



与「十」俱进,码出未来

2018源创会年终盛典

ElementUI 3.0 展望以及未来规划

饿了么大前端

邓芷阳

概述

1. ElementUI 开源两周年回顾
2. 组件库开发实践总结
3. ElementUI 3.0 要做出的改进
4. 不止于 3.0

1.ElementUI 开源两周年回顾

两年间取得的成就

- star 33k
- commit 3800+
- pr 3000+
- contributor 400+
- release 100+
- npm 下载量 3,700,000 +

2. 组件库开发实践总结

经验总结

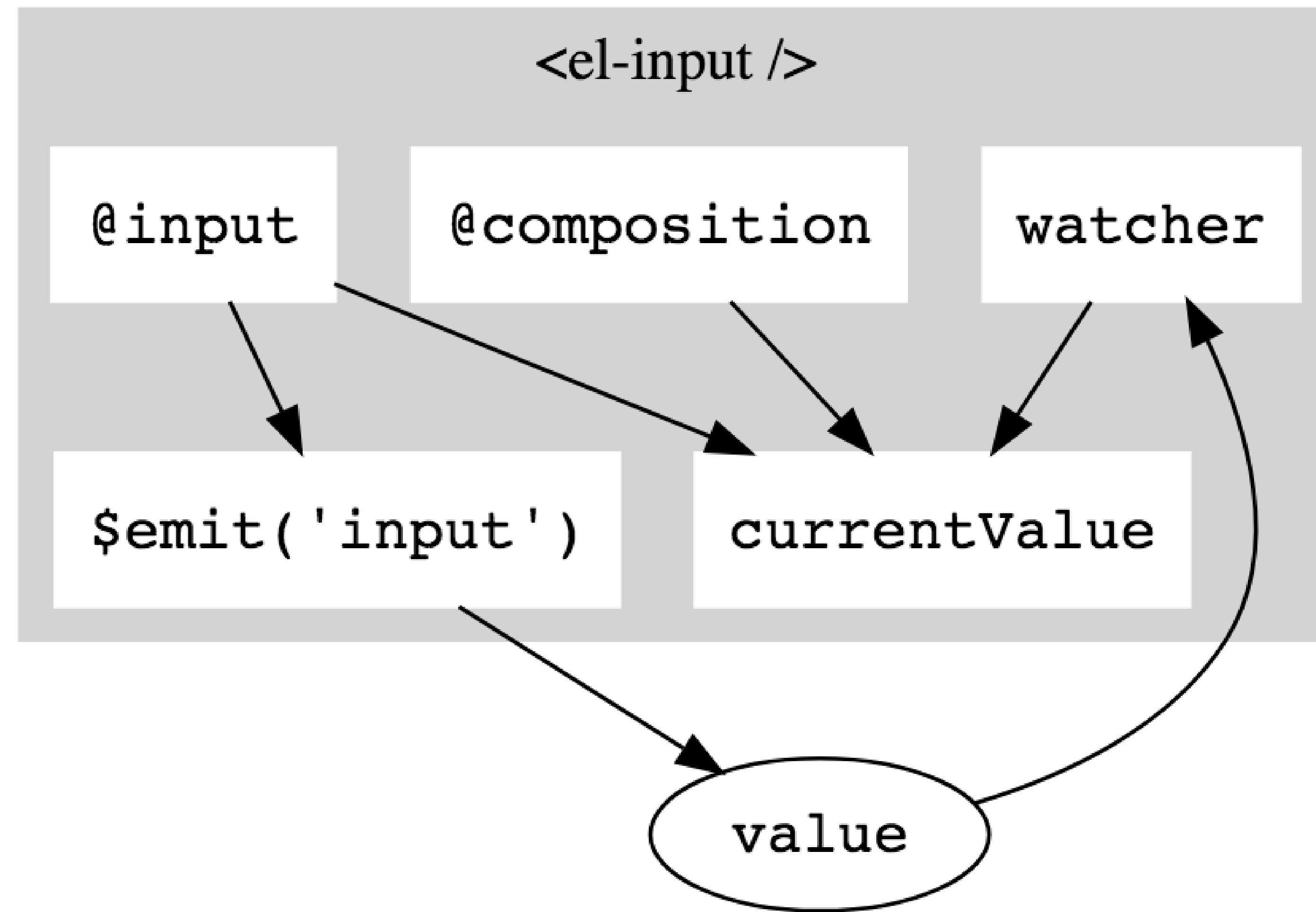
- Vue 新增 API 的影响
- 常见问题
- 一点经验之谈

Vue 2.0 以来新增 API

- provide 与 inject
- \$scopedSlots 与 scoped-slot
- .sync
- \$props, v-model ...

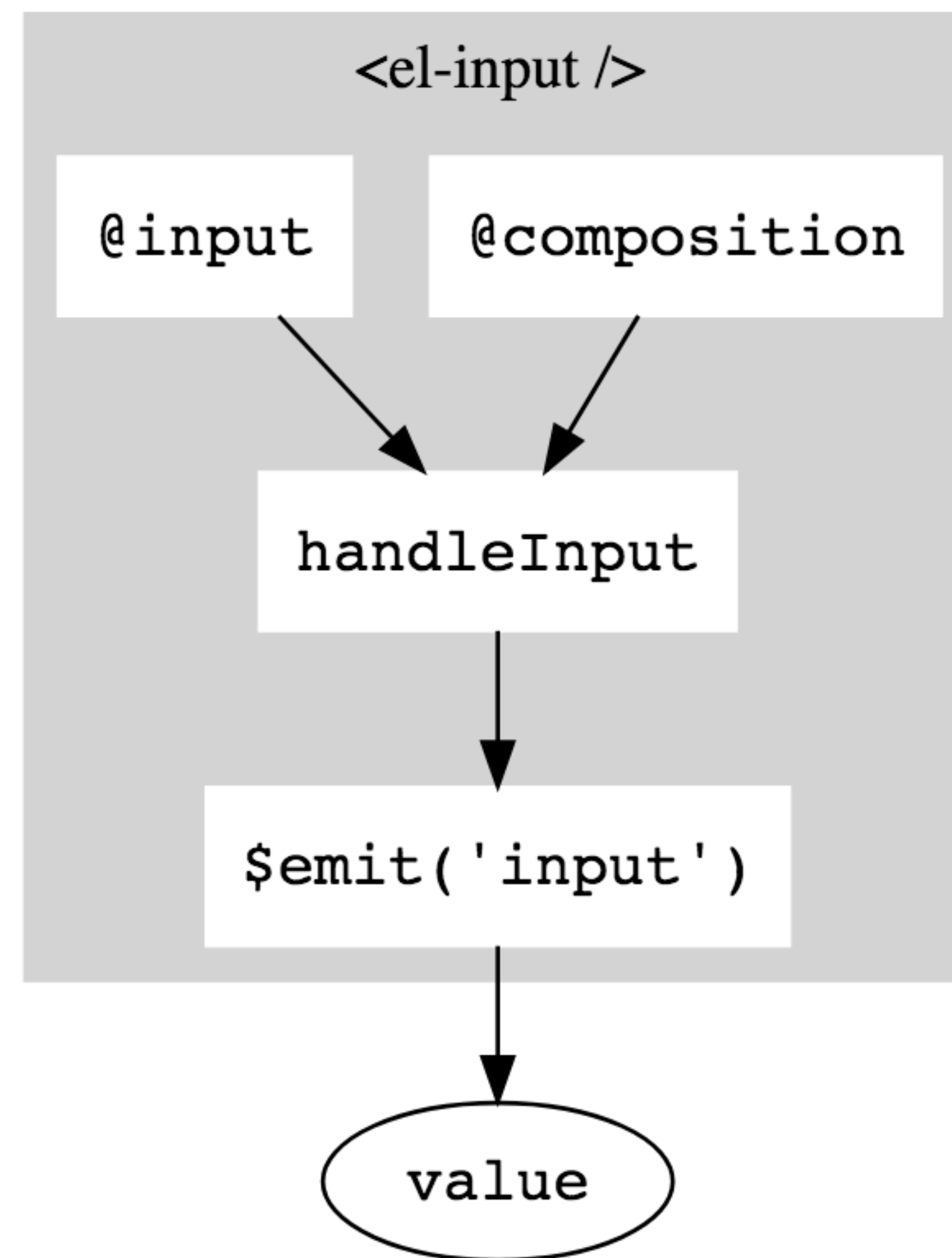
保持简洁

以 el-input 为例



[element-ui #13471](#)

- 去除不必要的属性
- watcher 有必要，但不能太复杂
- 向 React 学习



mixin 使用时需谨慎

- mixin 是一种对组件的复用
- 滥用让组件变得复杂
- [Mixins Considered Harmful](#)

```
const mixin = {  
  data() {  
    return {  
      foo: 'foo',  
      bar: 'bar'  
    }  
  }  
}  
  
export default {  
  mixins: [mixin],  
  data() {  
    return {  
      t: 't'  
    }  
  }  
}
```

替代方案

- 函数式组件
- Experimental Hooks API (Vue 3.0)

jsx 与 render

- 优先使用 `template`
- 符合 `Vue` 常见的用法: `slot-scope` 而非自定义的 `render` 函数
- `_c`, `_l`, `_v` 不应该出现



```
render() {  
  const description = <p>{ this.description }</p>  
  const list = this.arr.map((i, index) => <p key={index}>{i}</p>)  
  
  return (  
    <div>  
      { description }  
      <ul>{ list }</ul>  
    </div>  
  )  
}
```

?



```
render() {  
  const description = this._c('p', [this._v(this.description)])  
  const list = this._l(this.arr, (i, index) => <p key={index}>{i}</p>)  
  
  return (  
    <div>  
      { description }  
      <ul>{ list }</ul>  
    </div>  
  )  
}
```

?

key 一个特殊而实用的属性

*「发现组件出 **bug** 时，检查一下是不是设置组件的 **key**」*



```
<el-select v-model="value1" v-if="visible">
  <!-- ... -->
</el-select>
<el-select v-model="value2" multiple v-else>
  <!-- ... -->
</el-select>
```

Virtual DOM diff 不能满足所有场景

```

// src/core/vdom/path.js
function sameVnode (a, b) {
  return (
    a.key === b.key && (
      (
        a.tag === b.tag &&
        a.isComment === b.isComment &&
        isDef(a.data) === isDef(b.data) &&
        sameInputType(a, b)
      ) || (
        isTrue(a.isAsyncPlaceholder) &&
        a.asyncFactory === b.asyncFactory &&
        isUndef(b.asyncFactory.error)
      )
    )
  )
}

```

computed 也许会不生效

computed 仅仅对响应式数据有效



```
computed: {  
  visibleEl() {  
    return this.visible ? this.$refs.el1 : this.$refs.el2  
  }  
}
```

3. ElementUI 3.0 要做出的改进

代码层面

- 遵循「最佳实践」
- 开发新组件: Timeline, TreeSelect
- 改造组件: 统一用法, 推动组件重构

拆分&简化

- 把文档打包拆分: [doc-generator](#)
- 简化组件开发流程

```
build
├── bin
├── config.js
├── deploy-ci.sh
├── deploy-faas.sh
├── gen-single-config.js
├── git-release.sh
├── release.sh
├── salad.config.json
├── strip-tags.js
├── webpack.common.js
├── webpack.component.js
├── webpack.conf.js
├── webpack.demo.js
└── webpack.test.js
```

```
"scripts": {
  "bootstrap": "yarn || npm i",
  "build:file": "node build/bin/iconInit",
  "build:theme": "node build/bin/gen-css",
  "build:utils": "cross-env BABEL_ENV=ut",
  "build:umd": "node build/bin/build-loc",
  "clean": "rimraf lib && rimraf package",
  "deploy": "npm run deploy:build && gh-",
  "deploy:build": "npm run build:file &&",
  "dev": "npm run bootstrap && npm run b",
  "dev:play": "npm run build:file && cro",
  "dist": "npm run clean && npm run buil",
  "i18n": "node build/bin/i18n.js",
  "lint": "eslint src/**/*.js test/**/*.js pac",
  "pub": "npm run bootstrap && sh build/",
  "test": "npm run lint && npm run build",
  "test:watch": "npm run build:theme &&",
},
```

不仅仅是代码

- 让订制主题更加便捷
- 整理设计资源
- UI 库 → 设计系统

可访问性 (a11y)

- [Making Your Vue App Accessible](#)
- 抽象出 aria utils
- 避免直接对 DOM 的操作

```
mounted() {  
  let innerInput = this.$refs.input.$refs.input  
  innerInput.setAttribute('role', 'spinbutton')  
  innerInput.setAttribute('aria-valuemax', this.max)  
  innerInput.setAttribute('aria-valuemin', this.min)  
  innerInput.setAttribute('aria-valuenow', this.currentValue)  
  innerInput.setAttribute('aria-disabled', this.inputNumberDisabled)  
}
```

社区运营

如何吸引更多的人参与开源？

- 订制规范
- 开发文档
- 社区指南

4. 不止于 3.0

Vue 3.0 带来的改变

- [Vue 3.0 进展分享](#)
- 期待新的 API: hook, 响应式数据监听
- 代码精简, 至少 \$set 不会再使用了
- Typescript 值得一试

联系我们

- <https://github.com/elemeefe/element>
- <http://github.com/element-experimental/>
- fe.job@ele.me

Thanks!