



与「十」俱进,码出未来

2018源创会年终盛典

02

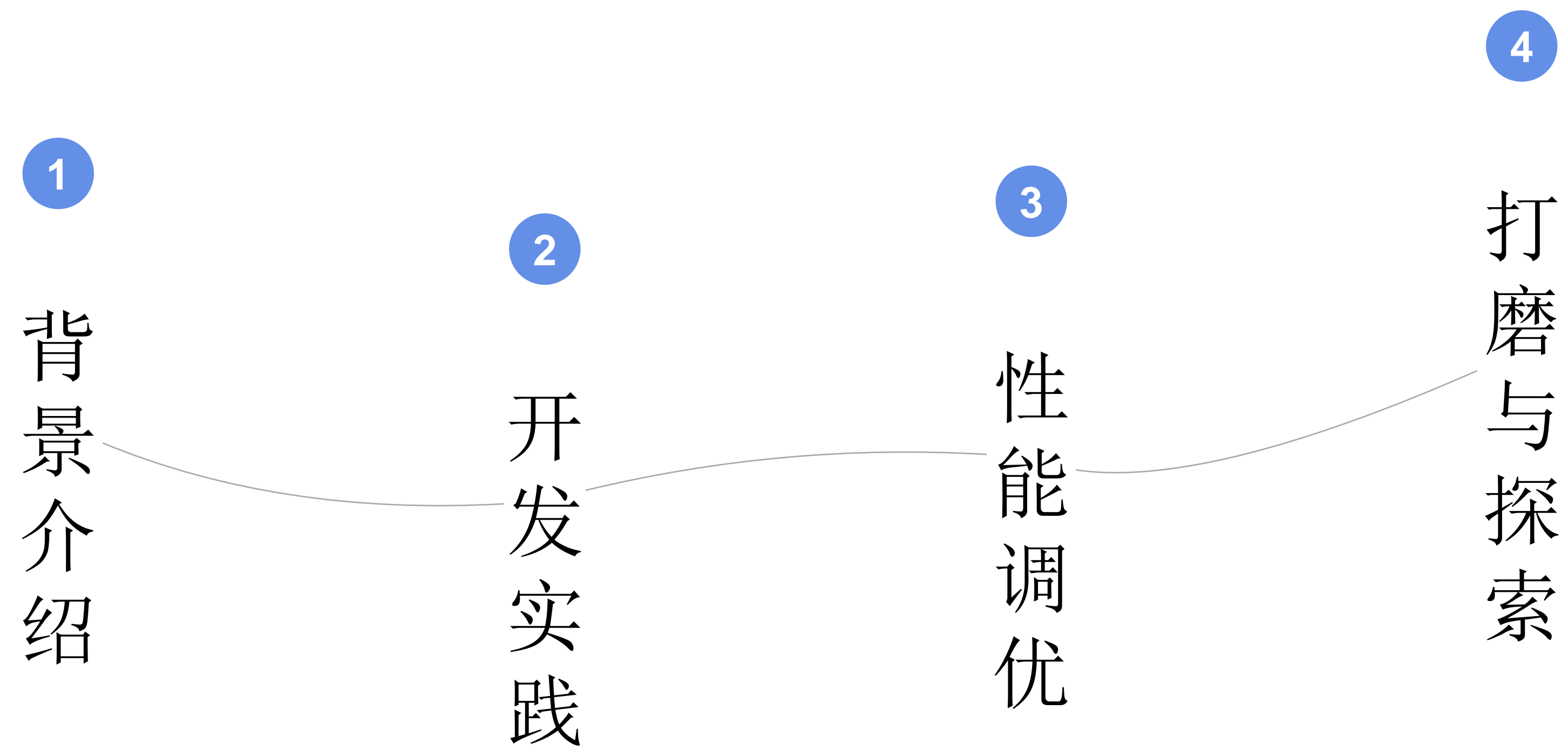
OPEN ORIENTED

凹凸实验室

基于 Taro 的多端项目实践

陈嘉健

aotu.io



1

背景介绍



背景介绍

02



荐一物

杂志类导购小程序



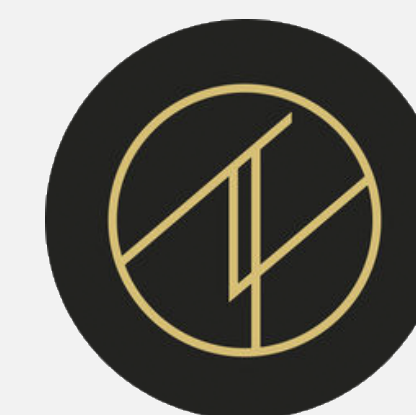
壹赫兹

短视频导购小程序



京东购物

京东购物主体小程序



TOPLIFE

国际奢侈品电商平台



7FRESH

生鲜食品超市

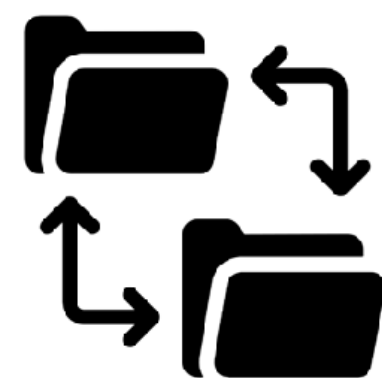
2 0 1 8 年 终 盛 典 · 深 圳



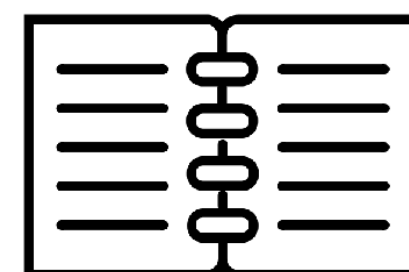
背景介绍

小程序开发痛点

02



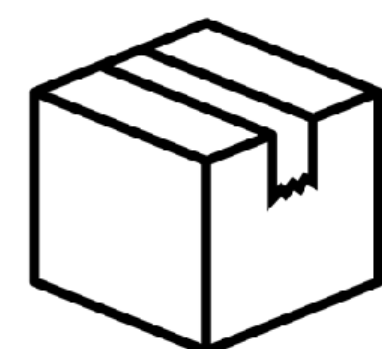
代码组织复杂



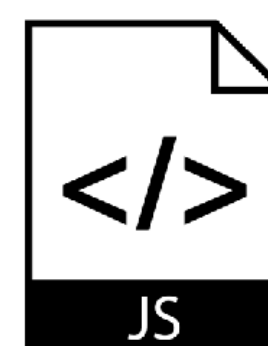
规范不统一



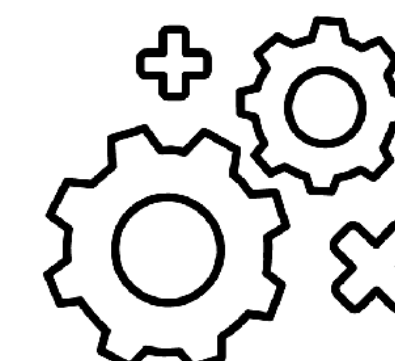
孱弱的字符串模板



依赖管理混乱



不能使用 ES Next

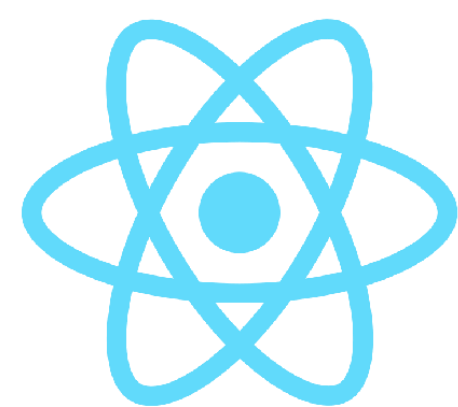


落后的开发方式

背景介绍

多端需求

02



...

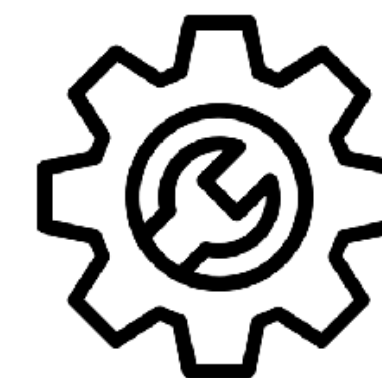
背景介绍

多端开发痛点

02



研发成本增加



代码维护困难

Which?



WePY



mpvue

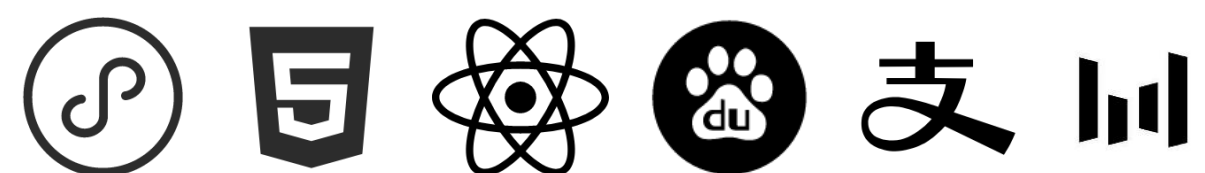


Started on
2018.6



Taro

一套遵循 React 语法规范的
多端统一开发框架



issues

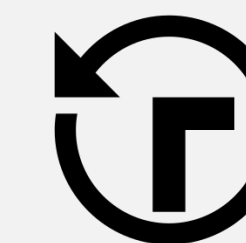
233 Open

1026 Closed



Stars

12308



commits

3213

背景介绍

Taro 特性

02



多端运行

一键生成可以在多端运行的代码



统一语法规范

采用 React 语法标准，支持 JSX 书写



开发体验

代码智能提示，实时代码检查



现代化开发流程

配套 Taro CLI 开发工具

背景介绍

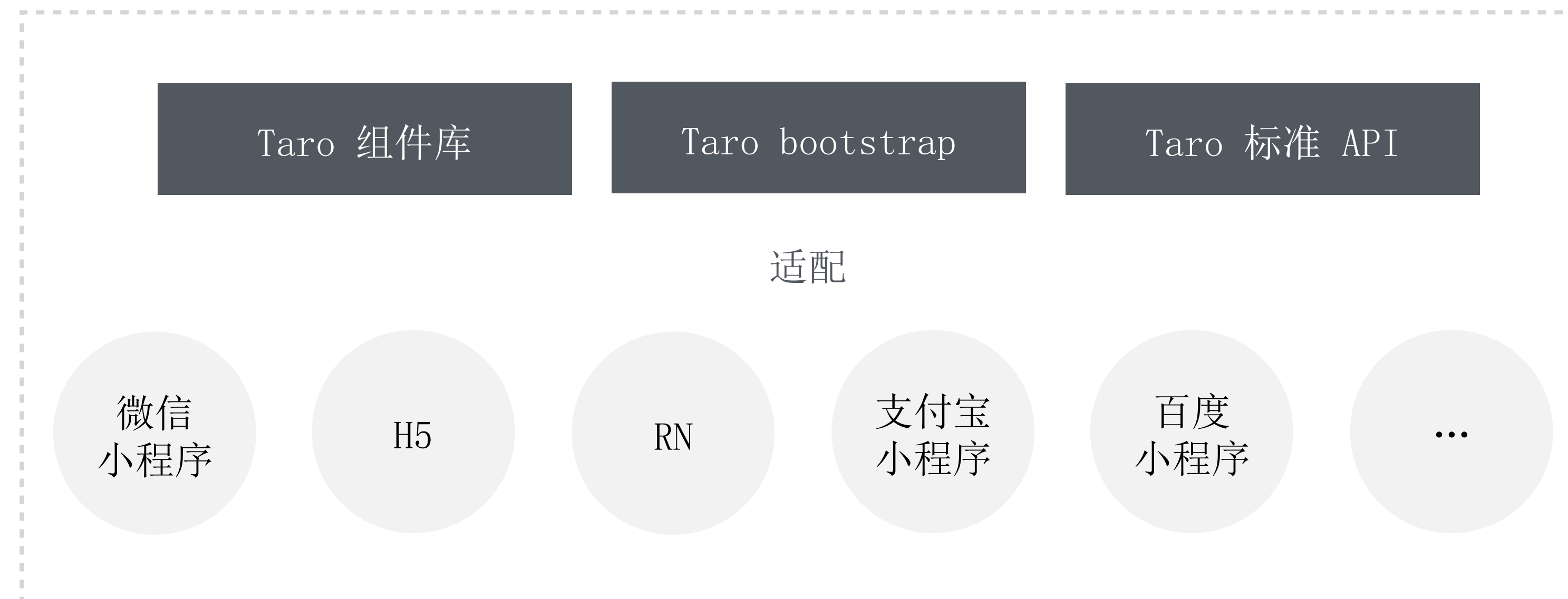
Taro 架构

02

编译时



运行时



2

开发实践



开发实践 · 基础篇

- 快速开始
- 路由配置
- Taro 语法


```
$ npm install -g @tarojs/cli  
$ taro init myApp  
$ taro build --type weapp --watch
```

```
class App extends Component {  
  config = {  
    pages: [  
      'pages/index/index', // 首页  
      'pages/detail/detail', // 详情页  
      'pages/my/my', // 个人中心  
    ]  
  }  
}
```

开发实践

Taro 代码

```
import Taro, { Component } from '@tarojs/taro'
import { View, Text } from '@tarojs/components'
import './index.scss'

export default class Index extends Component {
  config = {
    navigationBarTitleText: '首页'
  }

  // onLoad
  componentWillMount () { }

  // onReady
  componentDidMount () { }

  // onUnload
  componentWillUnmount () { }

  // onShow
  componentDidShow () { }

  // onHide
  componentDidHide () { }

  render () {
    return (
      <View className='index'>
        <Text>1</Text>
      </View>
    )
  }
}
```



```
// js
import defaultExportName from './utils/name.js'

// css
import './css/name.scss'

// image, video, font ...
import pic from './images/pic.png'
// 使用
<Image src={pic} />
```

小程序组件

Taro 组件

attached

componentWillMount

ready

componentDidMount

detached

componentWillUnmount

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

componentDidUpdate

```
import Taro from '@tarojs/taro'

Taro.getStorage({ key: 'key' })
  .then(res => console.log(res.data))
  .catch(err => console.log(err))
```


开发实践 · 进阶篇

- 如何使用 JSX 描述小程序模板
- Taro 的组件化方案

编译后的组件

Taro 组件

```
class Child extends Component {
  state = {
    str: 'Hello world!'
  }

  render() {
    return <View>{this.state.str.replace('!', '!!!!')}</View>
  }
}
```



```
class Child extends Component {
  state = {
    str: 'Hello world!'
  }

  _createData () {
    this.__state = arguments[0] || this.state || {}
    this.__props = arguments[1] || this.props || {}

    var anonymousState__temp = this.__state.str.replace('!', '!!!!')

    Object.assign(this.__state, { anonymousState__temp })
    return this.__state
  }
}
```

Taro 组件

```
class Child extends Component {
  state = {
    str: 'Hello world!'
  }

  render() {
    return <View>{this.state.str.replace('!', '!!!!')}</View>
  }
}
```



wxml

```
<block>
  <view>{{anonymousState__temp}}</view>
</block>
```

- state 的属性和 props 的属性不能重名
- 不渲染的变量不要放在 state 中

运行时

Taro 组件



小程序组件

计算小程序 data

生命周期适配

事件适配

.....

```
Component({  
  
  this.props.property  
  
  const { property } = this.props  
  
  const property = this.props.property  
  
})
```

微信小程序自定义组件间事件机制

```
<!-- 监听事件 -->  
<component-tag-name bindmyevent="onMyEvent" />  
<!-- 或者可以写成 -->  
<component-tag-name bind:myevent="onMyEvent" />
```

```
// 触发事件  
this.triggerEvent('myevent', myEventDetail, myEventOption)
```

父组件 JSX

```
<Child onTrigger={this.handleEvent}></Child>
```

编译

父组件 wxml

```
<child bindontrigger="handleEvent"></child>
```

1

```
export default class Child extends Component {  
  handler () {  
    this.props.onTrigger('hello')  
  }  
  
  render () {  
    return <View onClick={this.handler.bind(this)} />  
  }  
}
```

—— 编译 ——>

```
handler () {  
  this.__triggerPropsFn(  
    'onTrigger',  
    [null].concat(['hello'])  
  )  
}
```


2

```
export default class Child extends Component {  
  render () {  
    return <View onClick={this.props.onTrigger.bind(this, 'hello')} />  
  }  
}
```

—— 编译 ——>

```
function funPrivategdbqe() {  
  this.__triggerPropsFn(  
    "onTrigger",  
    [].concat(Array.prototype.slice.call(arguments))  
  );  
}
```

- props 中传递的函数属性的名字，必须以 'on' 开头
- 小程序端不要尝试打印父组件传入的函数

```
<View>{this.props.children}</View>
```

— 编译 —>

```
<view>  
  <slot></slot>  
</view>
```

父组件 render

```
render() {  
  return (  
    <Child  
      renderClose={  
        <Button>Close</Button>  
      }  
    >  
      <View>Hello world!</View>  
    </Child>  
  )  
}
```

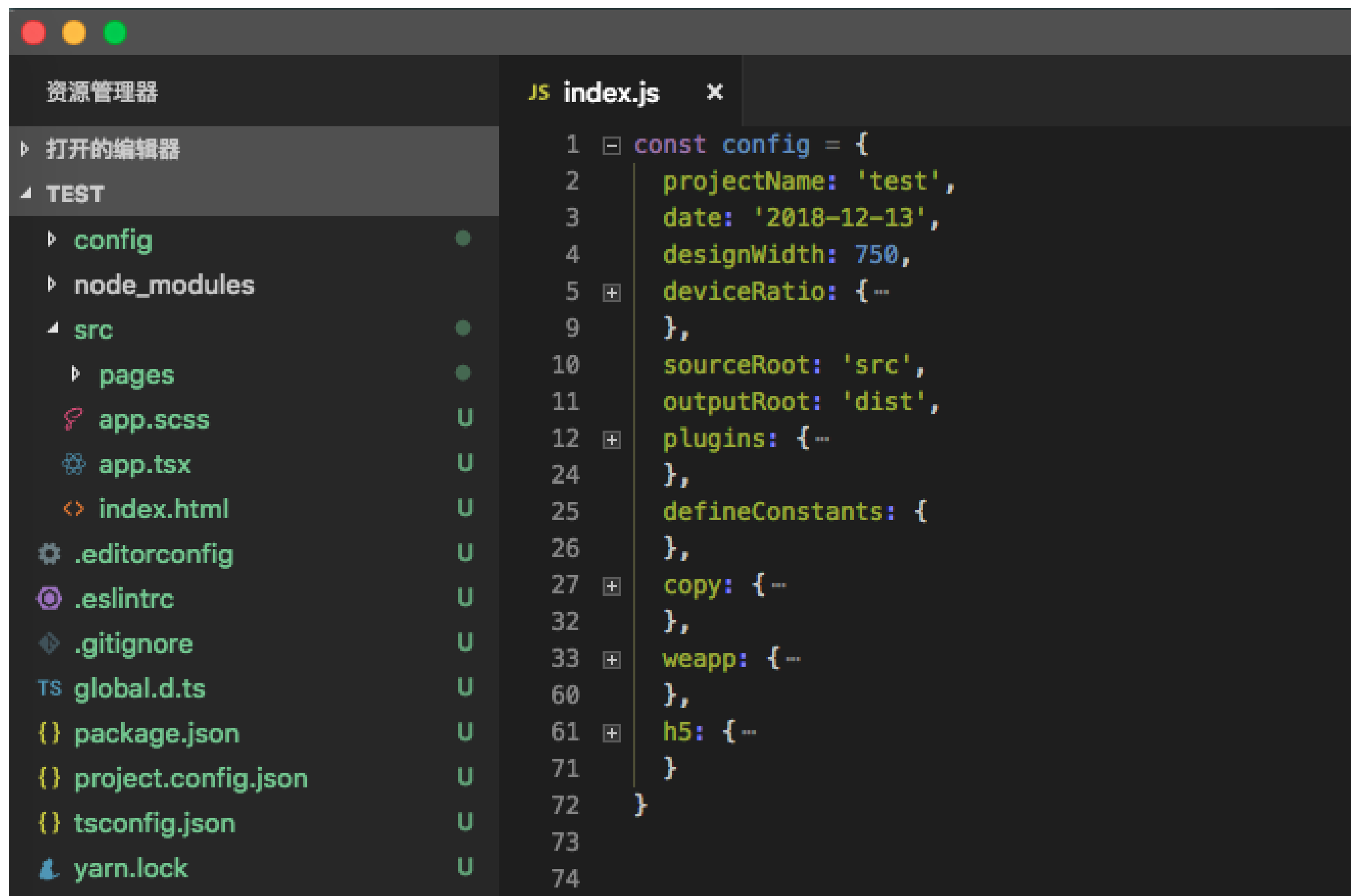
子组件 render

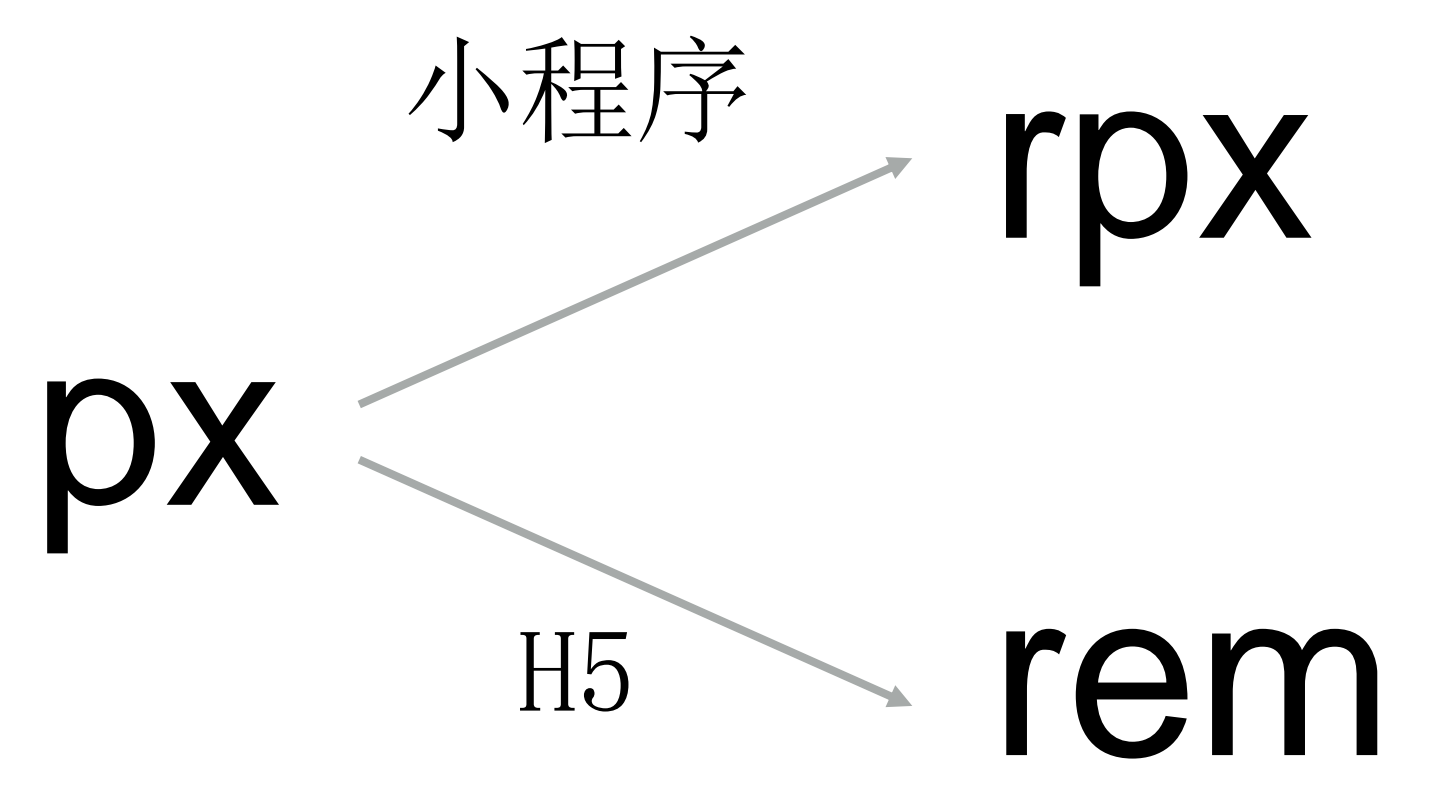
```
render() {  
  return (  
    <View>  
      <View>  
        {this.props.children}  
      </View>  
      <View>  
        {this.props.renderClose}  
      </View>  
    </View>  
  )  
}
```

- 不要对 Children 和各组合属性进行任何操作
- Children 和组合不能使用 defaultProps 设置默认内容
- 微信小程序不支持在循环中使用 Children 或组合

开发实践 · 工程篇

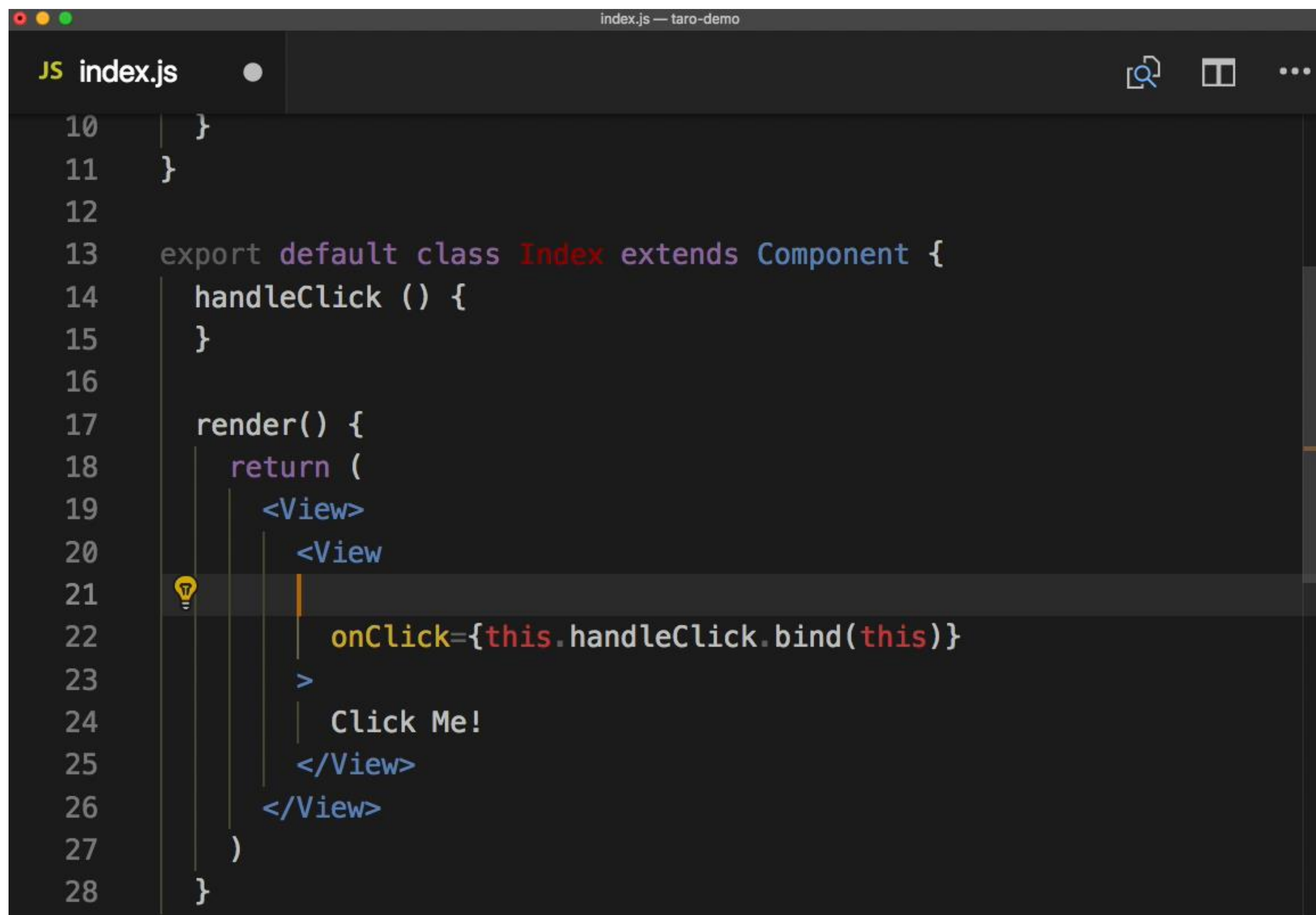
- 项目配置
- 智能提示与语法检查
- 经验总结
- Taro 提供的帮助





换算基准
config.designWidth





```
10   }
11   }
12
13   export default class Index extends Component {
14     handleClick () {
15     }
16
17     render() {
18       return (
19         <View>
20           <View
21             ⚡
22             onClick={this.handleClick.bind(this)}
23           >
24             Click Me!
25           </View>
26         </View>
27       )
28     }
```



```
index.js — taro-demo
JS index.js
27
28   render() {
29     return (
30       <View>
31         {this.state.list.map((item, index) => {
32           return <View key={index} />
33         })}
34
35       <View>Click Me!</View>
36     </View>
37   )
38 }
39 }
40
```

```
export default class Base extends Component {  
  $goto (url, params, options) {  
    return goto(url, params, options)  
  }  
  
  getParams (params) {  
    return getParams(params)  
  }  
  
  getImg (url) {  
    return getImg(url)  
  }  
}
```

process.env.TARO_ENV

weapp

/

h5

/

rn

/

alipay

/

swan

/

tt

根据平台引用依赖

```
if (process.env.TARO_ENV === 'weapp') {  
  require('path/to/weapp/name')  
} else if (process.env.TARO_ENV === 'h5') {  
  require('path/to/h5/name')  
}
```


根据平台渲染组件

```
{process.env.TARO_ENV === 'weapp' ? <View>weapp</View> : <View>others</View>}
```

页面：
配置路由

组件：
配置调用方的 `usingComponents` 字段

```
export default class Test extends Component {
  config = {
    // 定义需要引入的第三方组件
    usingComponents: {
      'ec-canvas': '../components/ec-canvas/ec-canvas' // 书写第三方组件的相对路径
    }
  }

  render () {
    return (
      <View>
        <ec-canvas id='mychart-dom-area' canvas-id='mychart-area'></ec-canvas>
      </View>
    )
  }
}
```

local

```
import Taro, { Events } from '@tarojs/taro'

const events = new Events()

// 监听
events.on

// 触发
events.trigger

// 取消监听
events.off
```

global

```
import Taro from '@tarojs/taro'

Taro.eventCenter.on
Taro.eventCenter.trigger
Taro.eventCenter.off
```

开发实践

状态管理

02



CSS MODULES

3

性能调优



视图层

WebView
渲染载体

evaluateJavascript

evaluateJavascript

逻辑层

JSCore/V8
运行环境

性能调优

常见的 setData 操作错误

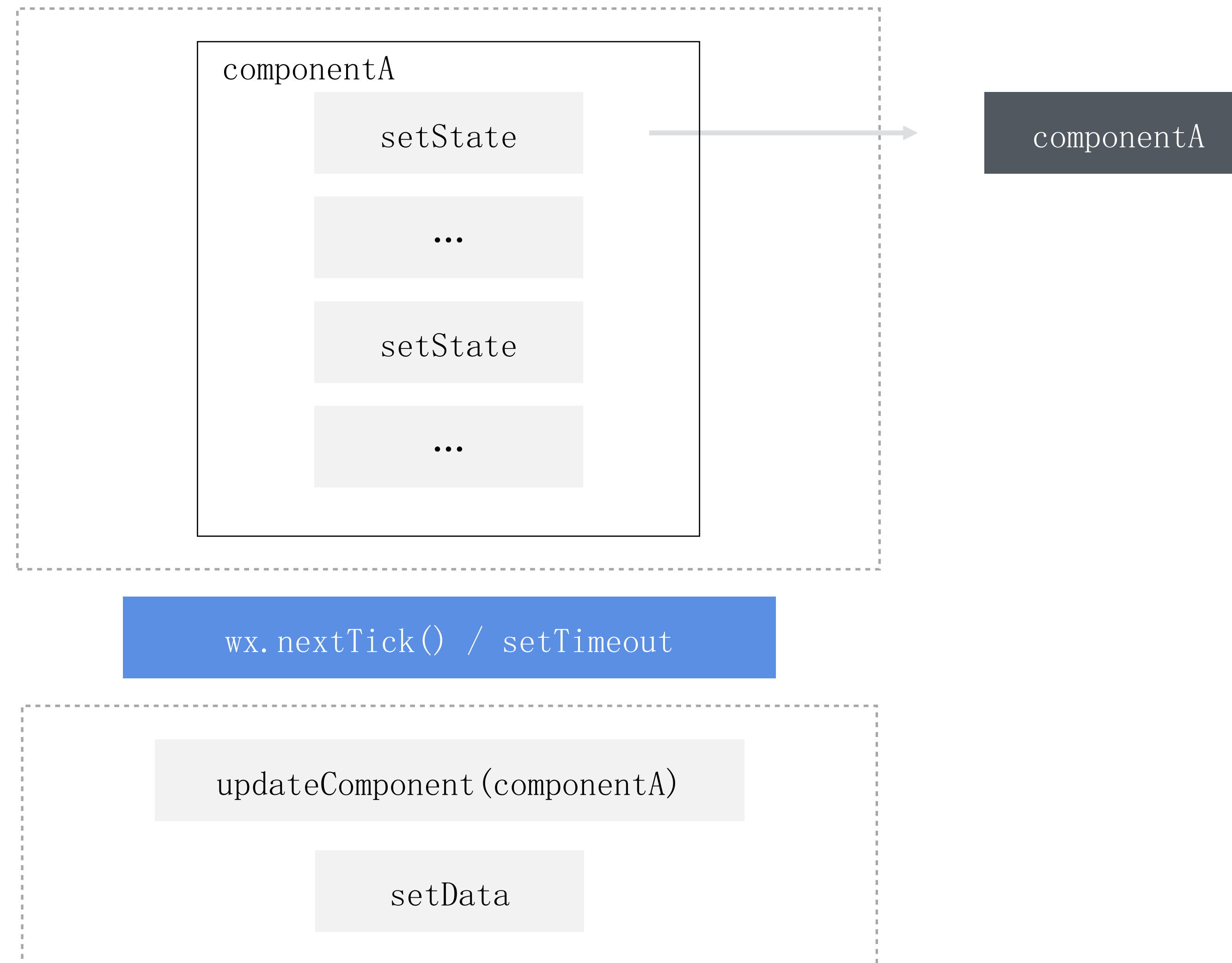
02

- 频繁地去 setData
- 单次 setData 传递大量的数据

性能调优

合并 setData

02



```
// 原始数据  
this.state = {  
  a: [1, 1],  
  b: {  
    x: 1,  
    y: 2  
  }  
}
```

```
// setState  
this.setState({  
  a: [1, 2, 3],  
  b: {  
    x: 10,  
    y: 2  
  }  
})
```

数据
diff

```
this.setData({  
  'a[2]': 2,  
  'a[3]': 3,  
  'b.x': 10  
})
```


性能调优

避免不必要的渲染

02

```
shouldComponentUpdate (nextProps, nextState) {  
  return (  
    !shallowCompare(this.props, nextProps) ||  
    !shallowCompare(this.state, nextState)  
  )  
}
```

性能调优

避免不必要的渲染

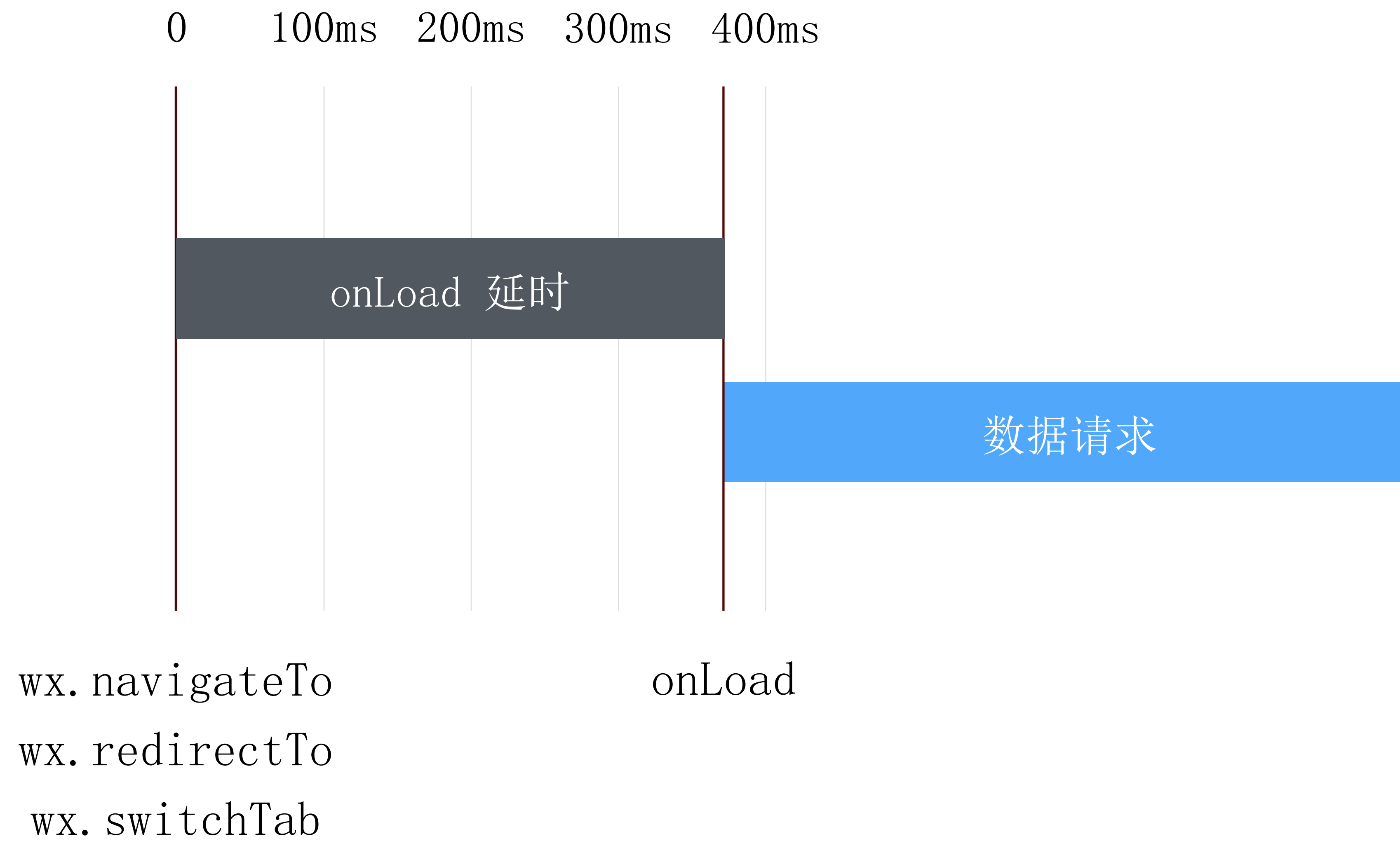
02

```
class Index extends Taro.PureComponent {  
  ...  
}
```

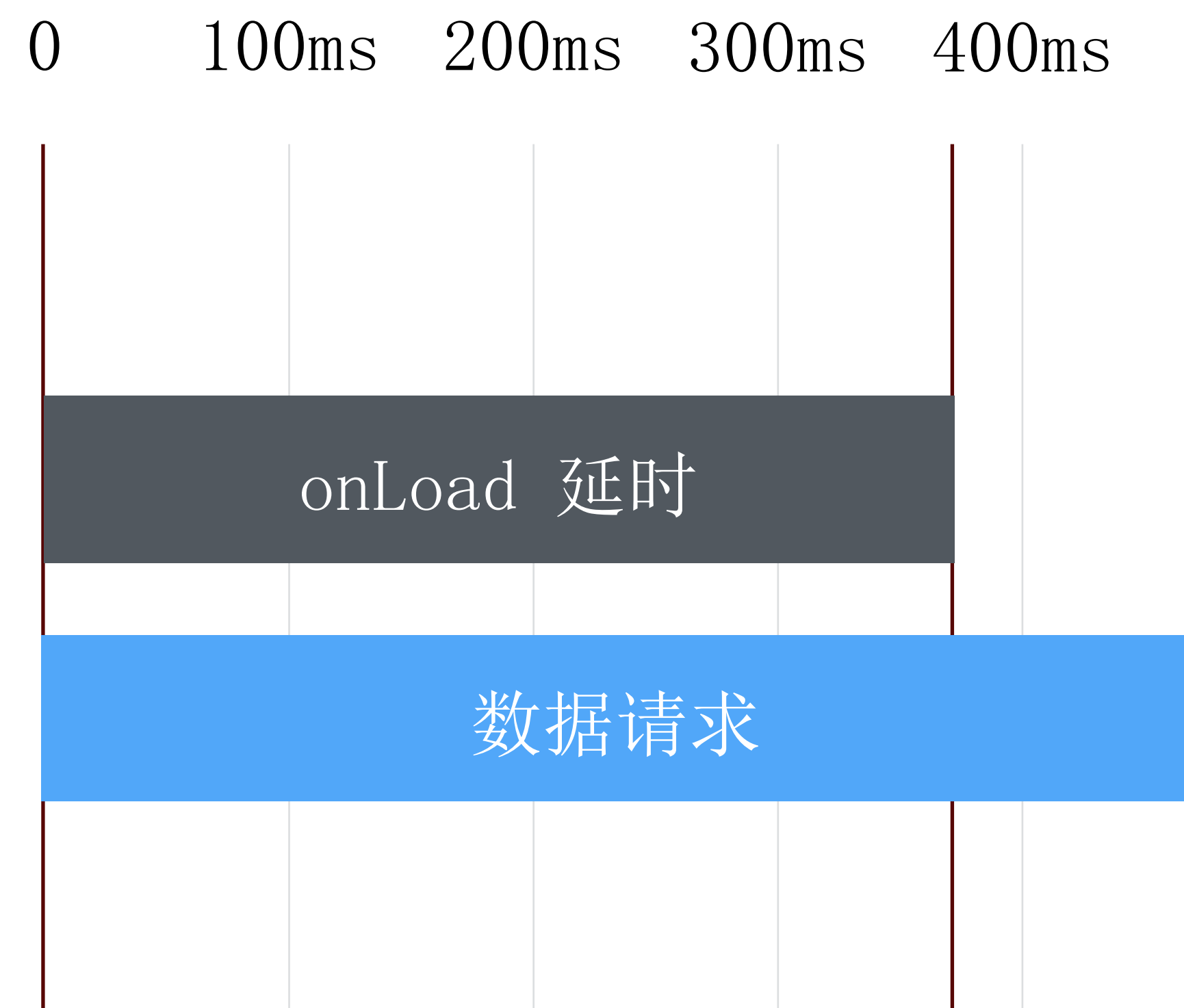
性能调优

微信小程序页面跳转预加载

02



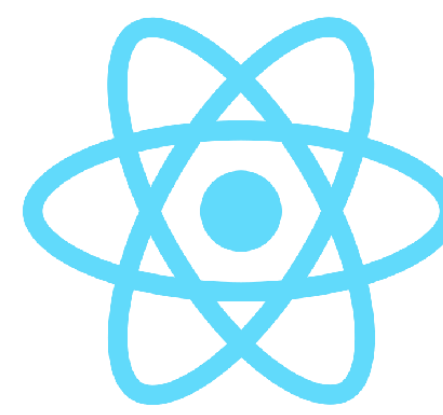
componentWillPreload



4

打磨与探索





OF COURSE!



打磨与探索

微信小程序代码转 Taro 代码

02



EastWorld/wechat-app-mall ★6000+

tumobi/nideshop-mini-program ★3000+

RebeccaHanjw/weapp-wechat-zhihu ★1000+

jectychen/wechat-v2ex ★400+

icindy/wxParse ★5000+

打磨与探索

接入 Taro Doctor 便捷测试

config/index.js

```
const config = {  
  projectName: 'Demo',  
  designWidth: 750,  
  sourceRoot: 'src',  
  outputRoot: 'dist',  
  plugins: {  
    ...  
  },  
  wxapp: {  
    ...  
  },  
  something: 'xxx'  
}
```

拼写错误

配置项不存在

打磨与探索

小程序使用 webpack 打包

02



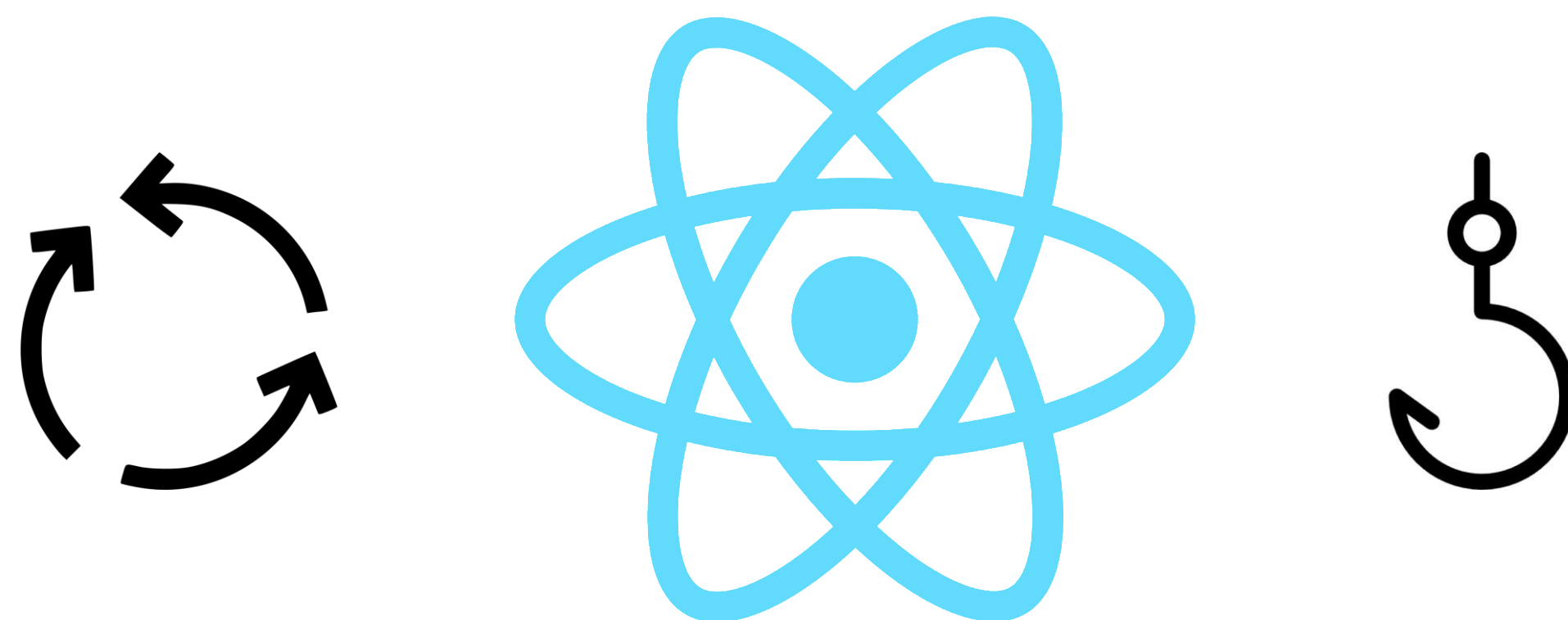
X



打磨与探索

同步 React 所有特性

02



打磨与探索

参与贡献

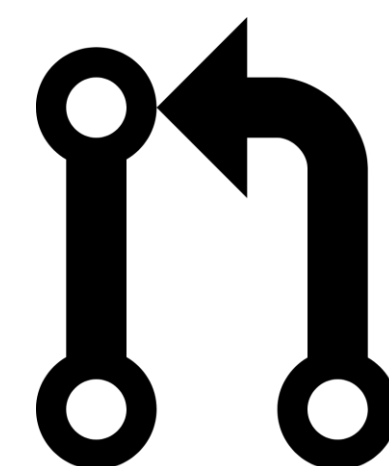
02



文档



Issues



Pull requests



生态

THANKS
FOR YOUR WATCHING

2 0 1 8 年 终 盛 典 · 深 圳

