



WEEX Conf

淘宝双促中的 Rax



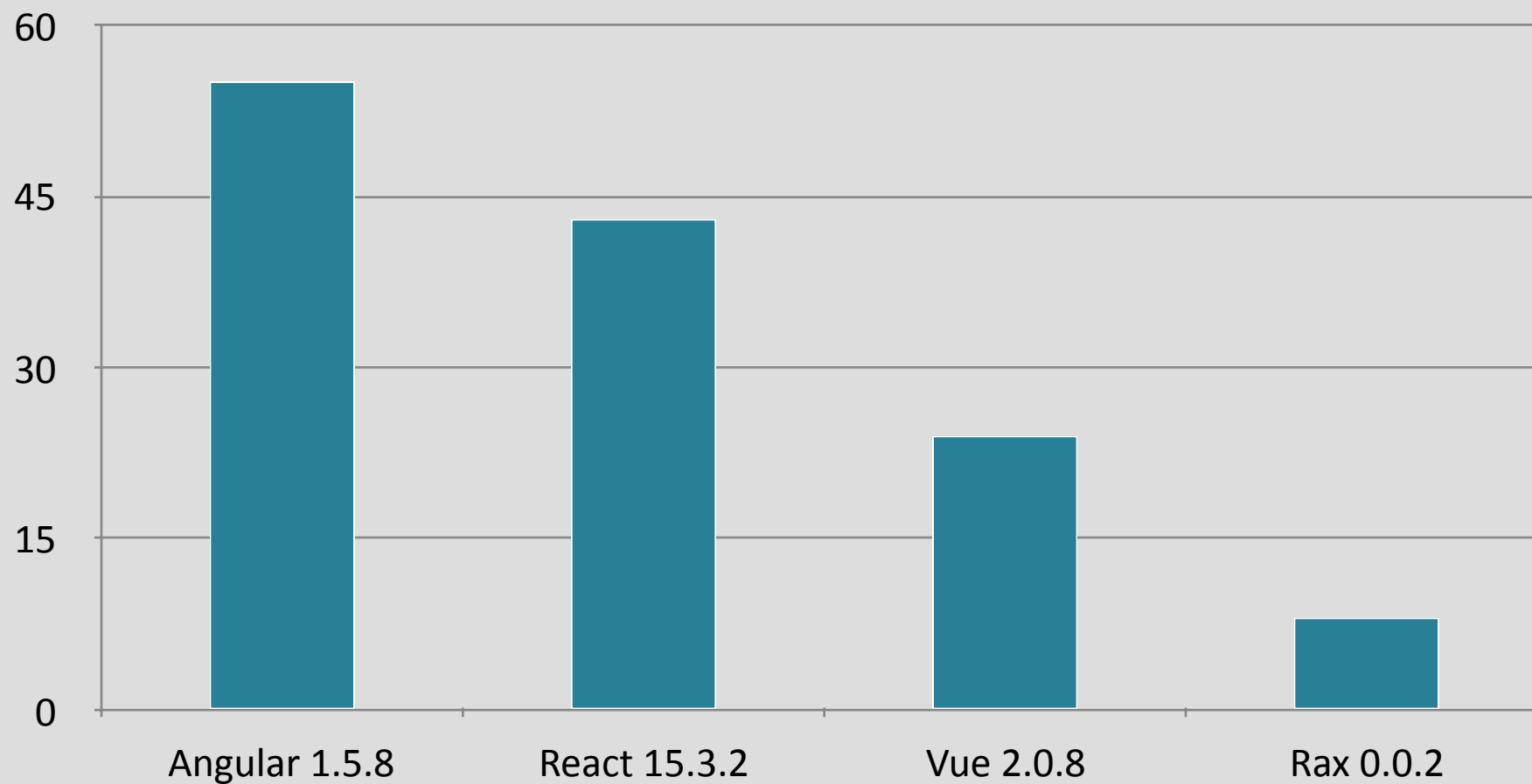
1. Rax

Rax 是什么

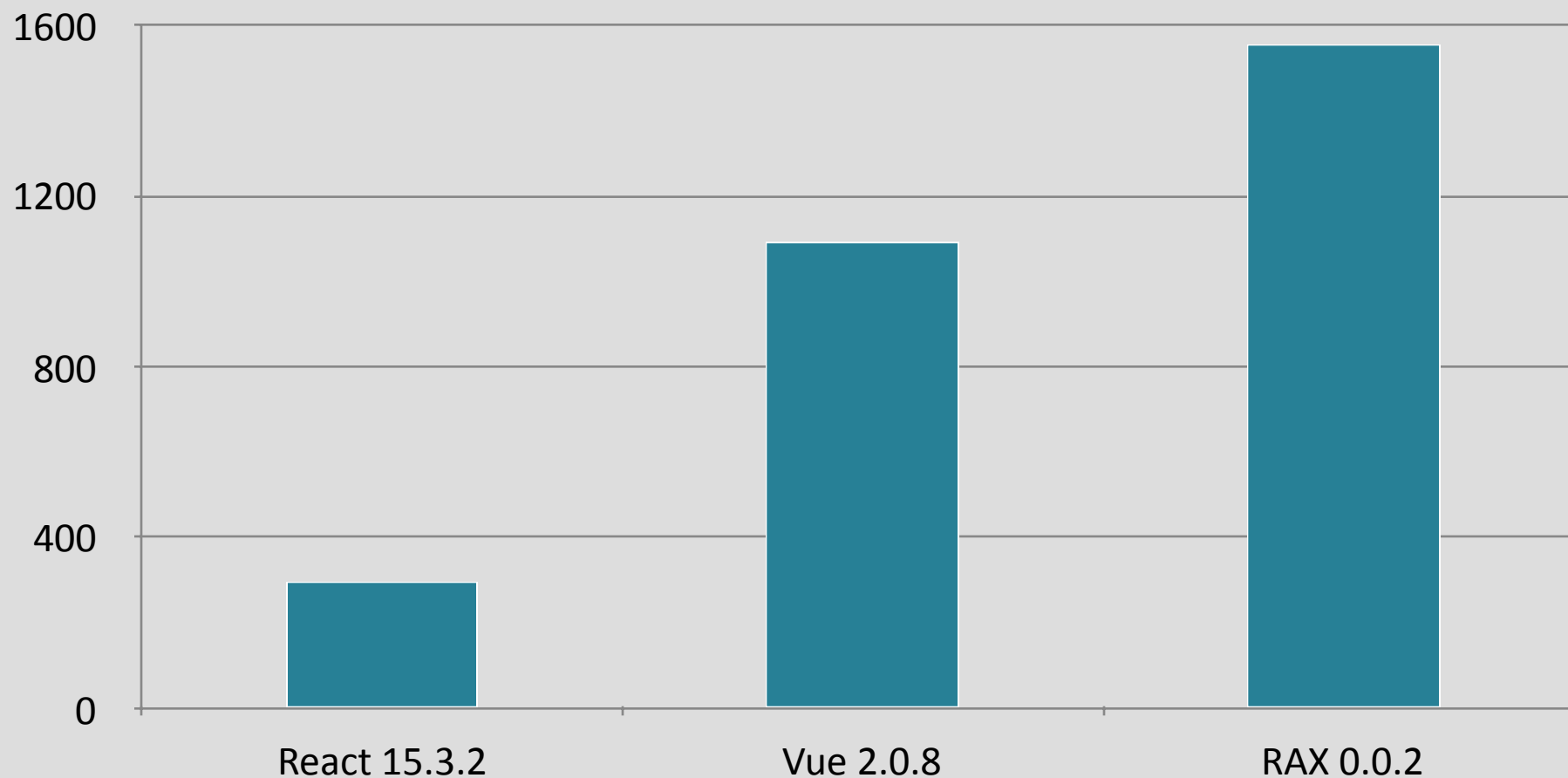
Rax 是一个基于 React 写法的跨容器的 js 框架

- ➡ React：你会用 React，那么你会很快上手 Rax
- ➡ 跨容器：Browser、WEEX、Node.js

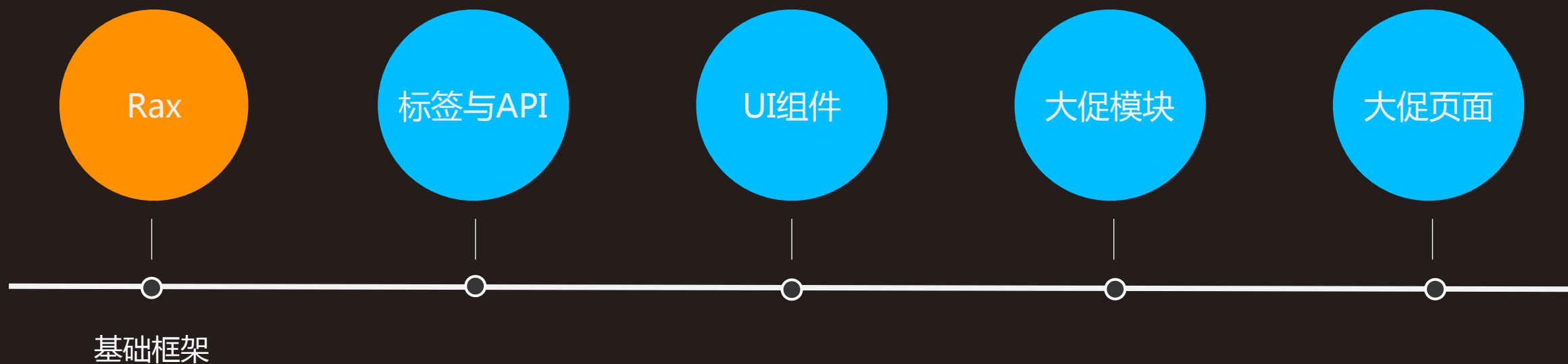
Tax 优势 (gzip 大小)



Rax 优势 (服务端渲染对比 op/s)



Rax 在双促中扮演了怎样的角色



2. 对大促的基础支持

大促中的应用



双11全球狂欢节

1212 淘宝亲亲节

API

Window

Location

StyleSheet

Appear

Toast

Dimensions

Alert

Webp

Evn

Promise

Spm

Asyncstorage

User

Panresponder

Fetch

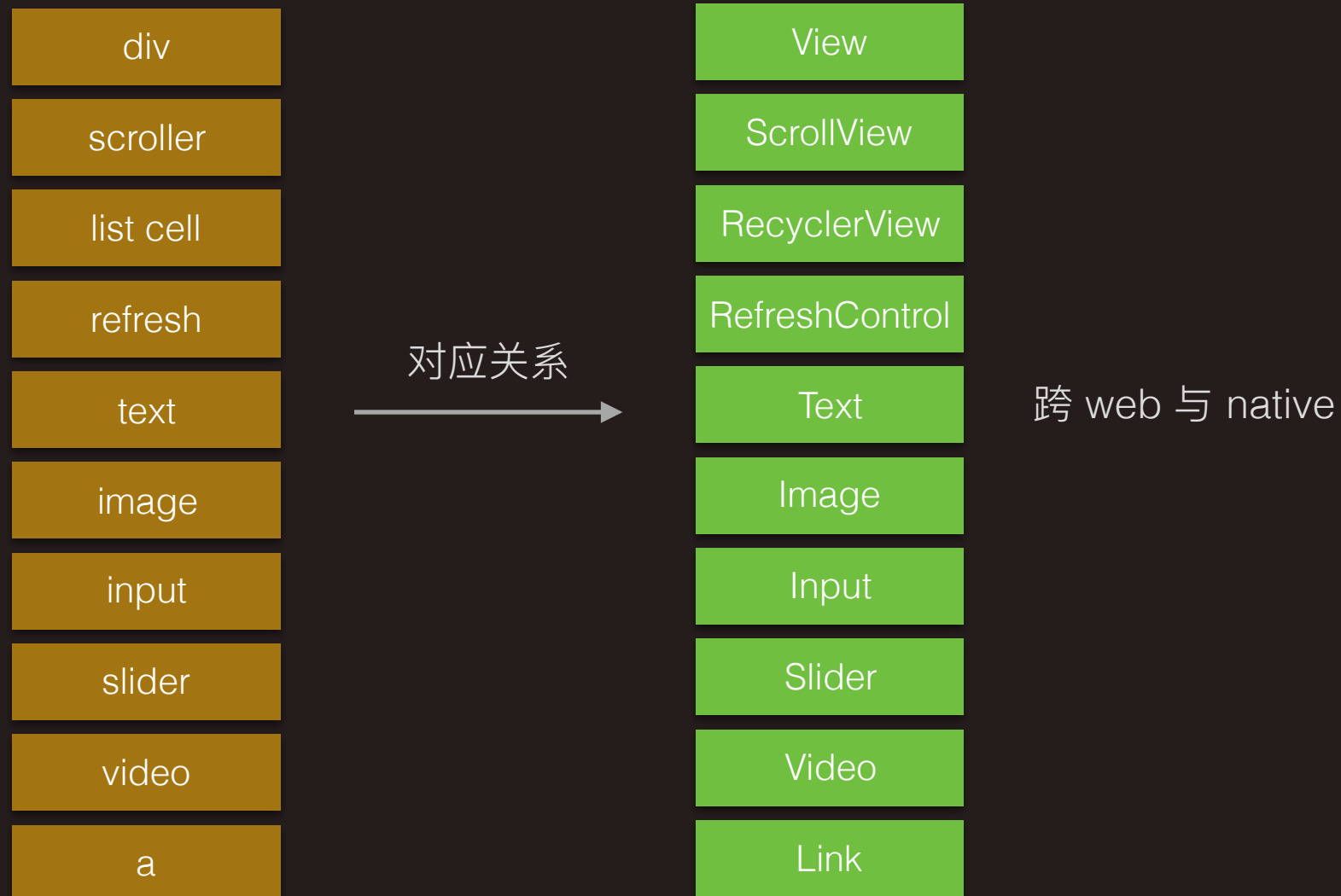
Downgrade

Windvane

Goldlog

Mtop

基础标签



双促模块的组成方式

大促模块

App级 业务上的公用

底部bar

Gotop

导航

Page级 滚动容器内

内容电梯

橱窗列表

RAXUI 组件支持

Tabheader

Player

Picture

页面渲染 rax-page

滚动容器

渲染优化

模块逻辑

Components 基础标签集合

Image

Link

Slider

Text

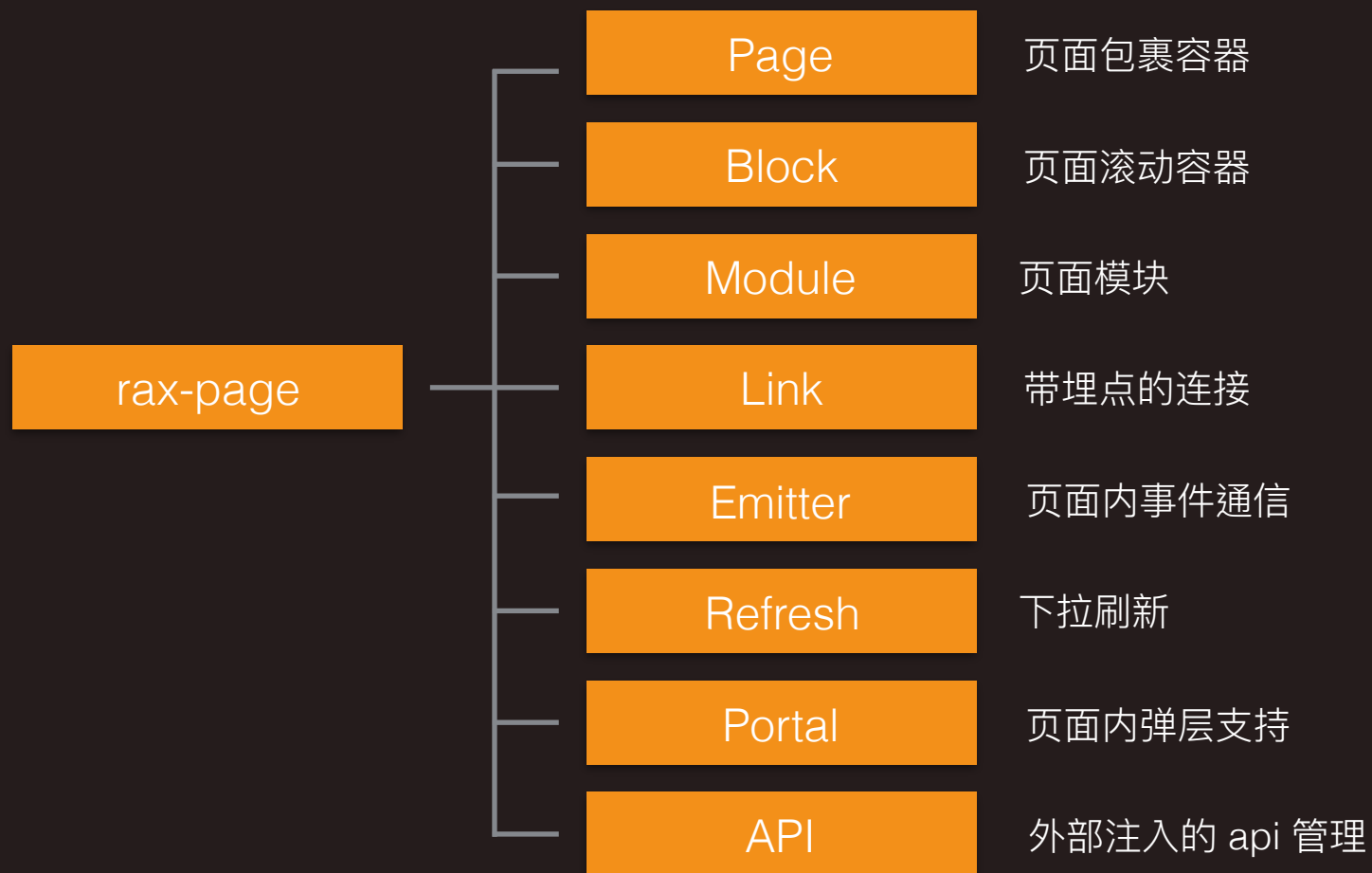
View

Video

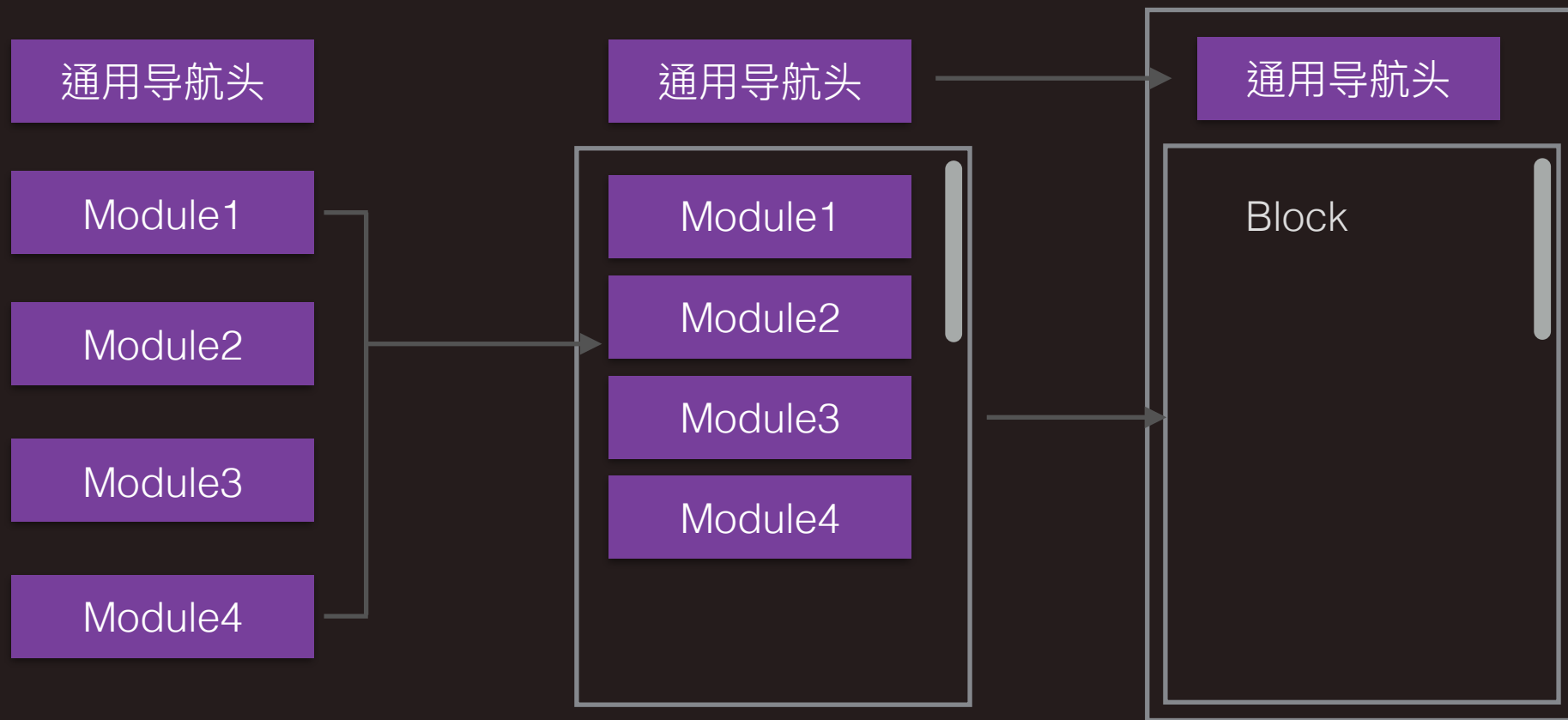
ScrollView

3. 淘宝双促的页面渲染逻辑

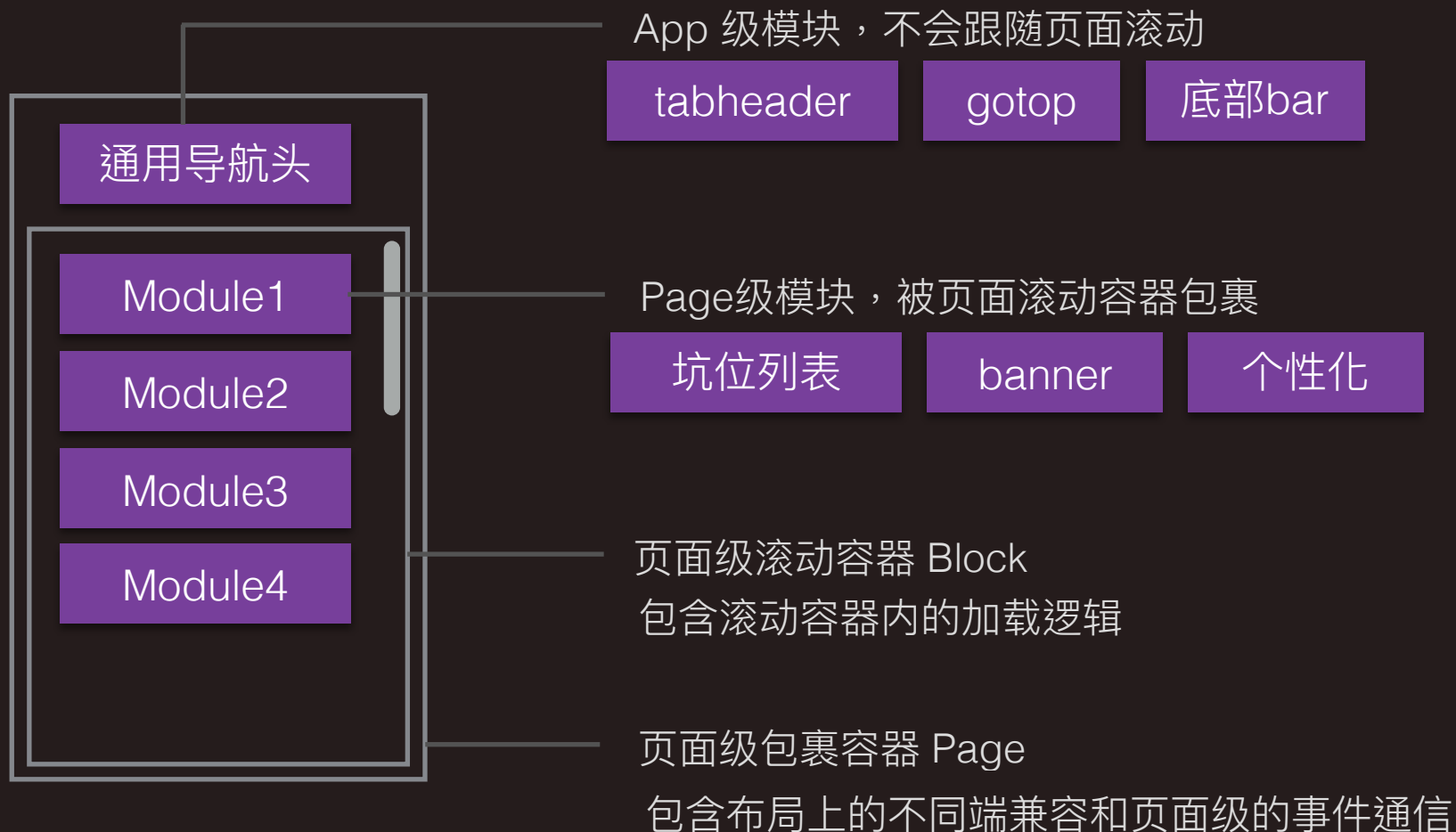
页面渲染层的基本构成



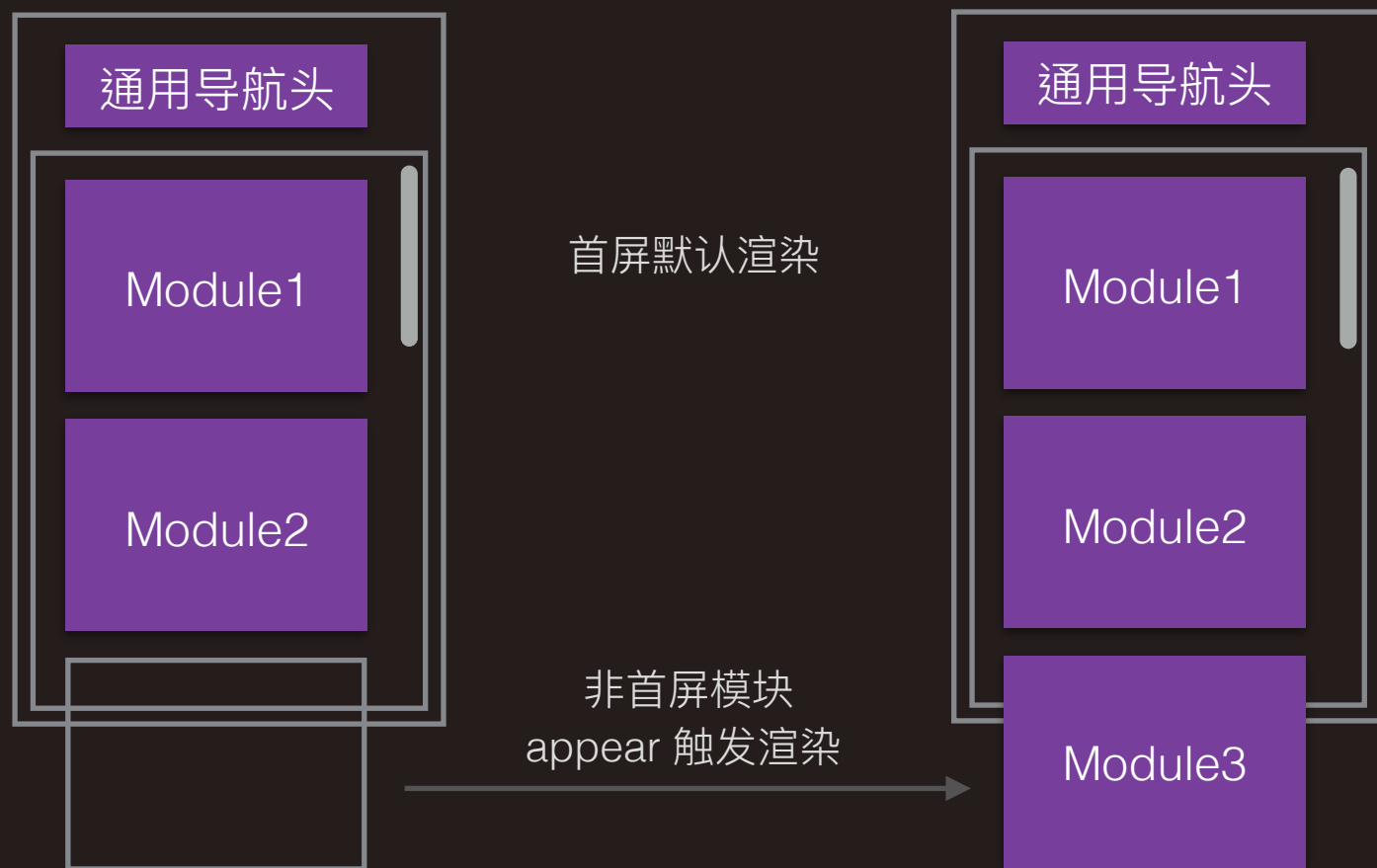
模块的组织



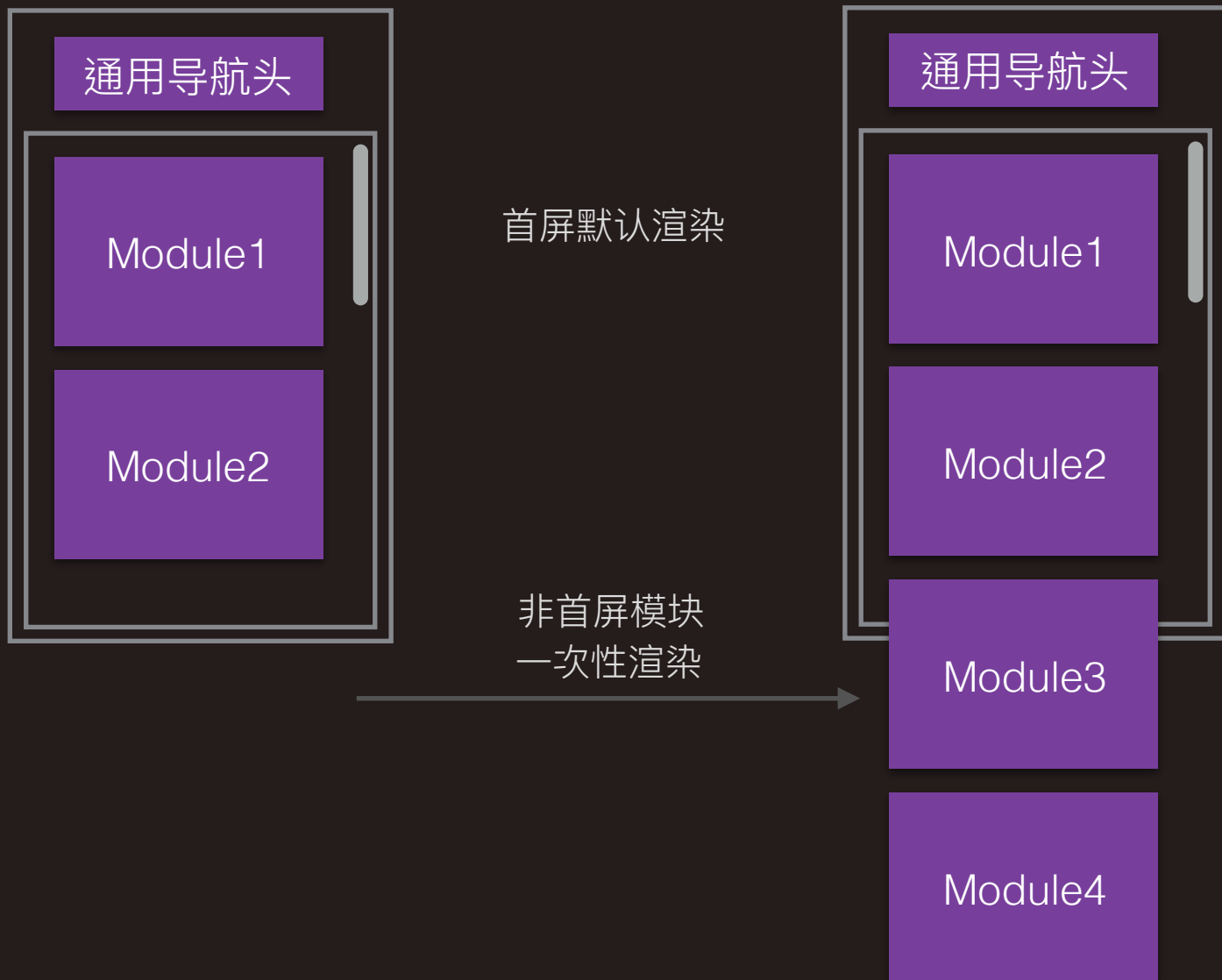
模块的组织



h5 页面渲染逻辑

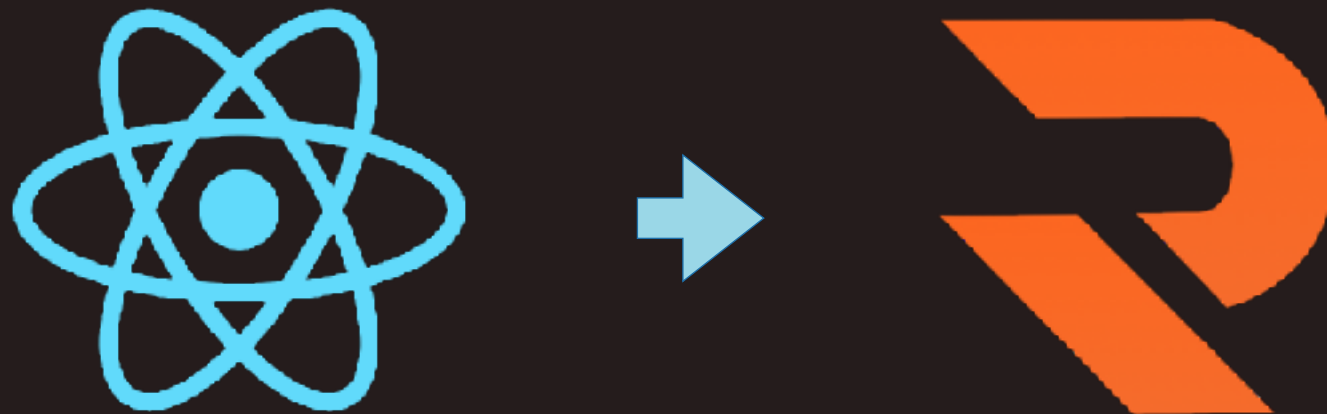


week 页面渲染逻辑



4. 页面内元素的组织

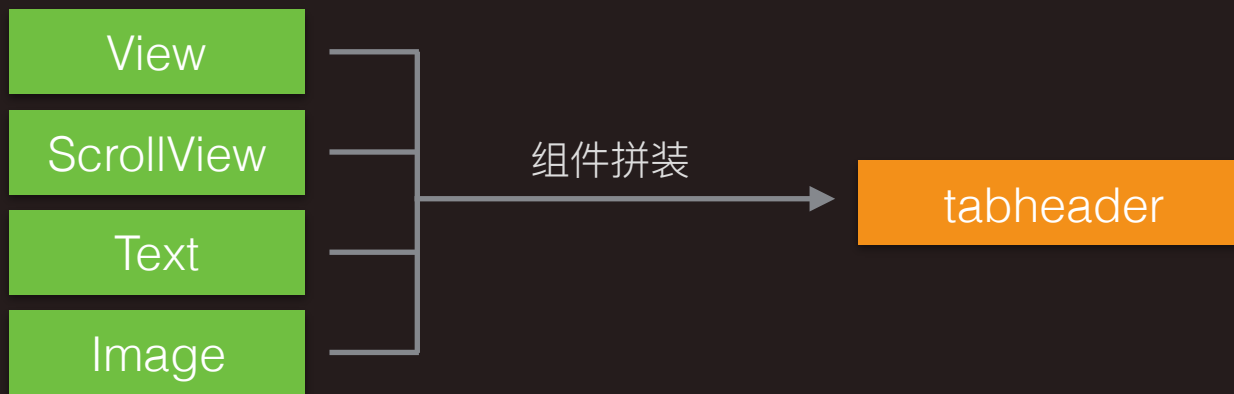
对于模块开发者的影响



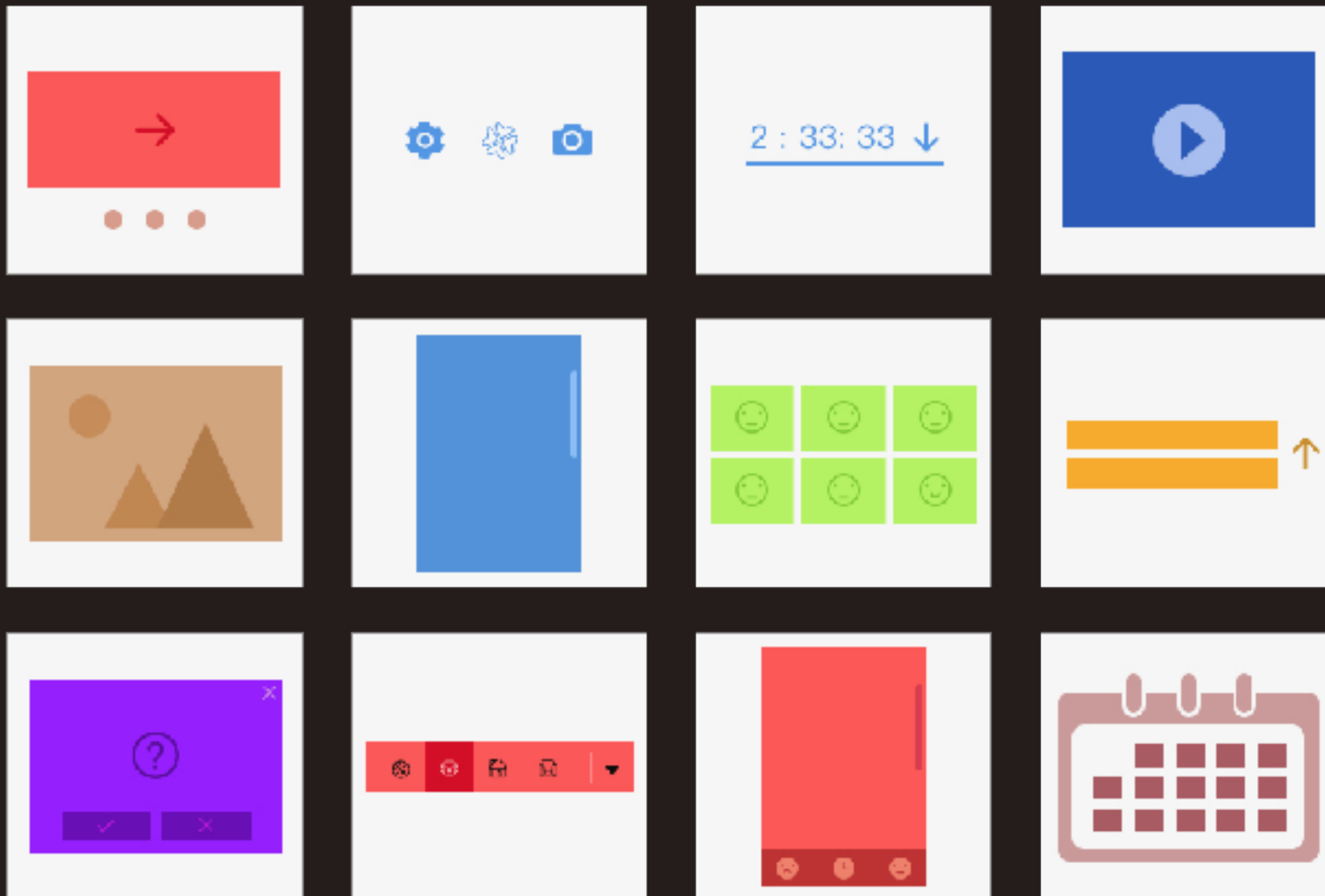
模块示例代码片段

```
class Module extends Component {  
  
  static contextTypes = {  
    page: PropTypes.object  
  };  
  static getModuleRowHeight = (attrs) => {  
    return 100;  
  };  
  static moduleRenderMode = 'repeat';  
  
  componentWillMount() {  
    let page = this.context.page;  
    if (page && this.props.isFirstRepeatModule) {  
      setTimeout(() => {  
        // 根据业务逻辑重设列表内容  
        this.props.setDataSource(['异步数据1', '异步数据2']);  
      }, 3000);  
    }  
  }  
  
  render() {  
    return (<View>data : {this.props.data}</View>);  
  }  
}
```

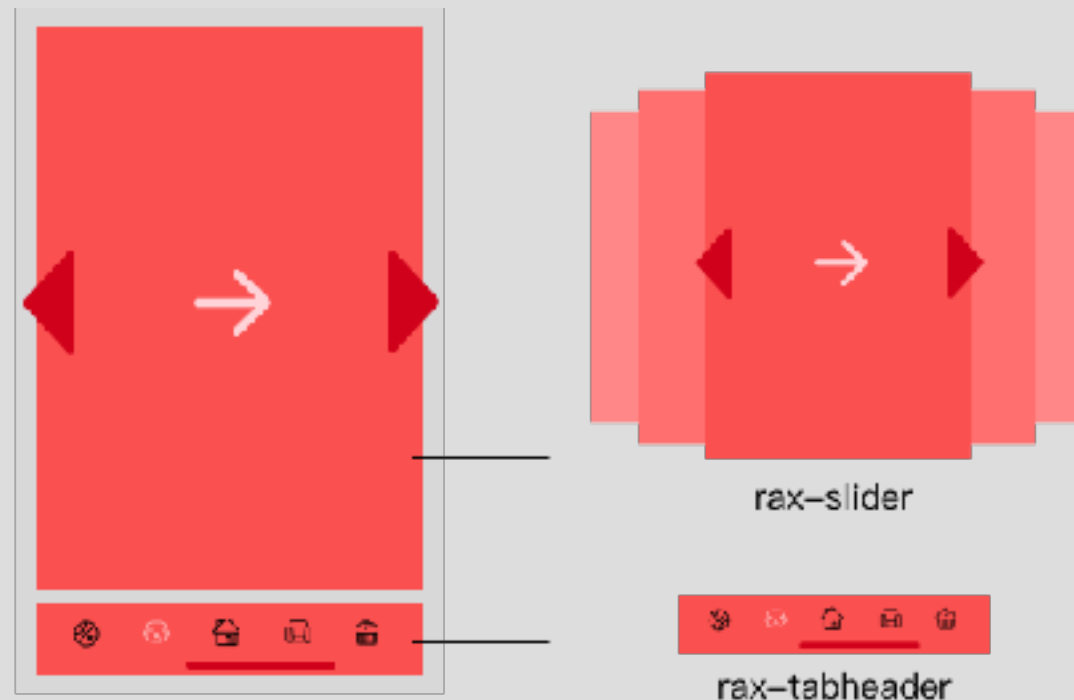
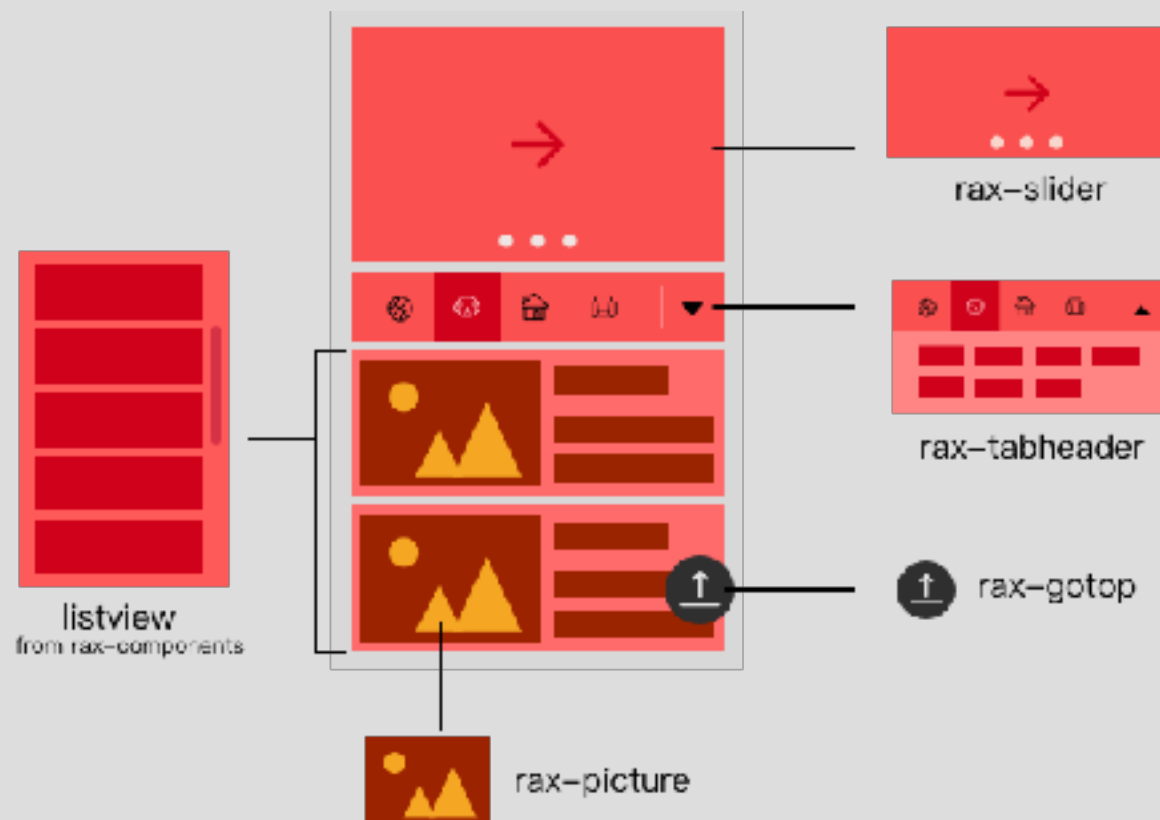
从标签到组件



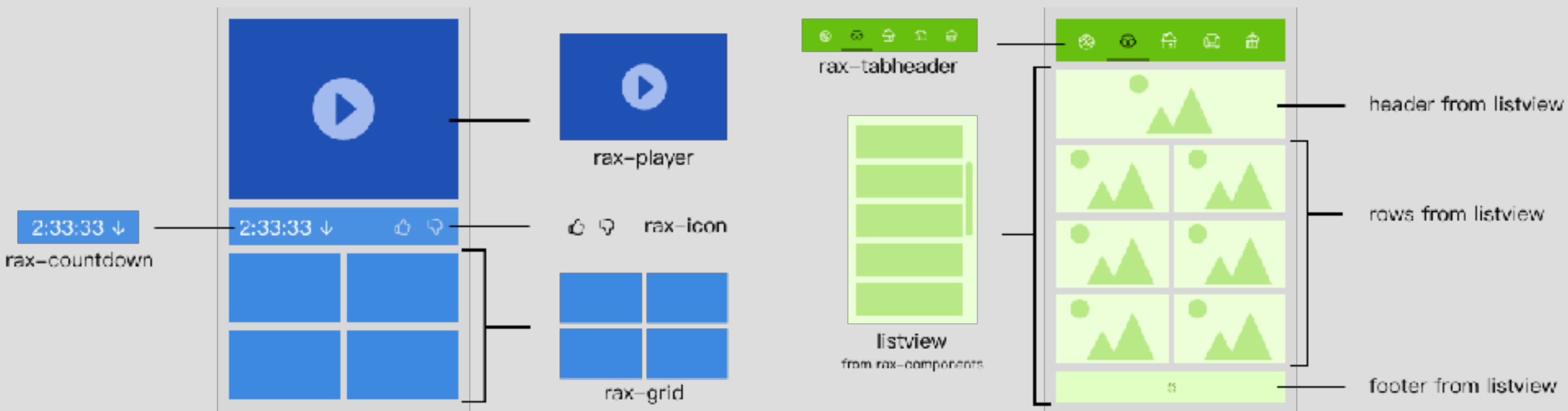
RaxUI



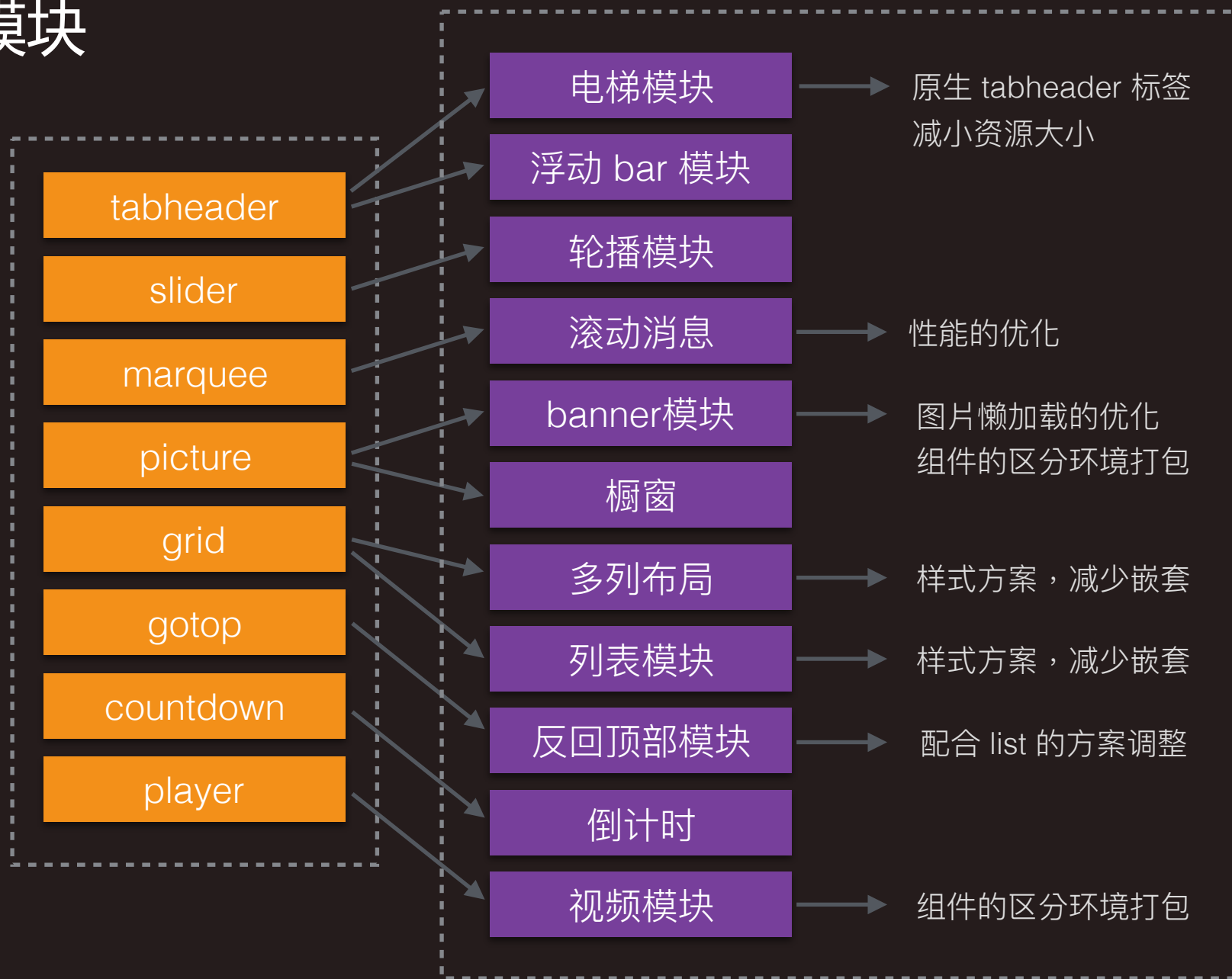
页面中的组件示例



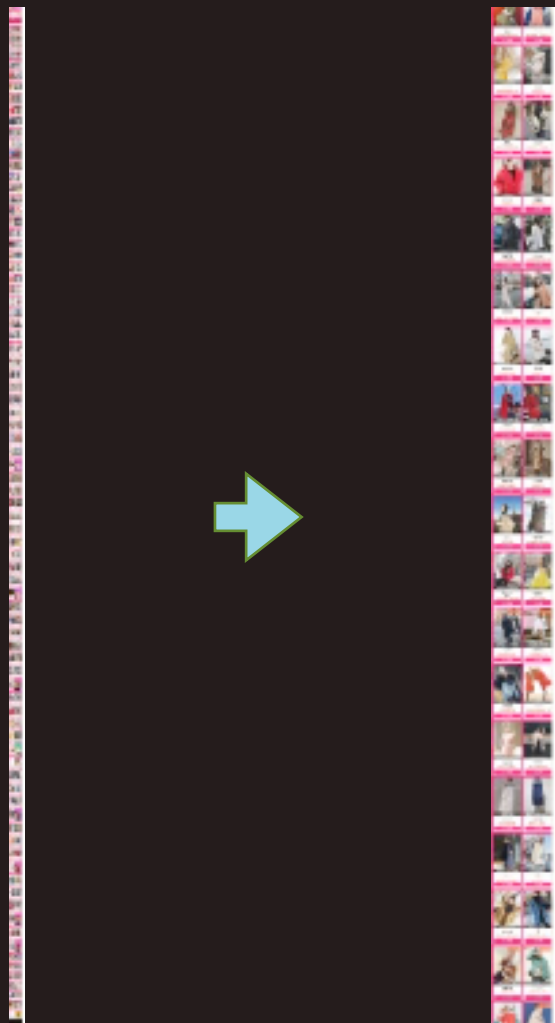
页面中的组件示例



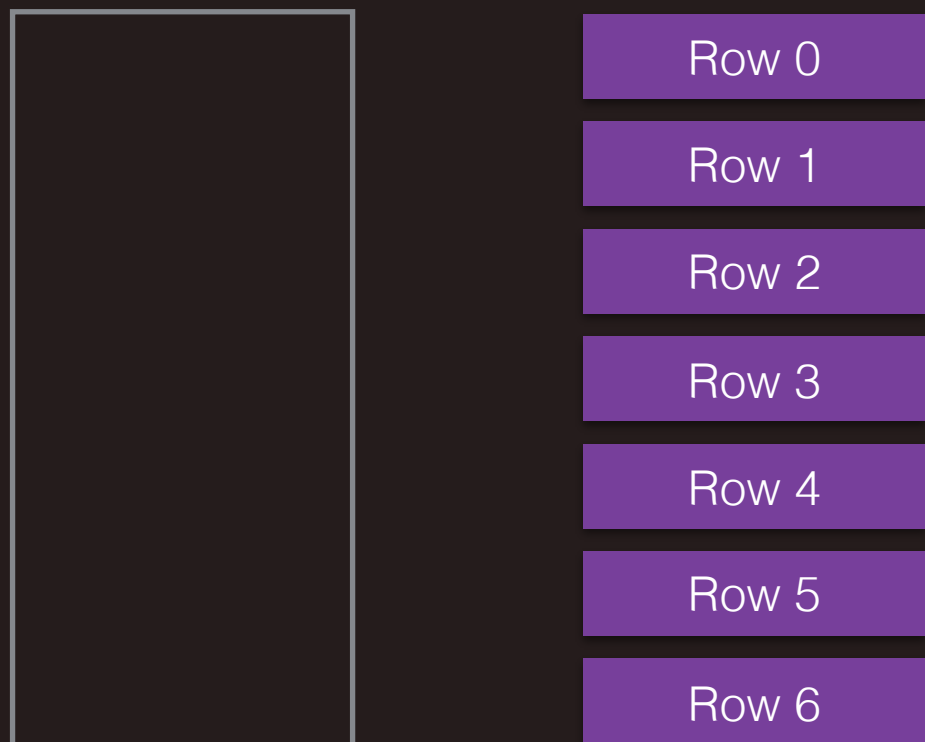
从组件到模块



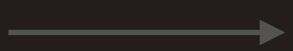
一个长列表的例子



repeat 类型模块详解



占位容器



占位容器

- h5下占位容器逻辑可以减少页面上的节点
- 渲染时占位容器将拆分为多行模块
- 单个 repeat 模块分段加载，以5行进行分段
- 最后一行模块的 appear 事件可以实现无限加载



<https://github.com/alibaba/weex>