**Taro**

文档

https://nervjs.github.io/taro/docs/GETTING-STARTED.html

1.安装

npm install -g @tarojs/cli

2.创建项目

taro init myApp

3.更新当前项目及依赖

taro update self && taro update project  
4.运行

小程序模式 npm run dev:weapp

H5模式 npm run dev:h5

RN 模式 npm run dev:rn

页面生命周期



1.componentWillMount 程序被载入时(未触发render)

2.componentDidMount 程序被载入时(已触发render)

3.componentWillUnmount 页面卸载

4.componentDidShow 进入路由时触发

5.componentDidHide 离开路由时触发

路由

路由定义

需要在项目入口文件 app.js 中 config 的 pages 数组中进行指定



路由跳转

在新页面打开

Taro.navigateTo({

url: '/pages/page/path/name?id=1'

})

在当前页面打开

Taro. redirectTo ({

url: '/pages/page/path/name?id=1'

})

返回上一页

Taro.navigateBack({

delta: 2

})

获取路由参数

this.$router.params

单位(设计稿50px)

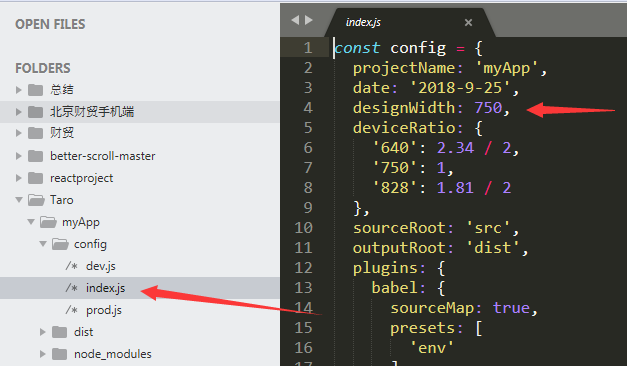
例如设计稿宽度为750px,元素长度为100px,则样式直接写为100px,Taro会自动转换单位.

注:不需要被转换的单位(px)则使用(PX);

Js中使用动态宽度需要使用

Taro.pxTransform(10) // 小程序：rpx，H5：rem

修改设计稿宽度,修改config文件夹下的index.js文件



小程序打开

需要关闭



http请求

Taro.request({

url:'',//请求地址

data:{},//携带的数据

header:'',//请求头

method:'GET',//请求方式,大写

jsonp:'callback',//jsonp请求,值为回调函数,仅H5使用

}).then(res => {

console.log(res);

}).catch(err => {

console.log(err);

})

数据缓存

异步存贮

Taro.setStorage({

key: 'key', //key值

data: 'value' //value值

}).then(res => {

console.log(res);

})

异步获取

Taro.getStorage({ key: 'key' }).then(res => {

console.log(res.data);

})

异步获取当前storage的相关信息

Taro.getStorageInfo().then(res => {

console.log(res.keys);

})

异步删除

Taro.removeStorage({ key: 'key' }).then(res => {

console.log(res);

})

异步清空缓存

Taro.clearStorage()

同步存贮

Taro.setStorageSync('key', 'value')

同步获取

const data = Taro.getStorageSync('key')

同步获取当前storage的相关信息

const res = Taro.getStorageInfoSync()

同步删除

Taro.removeStorageSync('key')

同步清空缓存

Taro.clearStorageSync()

系统信息

异步获取系统信息

Taro.getSystemInfo().then(res => {

console.log(res.brand);//手机品牌

console.log(res.model);//手机型号

console.log(res.system);//操作系统版本

console.log(res.pixelRatio);//设备像素比

console.log(res.screenWidth);//屏幕宽度

console.log(res.screenHeight);//屏幕高度

console.log(res.windowWidth);//可使用窗口宽度

console.log(res.windowHeight);//可使用窗口高度

console.log(res.platform);//客户端平台

console.log(res.language);//微信设置的语言

console.log(res.version);//微信版本号,H5不支持

console.log(res.statusBarHeight);//状态栏的高度,H5不支持

console.log(res.fontSizeSetting);//用户字体大小设置。以“我-设置-通用-字体大小”中的设置为准，单位：px,H5不支持

console.log(res.SDKVersion);//客户端基础库版本,H5不支持

})

同步获取系统信息

const res = Taro.getSystemInfoSync()

交互

显示提示框

Taro.showToast({

title: '成功',//内容

icon: 'success',//图标,"success", "loading", "none"

duration: 1500,//提示的延迟时间

})

隐藏提示框

Taro.hideToast()

显示加载框

Taro.showLoading({

title: 'loading',//内容

mask:false,//是否显示遮罩层,防止触摸穿透

})

隐藏加载框

Taro.hideLoading()

模块框

Taro.showModal({

title: 'xxx',//标题

content: 'hello world',//内容

showCancel:true,//是否显示取消按钮

cancelText:'取消',//取消按钮的文字

cancelColor:'#000000',//取消按钮的文字颜色

confirmText:'确定',//确定按钮的文字

confirmColor:'#3CC51F',//确定按钮的文字颜色

}).then(res => {

console.log(res.confirm)//true表示点了确定

console.log(res.cancel)//true表示点了取消

})

上拉菜单

Taro.showActionSheet({

itemList: ['a', 'b', 'c'],//每一项,最多6个

itemColor:'#000000',//文字颜色

}).then(res => {

console.log(res.tapIndex);//用户点击的按钮，从上到下的顺序，从0开始

}).catch(err => {

//而当用户点击取消或蒙层时，Promise 会 reject

console.log(res.errMsg);

})

组件

容器 <View></View>

滚动容器<ScrollView></ScrollView>

轮播图<Swiper></Swiper>

图标<Icon></Icon>

文本<Text></Text>

进度条<Progress></Progress>

富文本<RichText></RichText>

按钮<Button></Button>

复选框

<CheckboxGroup>

<Checkbox></Checkbox>

<Checkbox></Checkbox>

<Checkbox></Checkbox>

</CheckboxGroup>

单选框

<RadioGroup>

<Radio></Radio>

<Radio></Radio>

<Radio></Radio>

</RadioGroup>

表单<Form></Form>

输入框<Input></Input>

Label <Label></Label>

选择器<Picker></Picker>

嵌入页面的选择器

<PickerView>

<PickerViewColumn></PickerViewColumn>

<PickerViewColumn></PickerViewColumn>

<PickerViewColumn></PickerViewColumn>

</PickerView>

滑条 <Slider></Slider>

开关 <Switch></Switch>

文本域<Textarea></Textarea>

音频<Audio></Audio>

视频<Video></Video>

图片<Image />

相机<Camera></Camera>

环境判断

获取当前的环境值

Taro.getEnv()

环境

Taro.ENV\_TYPE

ENV\_TYPE.WEAPP 小程序环境

ENV\_TYPE.WEB WEB(H5)环境

ENV\_TYPE.RN ReactNative环境

异步编程

npm install --save @tarojs/async-await

