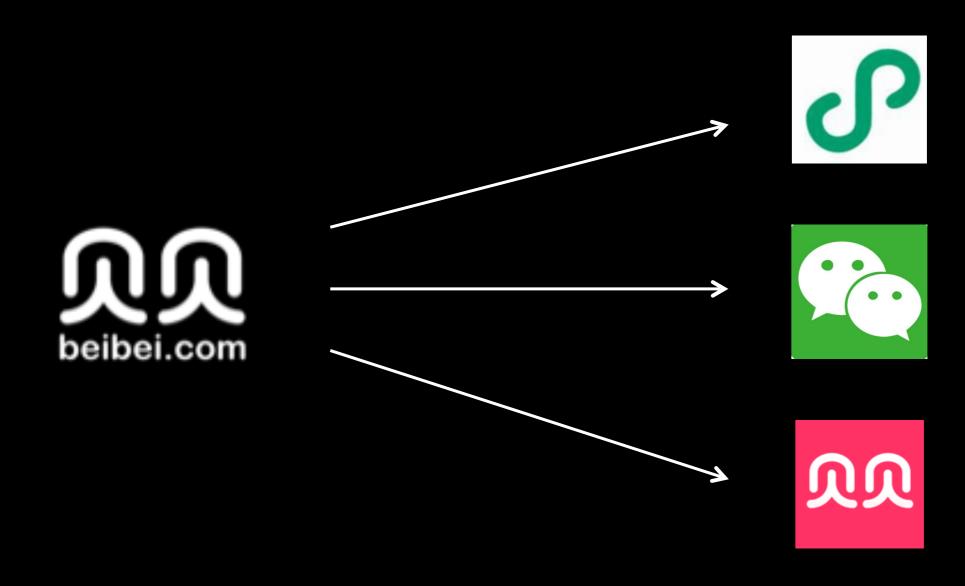
## 基于AST的跨端方案探索与实践

贝贝集团-何文俊

## 现状

贝贝是一家母婴电商公司,产品以手机为主要呈现载体。在手机APP、浏览器、微信端为用户提供个性化服务。



#### 现状

为了在小程序、微信、APP中更好地展现产品,同时考虑到开发成本、用户体验, 我们选择了VUE、安卓/IOS原生实现、WEEX、微信小程序作为开发技术。

## 技术概况 - WEB

**WEB** 

原生实现

**WEEX** 

小程序

简介:

基于VUE的H5技术栈。

开发效率:

\*\*\*\*

用户体验:



适用场景:

APP、微信、微信小程序



## 技术概况 - 原生

**WEB** 

原生实现

**WEEX** 

小程序

简介:

基于安卓/IOS原生技术实现业务场景。

开发效率:



用户体验:

\*\*\*\*

适用场景: APP



## 技术概况 - Weex

**WEB** 

原生实现

**WEEX** 

小程序

#### 简介:

Weex 是一个可以使用现代化的 Web 技术开发高性能原生应用的框架。

#### 开发效率:



用户体验:

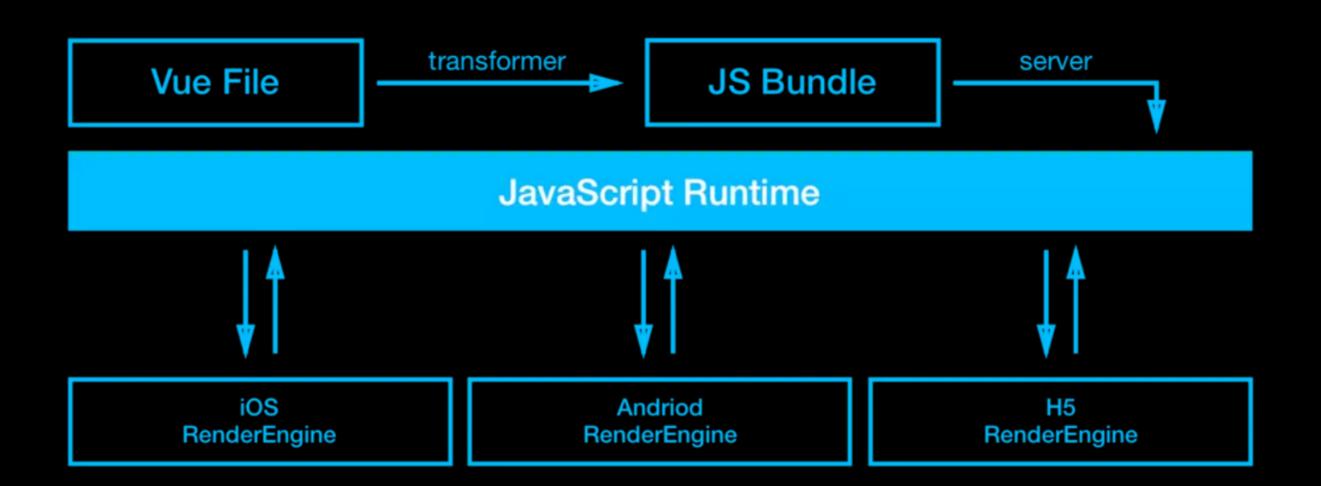


适用场景:

APP、微信

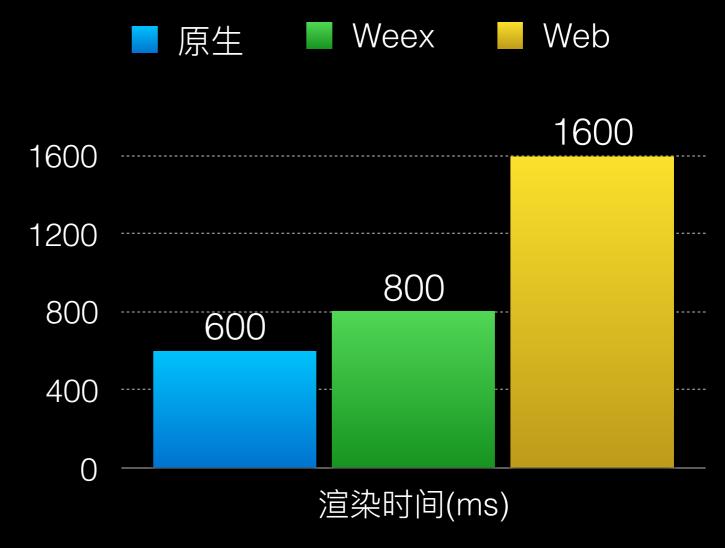


## 技术概况 - Weex



#### 技术概况 - Weex





## 技术概况 - 小程序

**WEB** 

原生实现

**WEEX** 

小程序

简介:

微信小程序。

开发效率:

\*\*\*

用户体验:

\*\*\*

适用场景:

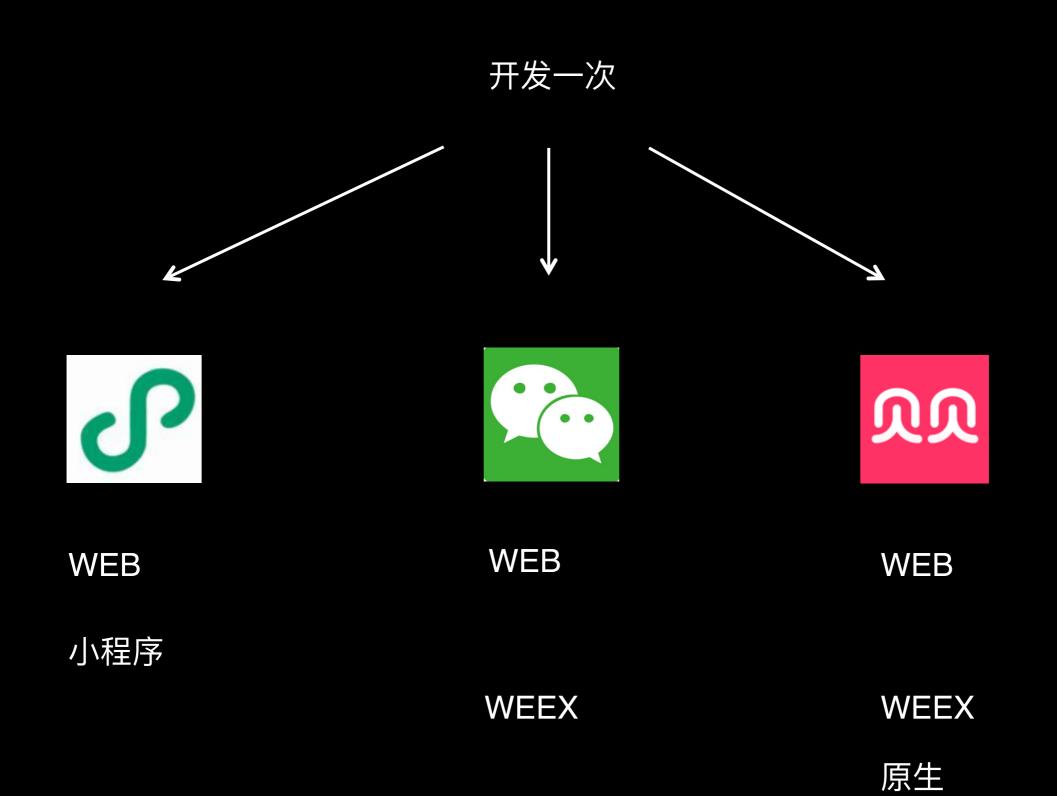
微信小程序



# 技术对比

|      | 开发效率 | 用户体验 | 使用场景       |
|------|------|------|------------|
| WEEX | **   | ***  | APP、微信     |
| WEB  | **** | **   | APP、微信、小程序 |
| 小程序  | ***  | ***  | 小程序        |
| 原生   | **   | **** | APP        |

# 我们的需求





18K React

H5

各种小程序



2K VUE

H5

各种小程序





## 业界方案

chameleon megola

uni-pp taro

需求:

基于VUE

支持小程序

支持WEEX

VUE子集



VUE多态

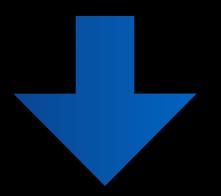
H5

各种小程序

WEEX

#### 贝贝的跨端方案

使用VUE开发



开发效率: ★★ ★★

用户体验: \*\*\*\*

适用场景:ALL

小程序

- 基础库支持

- H5般的开发体验

**WEB** 

- 基础库支持

**WEEX** 

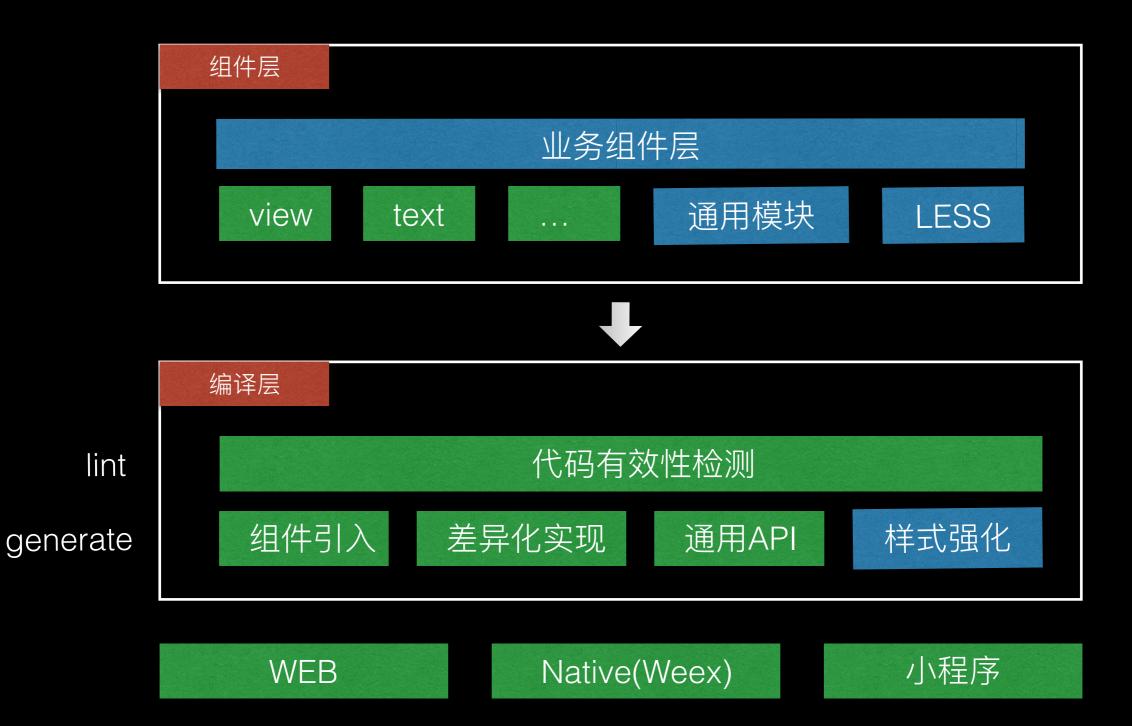
- 基础库支持

- 客户端深度优化

- 调试工具优化

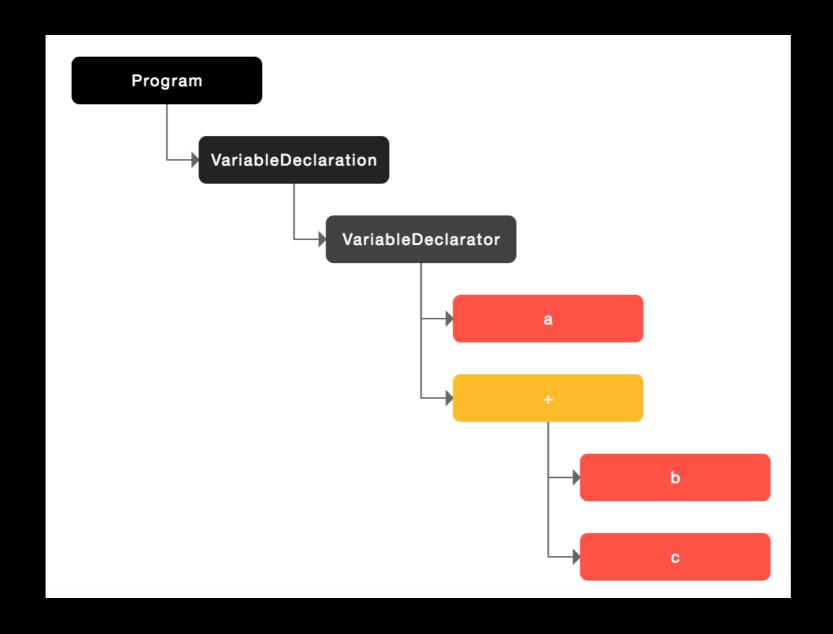
- H5般的开发体验

## 跨端方案架构

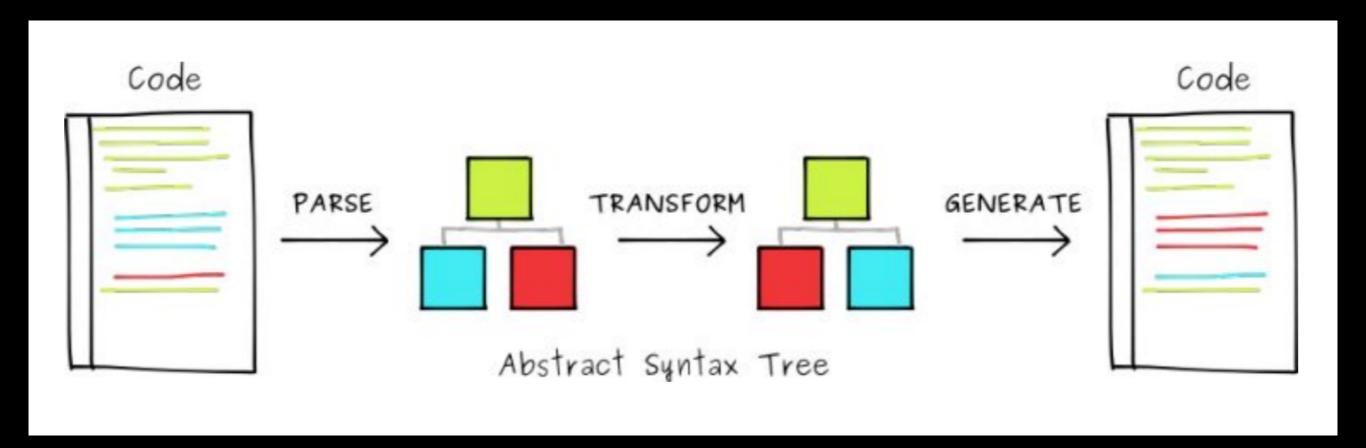


# 抽象语法树(AST)

var a = b + c;



## AST经典操作流程



#### 操作示意

#### 去除注释

```
// 推啊开发日地址
const address = '杭州数娱大厦';
                                            const address = '杭州数娱大厦';
               parse
                                                        generate
   [{
      type: 'VariableDeclaration',
                                             [{
      id: 'address',
                                transform
                                                type: 'VariableDeclaration',
      value: '杭州数娱大厦'
                                                id: 'address',
  }, {
                                                value: '杭州数娱大厦'
      type: 'CommentLine',
      value: '推啊开发日地址'
  }]
```

#### 抽象语法树(AST)

#### 通过定义一系列Node,精准描述一段代码

#### **Declarations**

- FunctionDeclaration
- VariableDeclaration
  - VariableDeclarator

#### **Expressions**

- ThisExpression
- ArrayExpression
- ObjectExpression
  - Property
- FunctionExpression
- Unary operations
  - UnaryExpression
    - UnaryOperator
  - UpdateExpression
    - UpdateOperator



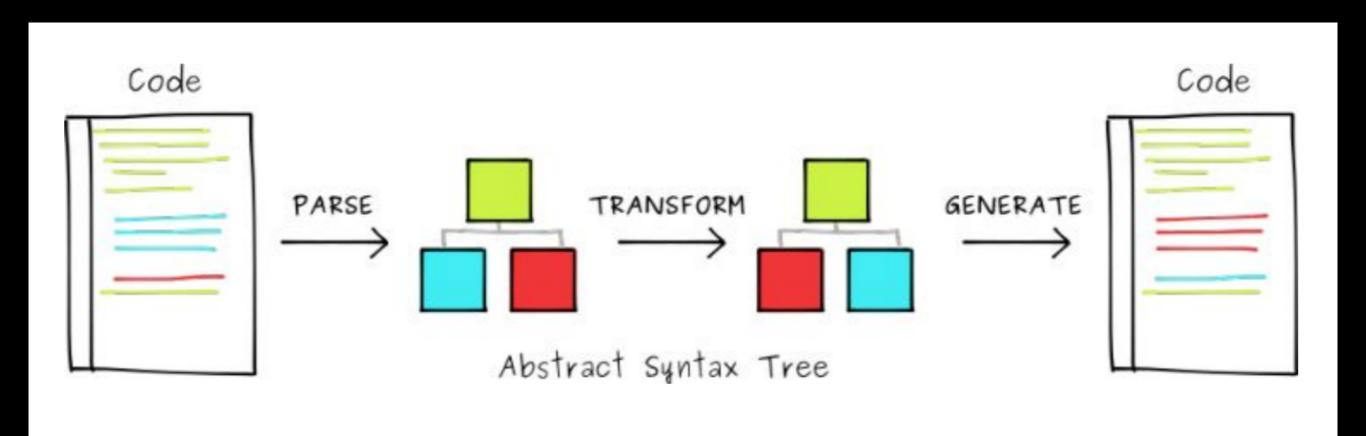




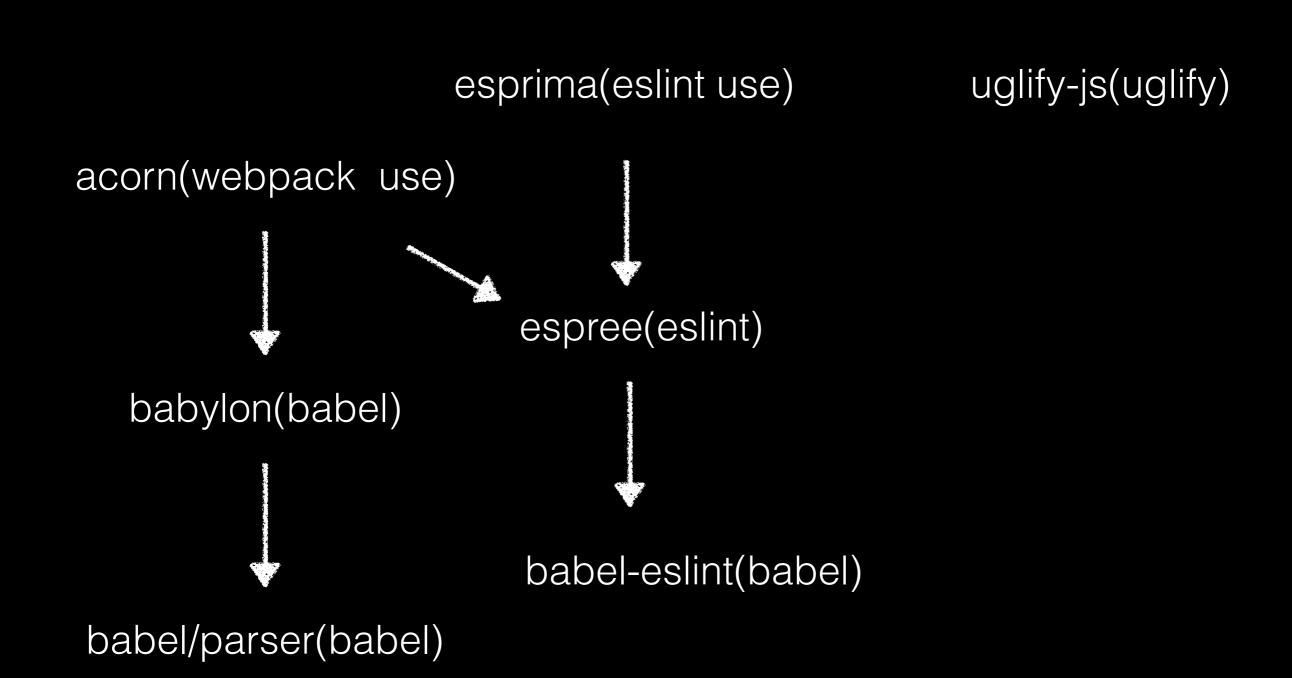
parse

```
* 根据输入的 unix 时间戳, 格式化为 201
  注意 unix 时间戳的单位是秒, 而用于 j
 * @param {Number} unixTimestamp
 * @returns 格式化后的时间: 2017-08-29
getGmtTime(unixTimestamp) {
    let time = new Date(unixTimestamp >
    let year = time.getFullYear();
    let month = time.getMonth() + 1;
    let date = time.getDate();
    let hour = time.getHours();
    let minute = time.getMinutes();
   month = month >= 10 ? month : `0${n}
   date = date >= 10 ? date : `0${date
   hour = hour >= 10 ? hour : `0${hour
   minute = minute >= 10 ? minute : `@
    return `${year}-${month}-${date} ${
},
```

#### 水一章



## AST引擎



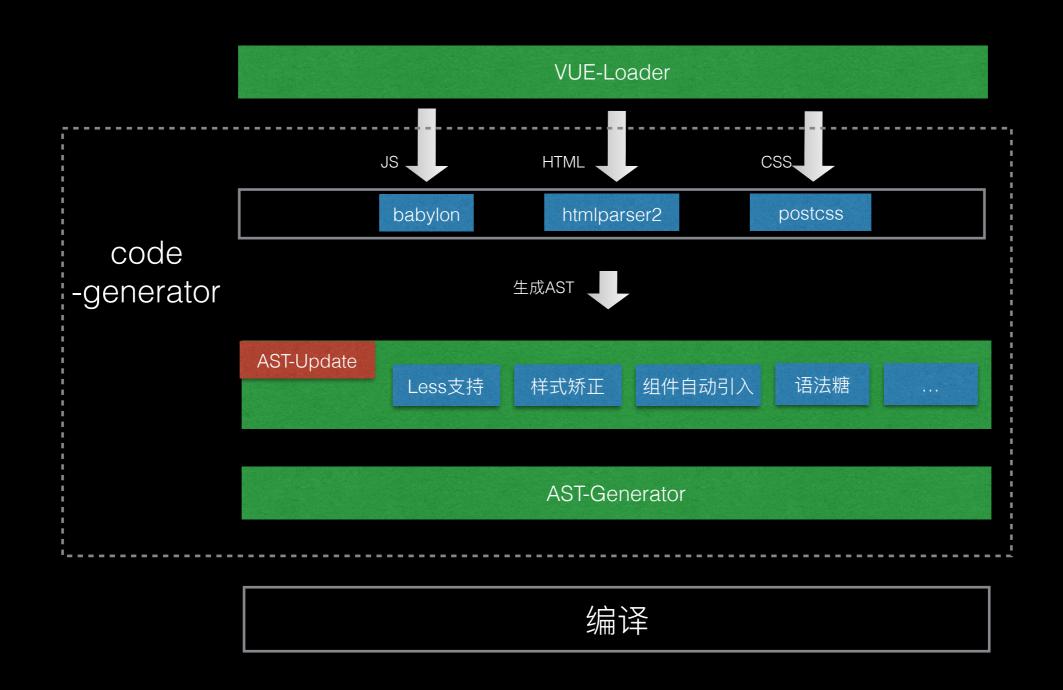
#### 无处不在的AST

### 他们都能被AST 这么多...



## 跨端项目中的代码转换

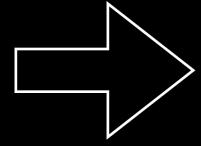
跨端项目中的代码转换模块示意



#### 样式扩展

WEB

background: red;



WEEX

background-color: red;

generate

```
parse
```

```
decl = $node {
    + raws: {before, between
        type: "decl"
    + source: {start, end}
        prop: "background"
        value: "red"
```

transform

```
decl = $node {
    + raws: {before, between}

        type: "decl"

    + source: {start, end}

        prop: "background-color"

        value: "red"
```

跨端的路上总有一些不如意, 某场景要求在APP和微信端都要投放。



 一
 商品
 十
 登录

登录模块在APP内外实现差异很大。

方案1:

微信登录

+

APP登录

通用登录

方案2:

if APP

=>

APP登录

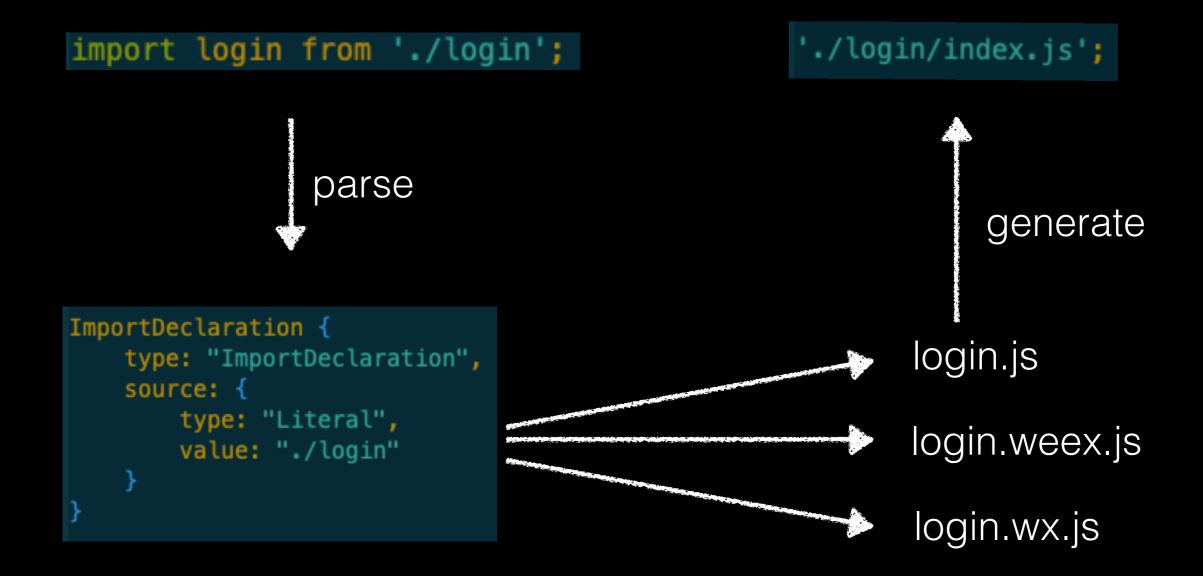
else if 微信 =:

微信登录

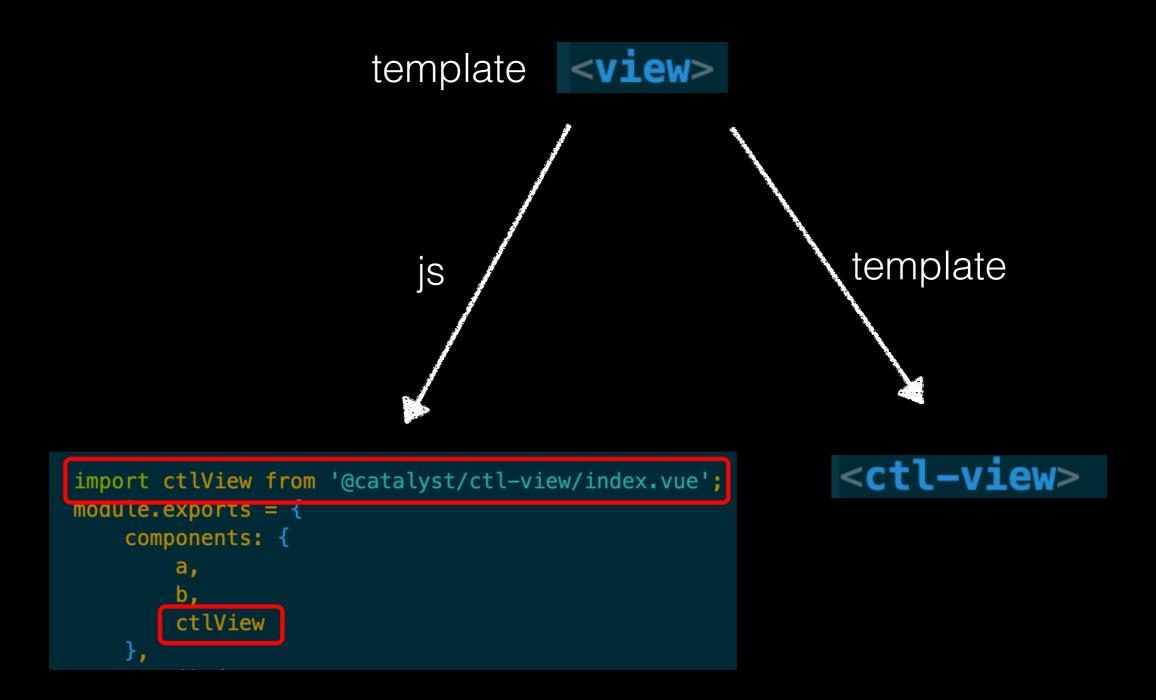
为各端独立实现一份,且编译时按需打包。



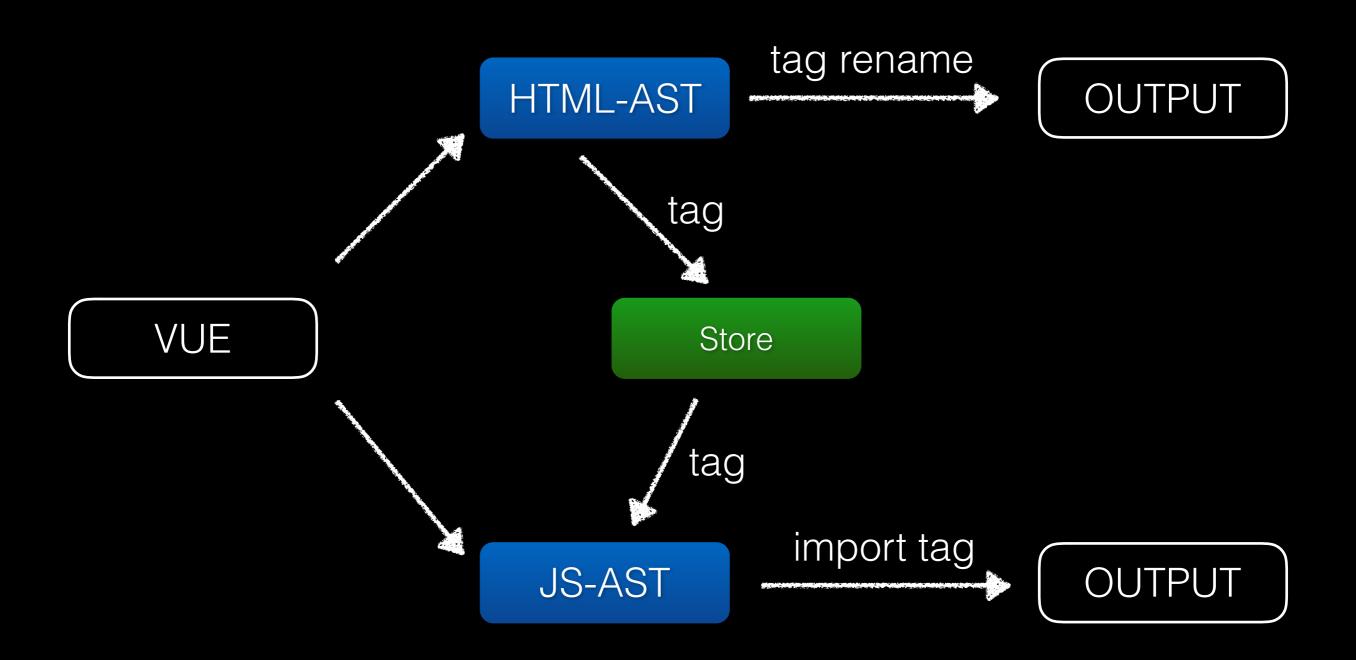
```
目录结构
                       import login from './login';
 login/
   index.js
   index.weex.js
                                 weex
   index.wx.js
                              './login/index.weex.js';
                   小程序
                                 找不到x.web.js
            ./login/index.wx.js';
                                     ./login/index.js';
```



## 组件自动引入

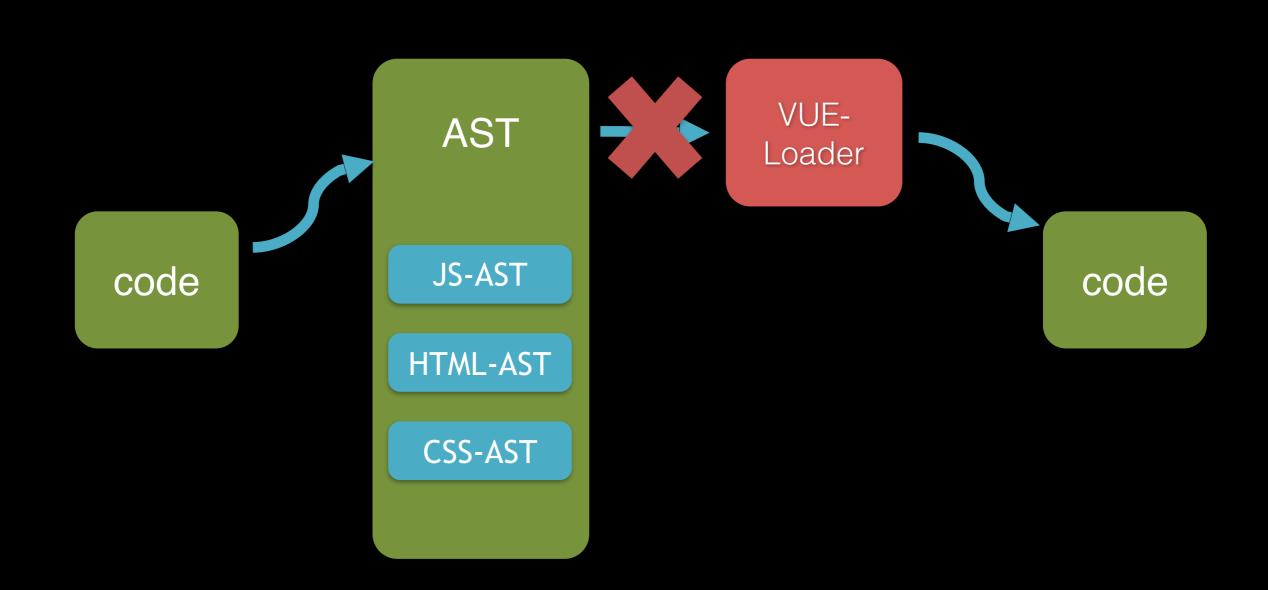


# 组件自动引入

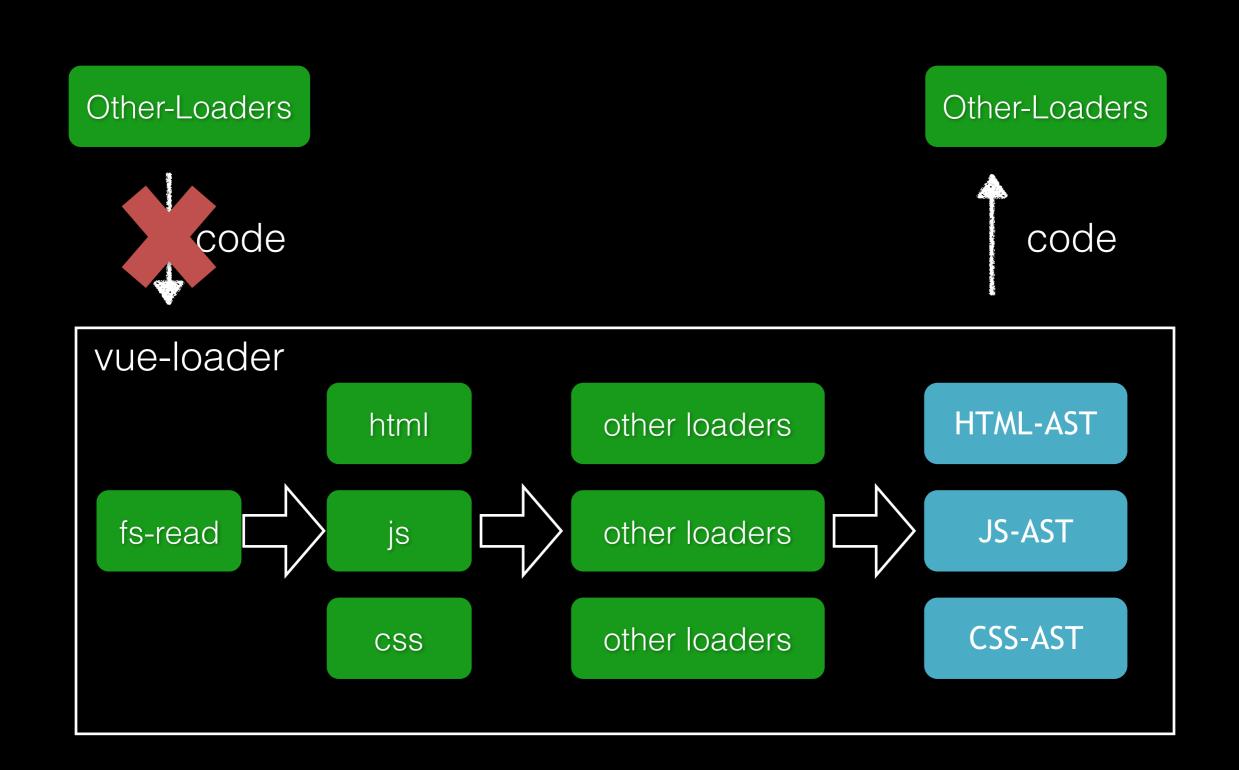


## 编译流程

webpack

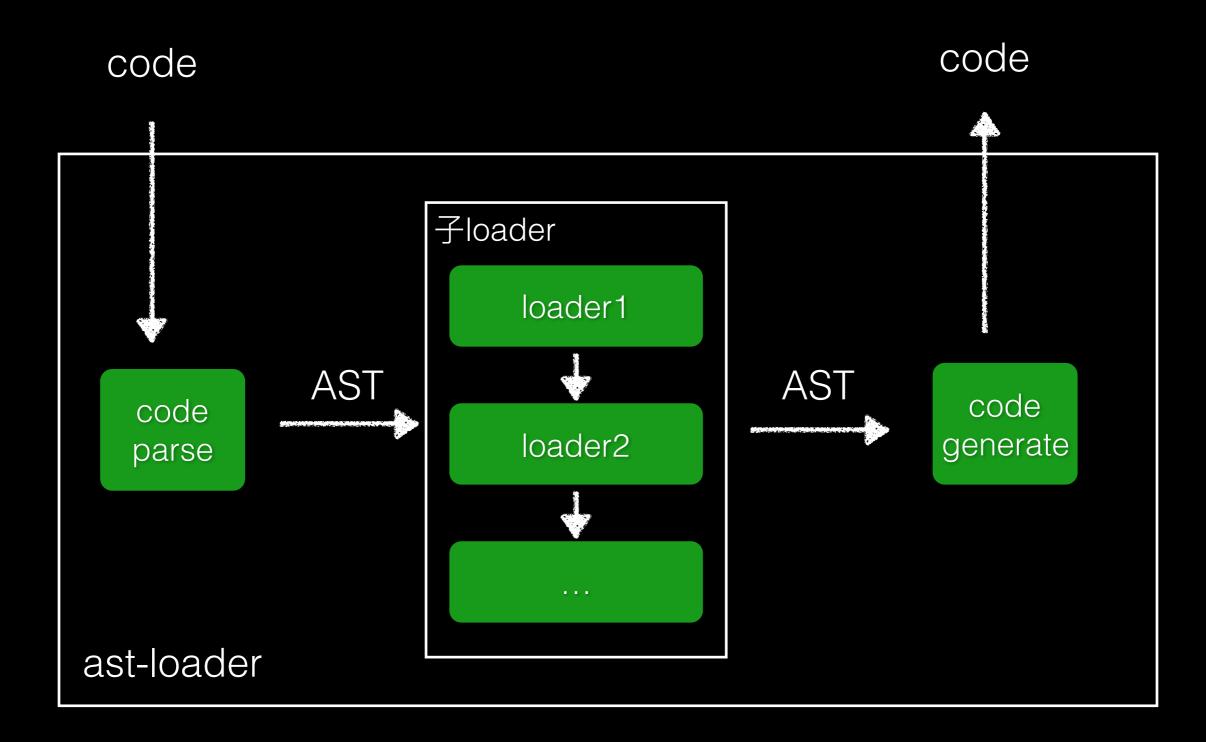


## 特别的loader

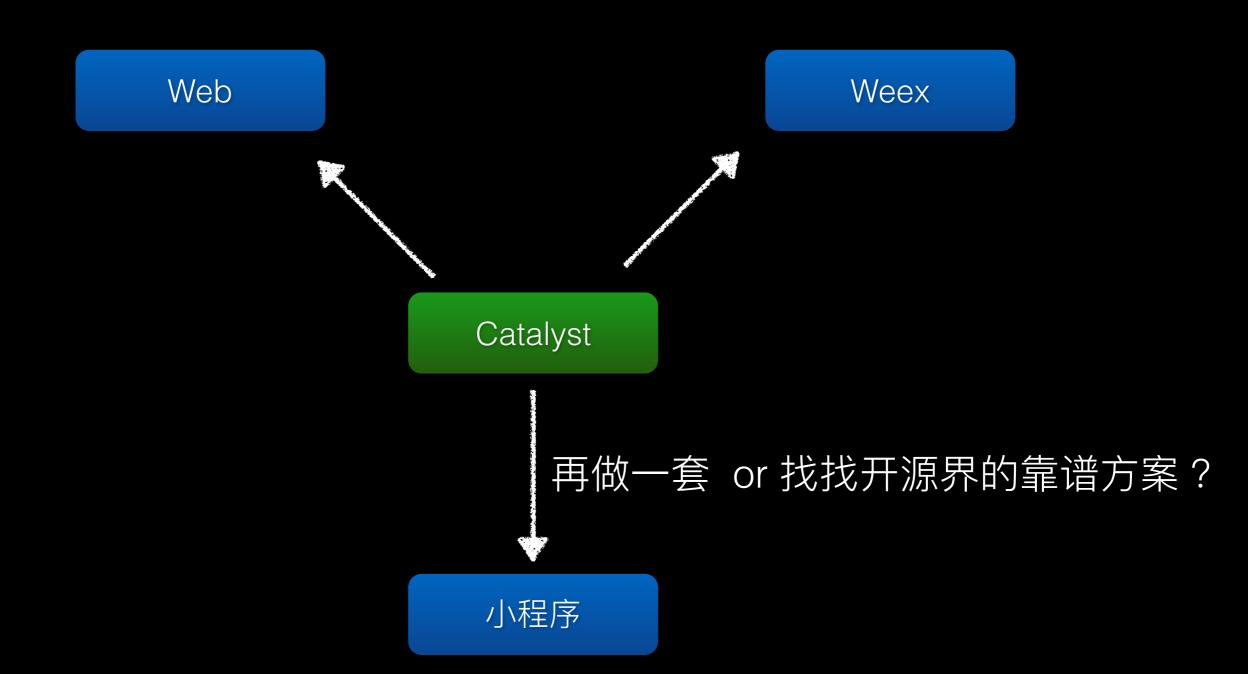


## AST-loader设计

AST解析只做一次,支持插槽式扩展

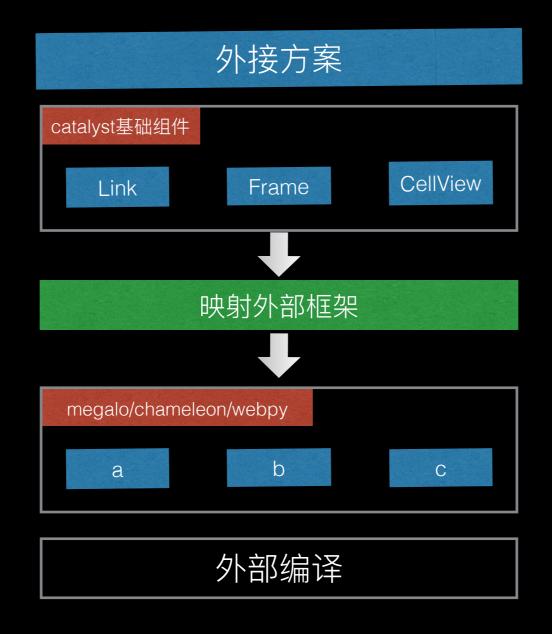


## 扩展新平台



#### Vue => 小程序

vue转化为小程序这块,目前业界已经有较多成熟方案。 所以我们选择桥接的方式,适配业界方案。



#### 扩展-我们的工具



```
Elements Console
                            Sources Network
<WXRootView frame="{{0, 88}, {375, 724}}" hidden="N0" alpha="1" opaque=</pre>
                                                                              Styles Computed Event Listeners >>>
"YES">
                                                                              Filter
                                                                                                             :hov .cls +
 v<div frame="{{0, 0}, {375, 724}}" hidden="N0" alpha="1" opaque="YES">
  ▼ <div frame="{{0, 0}, {375, 240}}" hidden="N0" alpha="1" opaque="YES">
                                                                              text {
                                                                                                              via inspector
     <div frame="{{0, 160}, {80, 80}}" hidden="N0" alpha="1" opaque="YES"</pre>
                                                                                width: 375;
     backgroundColor="blue"></div>
                                                                                height: 19;
    ▼ <div frame="{{0, 60}, {375, 100}}" hidden="N0" alpha="1" opaque="YES"
                                                                                top: 0;
    backgroundColor="green">
                                                                                left: 0;
       <text frame="{{0, 0}, {375, 19}}" hidden="N0" alpha="1" opaque="N0"
                                                                                hidden: NO;
                                                                                alpha: 1;
       value="1"></text> == $0
                                                                                opaque: NO;
                                                                                value: 1;
     <div frame="{{0, 0}, {90, 60}}" hidden="N0" alpha="1" opaque="YES"</pre>
     backgroundColor="red"></div>
    </div>
                                                                                         position
  </div>
 </WXRootView>
                                                                                           margin
                                                                                            border -
                                                                                              padding
                                                                                                 375 \times 19
```

## AST能做哪些事

代码压缩

错误追踪

代码混淆

代码覆盖率检查

浏览器兼容

Linting

条件编译

重构

依赖追踪

Q & A