---

link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=20 sentences=21, words=162

---

1. PostCSS

- PostCSS (https://www.postcss.com.cn/)是一个用 JavaScript 工具和插件转换 CSS 代码的工具
- 增强代码的可读性
- 将未来的 CSS 特性带到今天！
- 终结全局 CSS
- 避免 CSS 代码中的错误
- 强大的网格系统
- postcss会帮我们分析出css的抽象语法树

2. 文档

- api (http://api.postcss.org)
- astexplorer (https://astexplorer.net/#/2uBU1BLuJ1)

3. 类型

- CSS AST主要有3种父类型
    - `AtRule` @xxx的这种类型，如@screen
    - `Comment` 注释
    - `Rule` 普通的css规则
- 子类型
    - `decl` 指的是每条具体的css规则
    - `rule` 作用于某个选择器上的css规则集合

4.AST节点

- nodes: CSS规则的节点信息集合
    - decl: 每条css规则的节点信息
    - prop: 样式名,如width
    - value: 样式值,如10px
- type: 类型
- source: 包括start和end的位置信息，start和end里都有line和column表示行和列
- selector: type为rule时的选择器
- name: type为atRule时@紧接rule名，譬如@import 'xxx.css'中的import
- params: type为atRule时@紧接rule名后的值，譬如@import 'xxx.css'中的xxx.css
- text: type为comment时的注释内容

5.操作方法

5.1 遍历

- walk 遍历所有节点信息，无论是atRule、rule、comment的父类型，还是rule、 decl的子类型
- walkAtRules：遍历所有的AtRules
- walkComments 遍历所有的Comments
- walkDecls 遍历所有的Decls
- walkRules 遍历所有的Rules

```
root.walkDecls(decl => {
    decl.prop = decl.prop.split('').reverse().join('');
});
```

5.2 处理

- postCss给出了很多操作css规则的方法
- api (http://api.postcss.org/AtRule.html)
- 处理css的方式其实有2种：编写postcss plugin，如果你的操作非常简单也可以直接利用 `postcss.parse`方法拿到 `css ast`后分析处理

5.3 postcss plugin

- postcss插件如同babel插件一样，有固定的格式
- 注册个插件名，并获取插件配置参数 `opts`
- 返回值是个函数，这个函数主体是你的处理逻辑，有2个参数，一个是root,AST的根节点。另一个是result，返回结果对象，譬如 `result.css`，获得处理结果的css字符串

```
export default postcss.plugin('postcss-plugin-name', function (opts) {
  opts = opts || {};
  return function (root, result) {

  };
});
```

5.4 直接调用postcss命名空间下的方法

- 可以用postcss.parse来处理一段css文本，拿到css ast，然后进行处理，再通过调用toResult().css拿到处理后的css输出