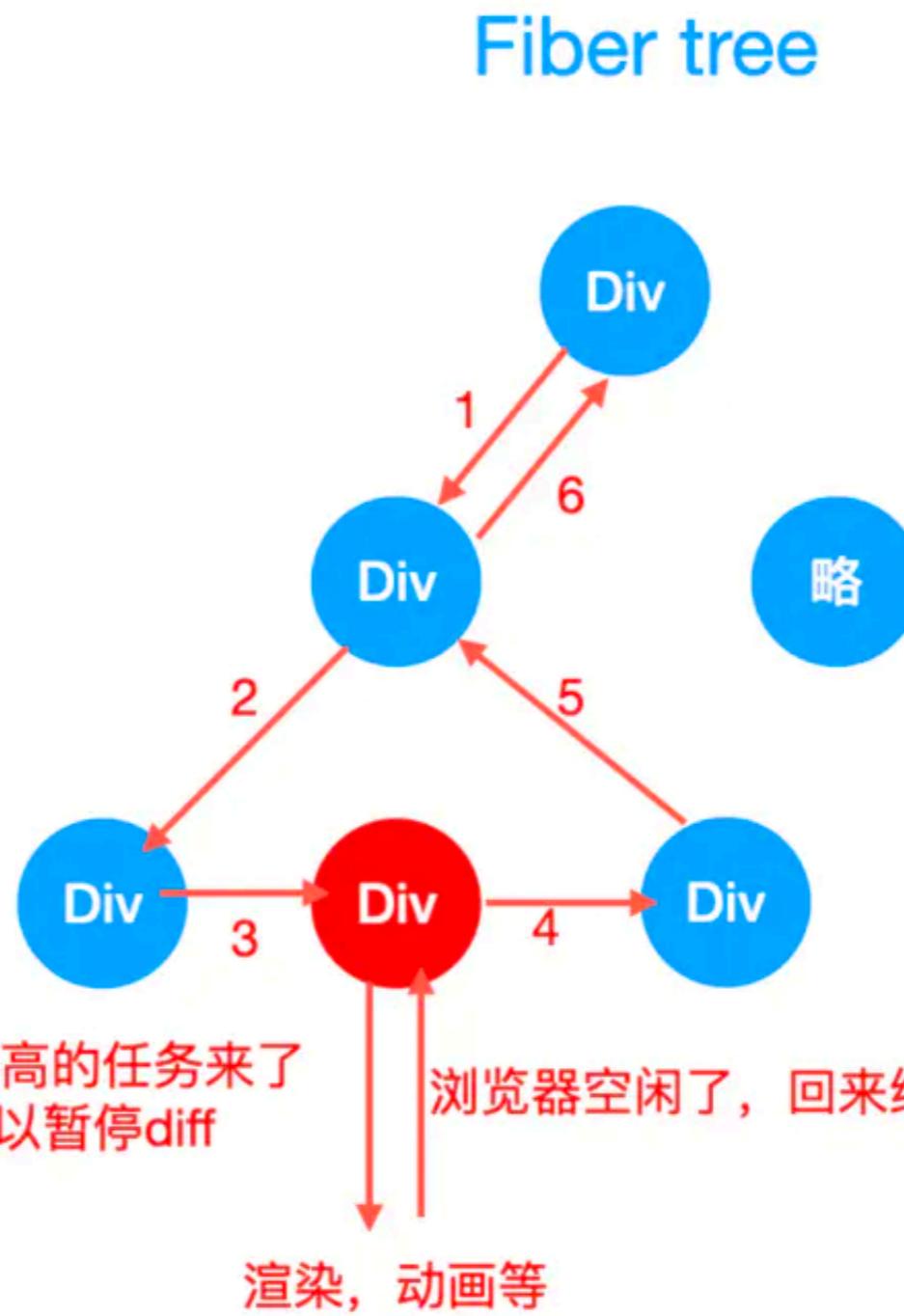
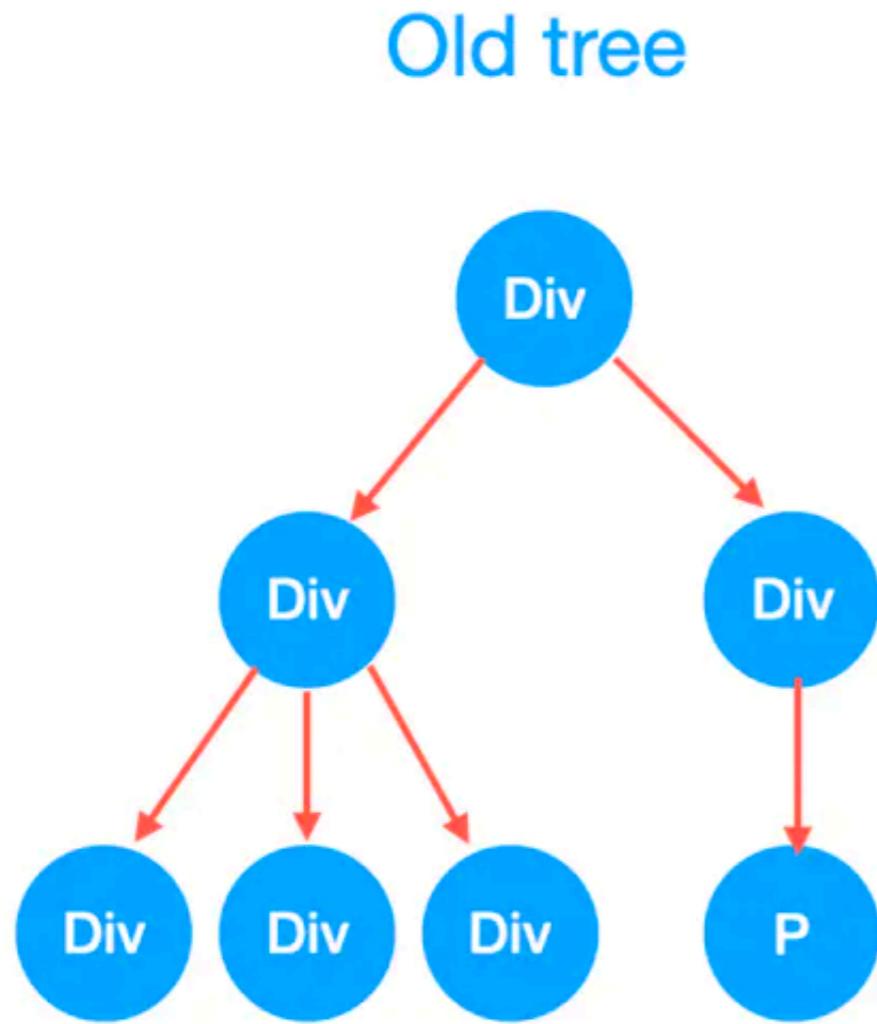
React15



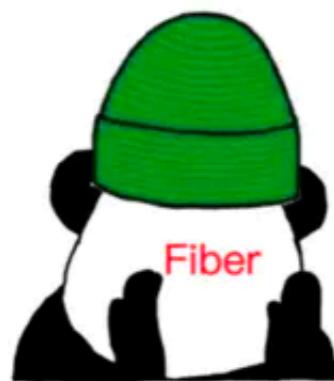
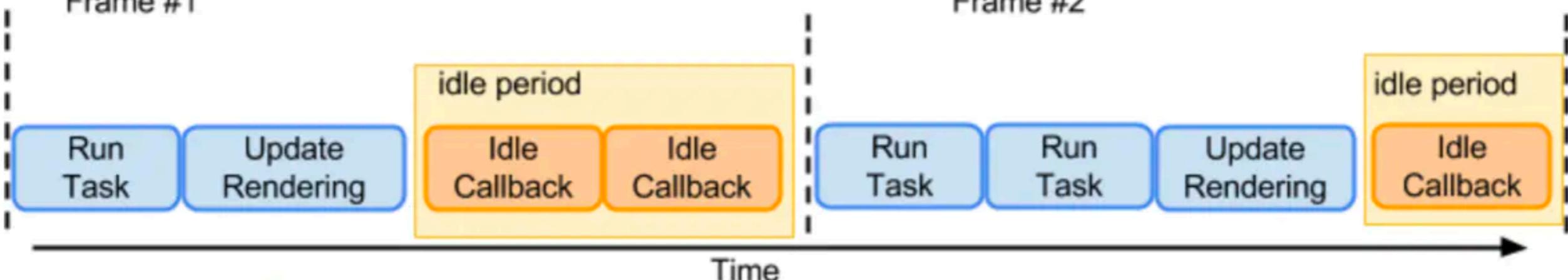
Vue2.x



React Fiber



Frame #1



女神有空了
快计算diff



优先级高的富二代来了
比如动画，用户输入
把女神换回去

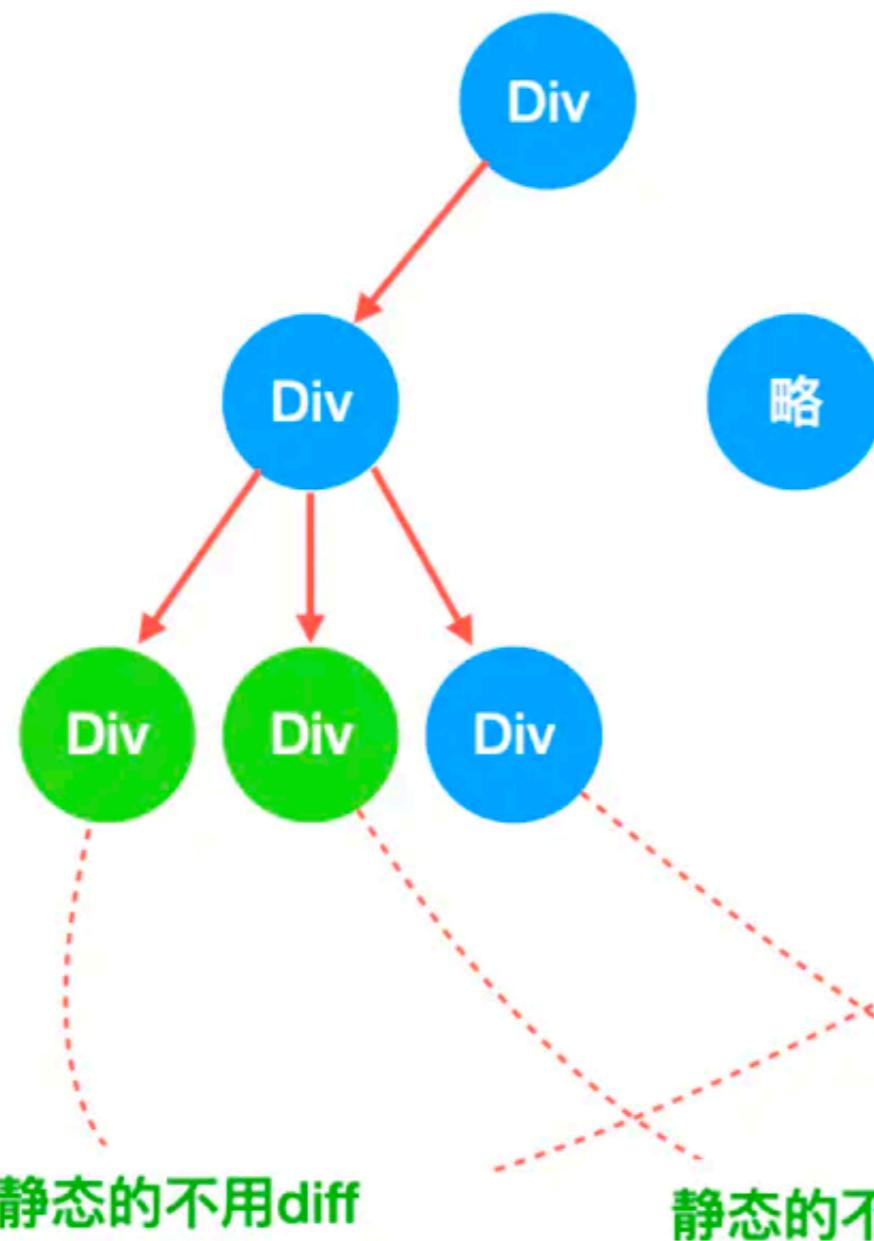


女神有空了
继续计算

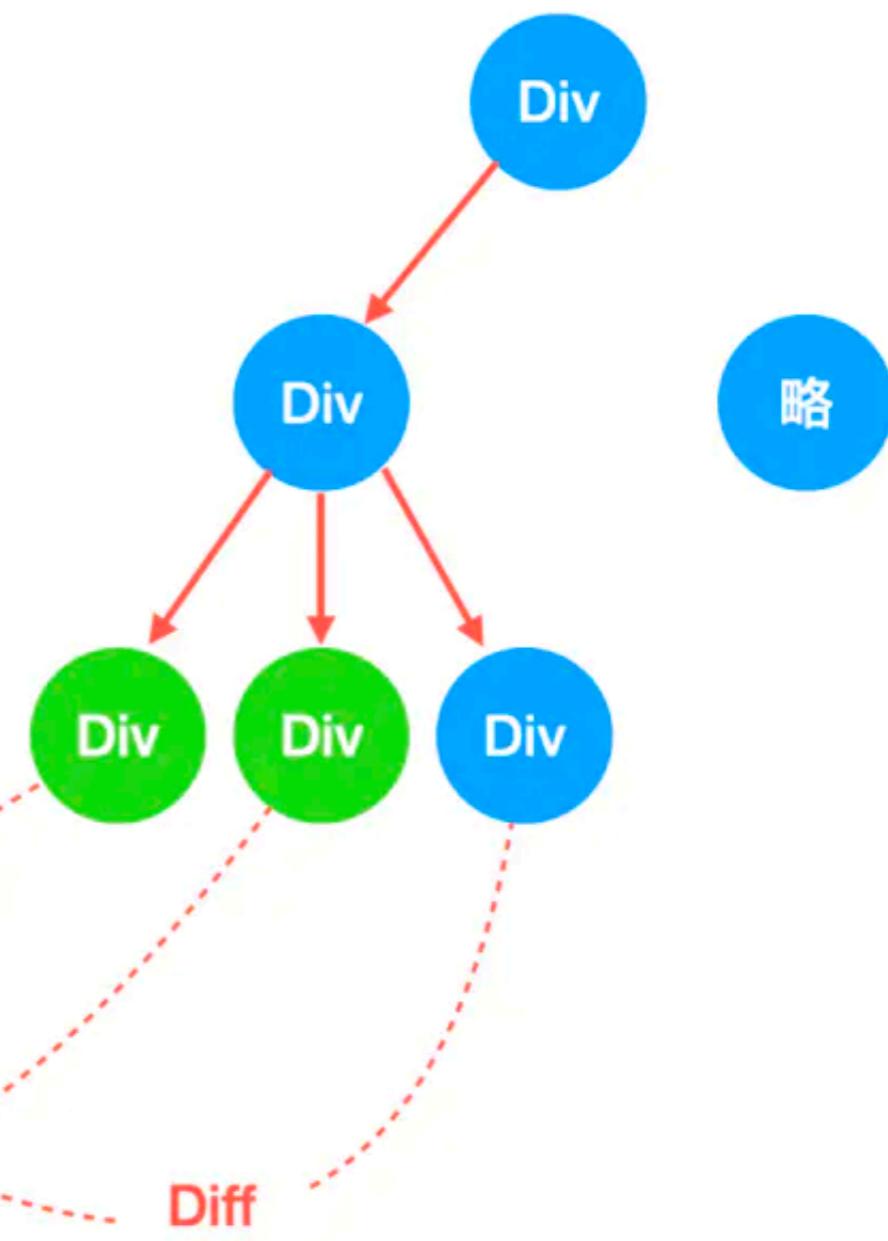


Vue3.0

Old tree



New tree



深夜话题

- 35岁危机
- 学习 VS 加班
- 成长经历
- 如何描述做的项目
- 早起看书
- vue3提源码

广告时间

- 奖学金班级！！！ 好好学习，全额返钱 疫情
- 新增热点技术



授课方式

直播

课程大纲

晚上8:30~10:30

直播互动答疑

上课听讲为主

100课时

训练营

独立知识 + 学员需求

针对某个话题的刻意训练

小群+强制交作业+奖励

录播

强制刷题 / 写作业

手写React ssr框架

算法刷题

浏览器原理

项目实战

网络协议

长期待续....

全栈架构师 P6

新内容

微前端

前端基建

团队管理

软技能

终身学习

大厂思维

Web安全实战

算法和数据结构

前端架构师体系构建

前端自动化测试

javascript设计模式

面试技巧

广度

小程序 + 云开发

Flutter实战

微信公众号

ReactNative实战

Webpack工程化

Docker部署全栈应用

深度

Vuejs全家桶源码

手写Eggjs MVC框架

Webpack原理

React16全家桶源码

小程序云开发实现支付

手写常见框架源码

实战

Vue全家桶企业级实战

Nodejs项目实战

Kkb-cli脚手架开发

React全家桶企业级实战

Mysql & Mongodb

TypeScript



Vue

Vue组件化设计

手写Vuex

Vue源码整体剖析

手写element表单

手写Vue-router

响应式原理

手写element弹窗

TypeScript

模板编译+虚拟dom

权限控制

自动化测试

双向绑定原理

动态路由

服务端渲染ssr

Vue源码规范

大圣训练营 专属福利



—仅限训练营期间—

奖学金班



上一节课返一节课学费，相当于
免费学（全国仅此一家）

普通班



训练营期间200定金抵扣500元

大圣专属福利（仅限前100名）



送价值299元小爱音响一个

专属福利口号



大圣真帅

名额有限

快快私聊你的课程顾问吧

姓名	手机号	课程名称	班次	班主任	已付金额	总课节数	每节返还金额	5月总课节数	符合要求课节数	上月累计零钱	5月返还总金额	5月实际返还金额	累计下月返还零钱	身份证号	银行卡账号	开户行 (详细到支行信息)	
周茜	18810185485	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	14052219	21	01 中国银行北
古婷	1355879	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	14058	21	02 工商银行丰
黄达	1732811	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	44098	99623	33 中国建设银
黄伟	1751980	web前端高级工程师	8	刘玲玲	12680	50		253.6	11	10	7.2	2543.2	2500	43.19999999999999	14260	91622	39 兴业银行惠
王凡	1801326	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	10	19.2	2615.2	2600	15.19999999999999	31010	22	12 招商银行上
李圣	1352369	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	9	19.2	2355.6	2300	55.59999999999999	13092519	21	02 中国工商银
李文	1851281	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	14232719	23	20 华夏银行北
李文	1823616	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	14112819	21	02 中国工商银
王伟	1836423	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	42112319	21	58 招商银行杭
王伟	1838914	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	51011219	23	02 四川天府银
孙予	1820	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	9	59.6	2396	2300		96 41132519	21485	招商银行北
周婧	15961596	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	10	19.2	2615.2	2600	15.19999999999999	23052319	21483	中国招商银
王心	1783402	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	41022319	21799	中国邮政储
许**	18703110	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	41302	21777	南京银行软
刘强	1851535	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	14272	89623573	中国银行上
王强	13213450	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	41148	91621226	中国工商银
张强	1862792	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	41142	21	招商银行上
张强	1355250	web前端高级工程师	8	刘玲玲	12980	50		259.6	11	11	19.2	2874.8	2800	74.80000000000000	42010	21	02 招商银行武
曹强	1851565	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181823023	87621	02 中国光大银		
沈强	1776240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181833062	89621	21 招商银行上		
强	1802240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181844018	88621	36 中国工商银		
强	1519240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181832128	92621	51 招商银行苏		
强	1316240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181843061	98621	30 建设银行广		
强	1589240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181841152	92621	11 中信银行郑		
强	1709240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181842112	94622	33 兴业银行深		
强	1804240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181823010	92621	45 招商银行哈		
强	18312437	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181814262	91621	02 工商银行亚		
强	1780226	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181814072	91621	01 招商银行北		
强	18082425	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181851132	92622	38 中国建设银		
强	1880240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181844023	91621	70 中国银行佛		
强	1565240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181835222	91621	18 中国建设银		
强	1510240	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181813053	91621	04 中国工商银		
强	13522743	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181841282	91621	01 招商银行北		
强	18500351	Web全栈架构师 (奖学金班)	18	王平	13980	55	254.18181818	10	10	8.36	2550.1781818	2500	50.1781818181814232	91623	57 杭州银行文		

Web全栈架构师 020期



上一节课 » 返一节学费！

郑重承诺: 官方赔付承诺+培优奖学金协议=双重保障返还无忧

奖学金返还标准
听课60% + 提交作业

只限100名



**WEB全栈架构师
教研团队**

Winter
前手机淘宝前端负责人
阿里 P8 高级技术专家, JavaScript 专家, 带领团队
开发了阿里巴巴开源项目 Weex 移动前端开发框架。

大圣老师
前百度和360前端架构师
9年前端开发经验, 精通Vue/React、源码架构、小程序、
移动端和Nodejs整个前端技术栈, 对前端萌新如何快速进
阶有丰富的经验, 业内少有既精通技术又精通授课的大咖。

杨老师
前高伟达高级软件工程师
9年前端开发经验, 精通Angular、Vue、React、Nodejs、
Express、Koa等主流全栈开发框架, 精通RN、Weex等
Hybrid App开发框架, 精通MySQL、MongoDB等, 对
模块化、组件化、自动化开发有深入实践和理解。

夏老师
前711电子商务平台前端负责人
8 年前端开发经验, 涉及的行业主要有电商、证券、金融, 曾参
与许多大型电商项目的开发和设计工作, 比如 711 电子商
务平台、野村证券交易平台瑞穗银行信用卡结算平台、Reach-
Max 广告数据分析平台等, 涉及前后端多种开发项目, 具有丰
富的全栈开发及架构设计经验。

韩老师
前阿里/淘宝资深前端开发
7 年前端开发经验, 任职开发期间, 对大型电商平台、办公
平台、性能优化、数据可视化、移动应用、SEO 等。有丰富
的实战经验, 比如阿里巴巴国际平台负责 SEO 业务对接和
性能优化, 手机淘宝 H5 技术开发, 双 11、双 12 技术支撑,
3.8 生活节前端负责人。精通 Vue、React、Node、小程序、
公众号等主流前端技术栈。

高老师
前京东资深前端开发
6 年前端开发经验, 曾就职京东、万科、用友, 精通Re-
act/Vue, 曾参与主导过很多大型移动端和PC端项目, 如
活动模板开发、金融支付、财务系统、房产管理系统等,
有着极其丰富的大型项目实战经验。





后厂理工学院

Kaikeba 开课吧

40

Kaikeba
开课吧



Kaikeba
开课吧





大圣训练营

专属福利



仅限训练营期间



奖学金班：

上一节课返一节课学费



普通班：

训练营期间200定金抵扣500



大圣学员专属福利：

送价值299元小爱音箱一个



专属福利口号：
我爱大圣

名额有限
私聊你的课程顾问，了解课程吧

深夜话题

- 35岁危机
- 学习 VS 加班
- 成长经历
- 如何描述做的项目
- 明天六点

谁帮我

可利用的资源

我是谁

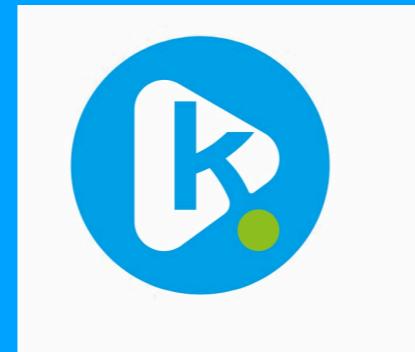
优势

缺少什么

加入这个公司我要付出什么

跳槽成本

给公司的价值



让雇主知道我

简历，影响力

让雇主相信我

项目，内推

目标公司想要
什么人

JD分析

加入这个公司我能的到什么

收入，自我价值



联想

快手

IBM

开课吧

滴滴

网易

新浪

百度

腾讯

百望山第一视角