

React Native 跨平台响应式布局

...

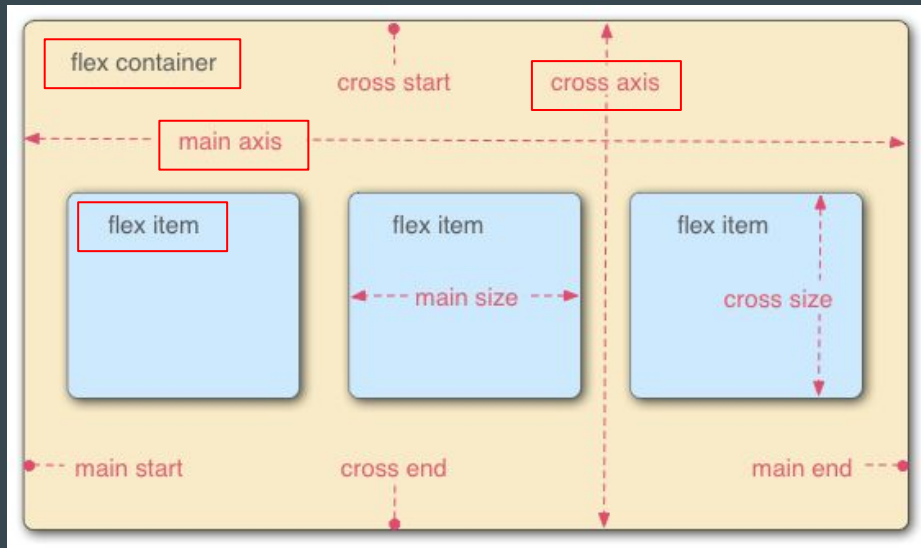
张臻

概要

1. Flexbox布局原理介绍
2. Platform API介绍
3. Dimensions API介绍
4. Flexbox布局案例(Demo)
5. 响应式布局案例(Demo)
6. 相关开源项目, 工具介绍

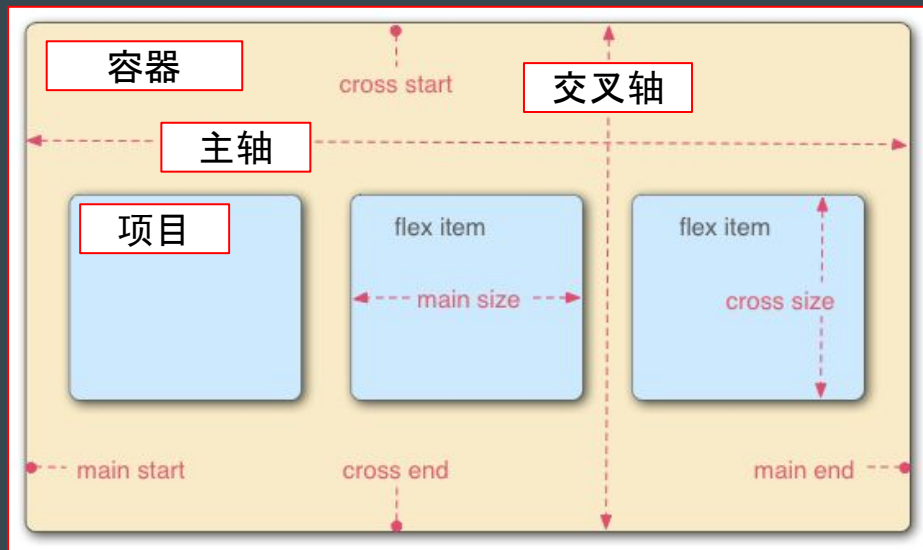
Flexbox用语

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox



Flexbox用语

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox



Flexbox算法概要

1. 首先在主轴上按各项目默认长宽进行排布
2. 如果主轴方向上需要换行, 进行换行处理
3. 计算每一行主轴上是否有剩余长度
4. 如果剩余长度 >0 , 拉长各个项目, 以填满剩余空间
5. 如果剩余长度 <0 , 则缩短各个项目, 确保没有要素溢出
6. 处理各个项目交叉轴上的排布

容器属性(作用于项目全体的布局)

- flexDirection
 - 设置容器的主轴方向
 - React Native内默认为column
- flexWrap
 - 设置项目在主轴方向上的换行方式
- justifyContent
 - 设置项目在主轴上的布局方式
- alignItems
 - 设置项目在交叉轴上的布局方式

项目属性(作用于项目各自的布局)

- width/height/minWidth/minHeight/margin
- flexBasis
 - 设置开始分配空间之前, 项目在主轴上所占长度, 效果上相当于 width/height 指定
- flexGrow
 - 设置主轴上空间充分, 需要伸长时, 项目所伸长的比重
- flexShrink
 - 设置主轴上空间不足, 需要收缩时, 项目所收缩的比重
- flex
 - $N > 0$ 时, 设置主轴上所占长度, 效果上相当于设置了 flexGrow: N, flexShrink: 1
 - $N = 0$ 时, 根据 width/height 来设置尺寸, 效果上相当于设置了 flexGrow: 0, flexShrink: 0
 - $N = -1$ 时, 根据 width/height 来设置尺寸, 主轴空间不足时进行收缩, flexGrow: 0, flexShrink: 1
- alignSelf
 - 设置项目在交叉轴上的布局方式

Flexbox算法概要

1. 首先在主轴上按指定长宽 (flexBasis/width/height/margin等) 排布
2. 如果主轴方向上需要换行, 根据flexWrap处理换行
3. 计算每一行主轴上是否有剩余长度
4. 如果剩余长度 >0 , 按flexGrow对项目进行分配
5. 如果剩余长度 <0 , 则按flexShrink对项目进行分配
6. 根据alignItems和alignSelf处理各个项目交叉轴上的排布

Platform API

- 用于判断OS种类
 - Platform.OS
 - Platform.select()
- 用于判断OS版本
 - Platform.Version

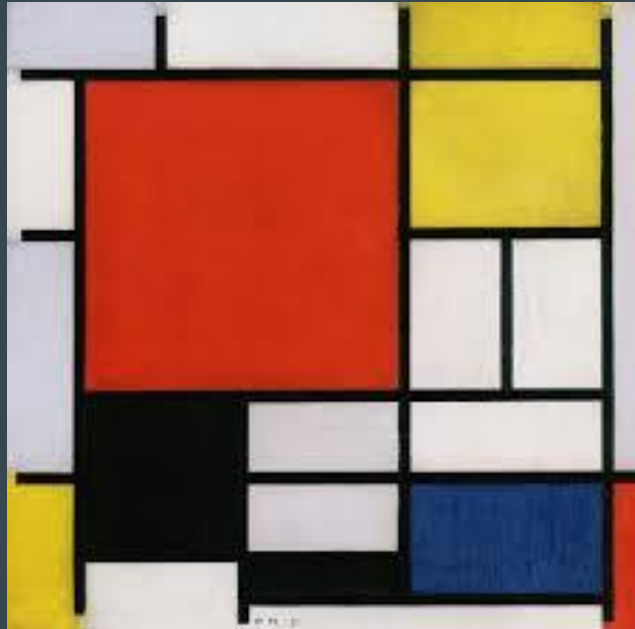
Dimensions API

- 用于获取屏幕大小
 - `Dimensions.get('window')`
 - 配合`<View />`组建的`onLayout`使用

Flexbox布局案例(Demo)

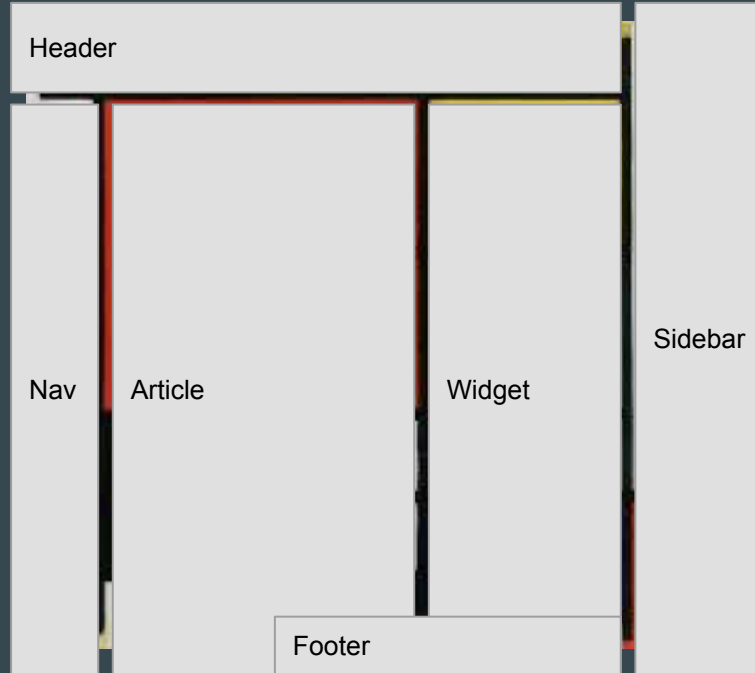
Composition with Large Red Plane, Yellow, Black, Gray, and Blue (1921)

- <https://www.pubhist.com/w28886>



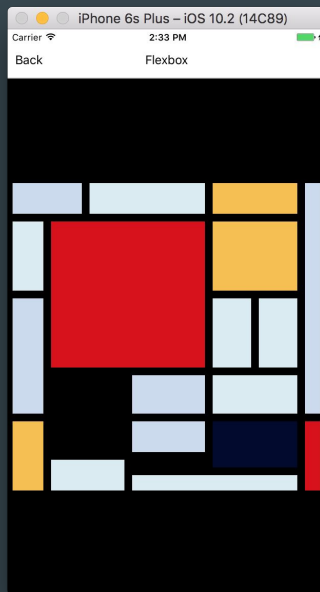
Composition with Large Red Plane, Yellow, Black, Gray, and Blue (1921)

- <https://www.pubhist.com/w28886>



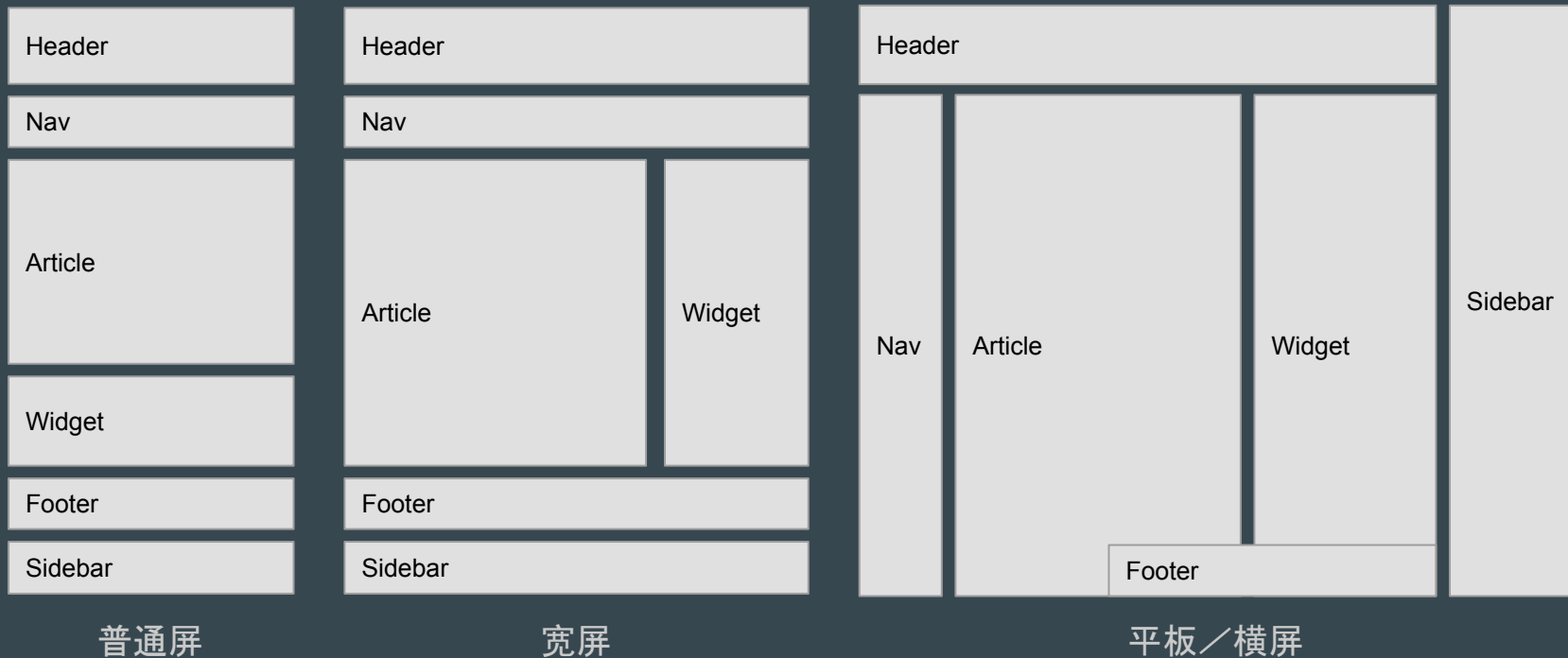
Flexbox

- <https://github.com/CSDN-mobilehub/react-native-responsive-layout>



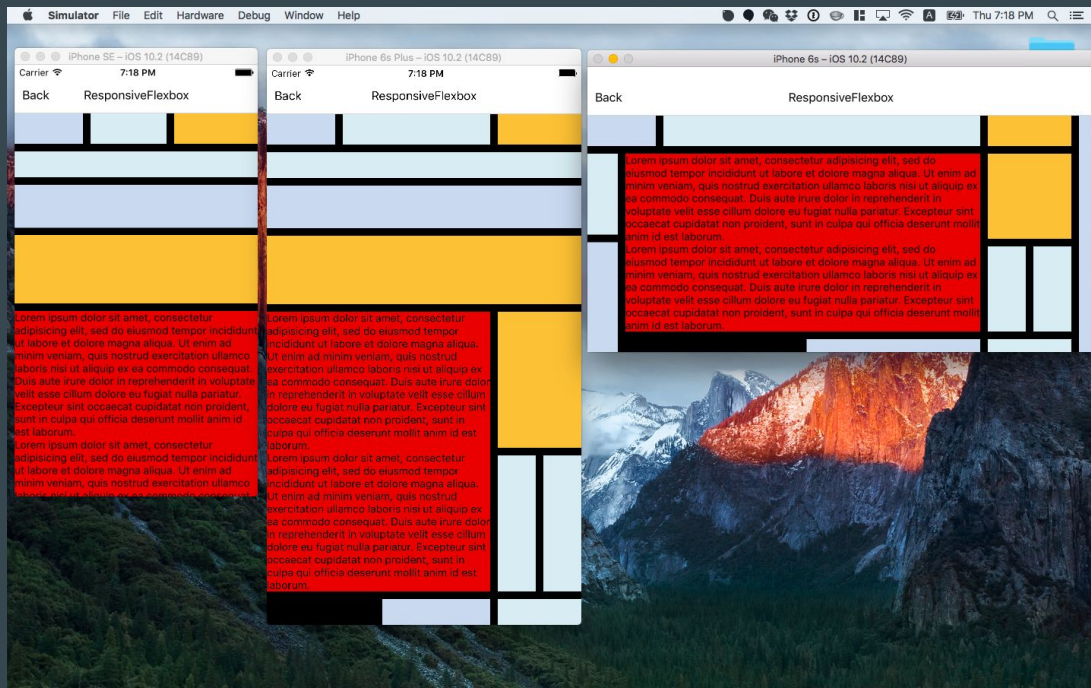
Composition with Large Red Plane, Yellow, Black, Gray, and Blue (1921)

- <https://www.pubhist.com/w28886>



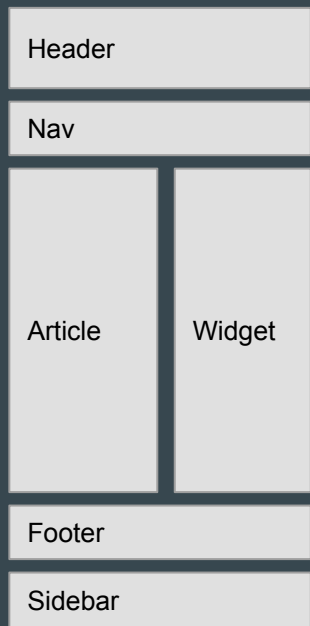
ResponsiveFlexbox

- <https://github.com/CSDN-mobilehub/react-native-responsive-layout>

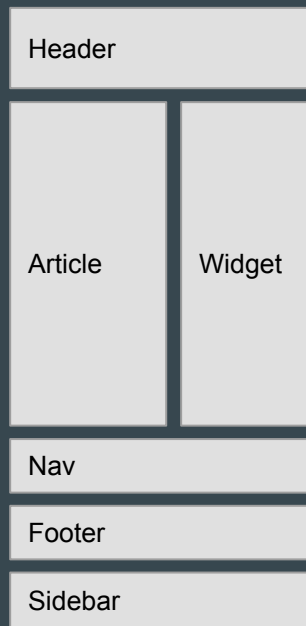


Composition with Large Red Plane, Yellow, Black, Gray, and Blue (1921)

- <https://www.pubhist.com/w28886>



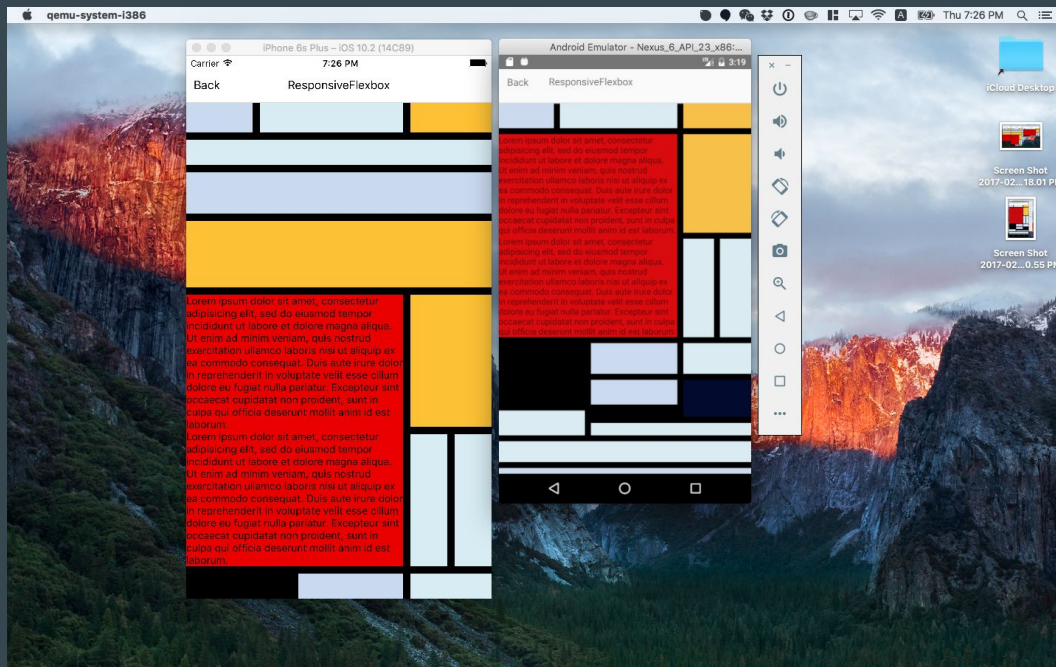
iOS



Android

ResponsiveFlexbox

- <https://github.com/CSDN-mobilehub/react-native-responsive-layout>



Nuclide

- Facebook在Atom的基础上开发的IDE
- <https://nuclide.io/>
- 杀手锏:利用React Native Inspector调试布局

fbsimctl

- Facebook开发的命令行工具, 可以同时启动并操控多个iOS Emulator
- <https://github.com/tapthaker/FBSimulatorClient>
- 杀手锏: 可以同时确认多个机型上的显示效果, 方便调试响应式布局

react-native-web-player

- 可以在浏览器内运行的react-native测试环境
- <https://cdn.rawgit.com/dabbott/react-native-web-player/gh-v1.9.1/index.html>
- 杀手锏:节省构建测试环境的时间和精力, 在浏览器里轻松实验布局效果
- 杀手锏:支持URL分享

react-native-extended-stylesheet

- 开源项目, RN的StyleSheet的基础上拓展支持数种CSS中常用的功能
- <https://github.com/vitalets/react-native-extended-stylesheet>
- 杀手锏:支持Media Queries语法, 支持百分比尺寸(相对于屏幕长宽)
- 掉过坑:不对应设备旋转

<https://github.com/vitalets/react-native-extended-stylesheet/issues/9>

uranium & react-native-match-media

- 两者并用可以支持Media Queries语法
- <https://github.com/tuckerconnelly/uranium>
- <https://github.com/tuckerconnelly/react-native-match-media>
- 杀手锏:支持Media Queries语法, 支持设备旋转, 并且兼容react-native-web
- 副作用:需要指定JS全局变量 `global.matchMedia`

