

_Wake: 新手入门攻略《二》配置及生命周期，布局基础&样式基础

一、app 生命周期

```
App({  
  onLaunch: function () {  
    //App Launch ,生命周期函数--监听小程序初始化  
  
    // 当小程序初始化完成时，会触发 onLaunch（全局只触发一次）  
  
  },  
  
  onShow: function () {  
    // onShow,生命周期函数--监听小程序显示  
  
    // 当小程序启动，或从后台进入前台显示，会触发 onShow  
  
  },  
  
  onHide: function () {  
    // onHide,生命周期函数--监听小程序隐藏  
  
    // 当小程序从前台进入后台，会触发 onHide  
  
  }  
})
```

二、page 生命周期

```
Page({  
  data:{  
    //页面的初始数据  
  },  
  onLoad:function(options){  
    // 生命周期函数--监听页面加载  
    // 页面初始化 options 为页面跳转所带来的参数  
    // 一个页面只会调用一次。  
  },  
  onReady:function(){  
    // 生命周期函数--监听页面初次渲染完成  
    // 页面渲染完成  
  },  
  onShow:function(){  
    // 生命周期函数--监听页面显示
```

```
    // 页面显示,每次打开页面都会调用一次。

},

onHide:function(){

    // 生命周期函数--监听页面隐藏

    // 页面隐藏,当 navigateTo 或底部 tab 切换时调用。

},

onUnload:function(){

    // 生命周期函数--监听页面卸载

    // 页面关闭,当 redirectTo 或 navigateBack 的时候调用。

}

})
```

三、页面跳转

- 1.wx.navigateTo , 压栈式跳转 , 跳转后首页隐藏 onHide , 自带返回
- 2.wx.redirectTo , 跳转后 首页关闭 onUnload , 不带返回
- 3.tabBar 跳转 , 页面相互切换 onShow 与 onHide
- 4.页面传值 : 跳转链接 url 后添加参数 url: '../logs/logs?id=1' 表示传 id=1 的参数

补充： 跳转方法可以通过 bindtap= “click”绑定点击事件来触发，

也可以直接写在 wxml 中

```
<navigator url="../../logs/logs?id=23&title=标题>
```

也可以添加 redirect 执行 redirect 跳转

```
<navigator url="../../logs/logs?id=23&title=标题" redirect>
```

四、布局基础

1. flex 容器属性

flex-direction： 决定元素的排列方向,同时决定主轴与交叉轴的方向

 `row`：横向排列，`column`：纵向排列

flex-wrap： 决定元素如何换行（排列不下时）

 `nowrap`：不换行，`wrap`：自动换行，`reverse`：反转换行

flex-flow： flex-direction 和 flex-wrap 的简写

例如：`flex-flow:row wrap;`

justify-content： 元素在主轴上的对齐方式

 `center`：在主轴上居中对齐，

 `flex-start(flex-end)`：从左(右)开始对齐

 `space-around`：每个元素的 padding 相同

``space-between`` : 每个元素间隔相同

`align-items` : 元素在交叉轴的对齐方式

``flex-start(flex-end)`` : 从上(下)开始对齐

``baseline`` : 根据元素内文本对齐

2. flex 元素属性

3. ``flex-grow`` : 当有多余空间时, 元素的放大比例

4. ``flex-shrink`` : 当空间不足时, 元素的缩小比例

5. ``flex-basis`` : 元素在主轴上占据的空间

6. ``flex`` : 是 `grow`、`shrink`、`basis` 的简写

7. 例如 : ``flex: 0 1 50px;``

8. ``order`` 定义元素的排列顺序, 默认为 0

``align-self`` 定义元素自身的对齐方式

9. 相对定位和绝对定位

相对定位的元素是相对自身进行定位, 参照物是自己。

例如 :

`position: relative;` `// 相对定位`

left: 150rpx; // 与原位置的左端距离 150

top: 50rpx; // 与原位置的上端距离 50

绝对定位的元素是相对离它最近的一个已定位的父级元素进行定位。

例如：

position: absolute; // 绝对定位

left: 50rpx; // 与已定位的父级元素左端距离 50

top: 50rpx; // 与已定位的父级元素上端端距离 50

五、样式基础

1. 尺寸

width: 228rpx; // 宽度

height: 228rpx; // 高度

min-width: //最小宽度

max-width: //最大宽度

min-height: //最小高度

max-height: //最大高度

...

2. 背景

background-color: darkcyan; //背景颜色

...

3. 边框

border-radius: 20%; //边框圆角

border-width: 5px; //边框宽度

border-color: #ddd; //边框颜色

border-style: solid; //边框样式

...

4. 边距

margin: 20px; //外边距，边框距离父级元素的距离

margin: 10px 20px 30px 40px; // 上右下左

padding: 20px; //内边距，边框距离元素的距离

padding: 10px 20px 30px 40px; // 上右下左

5. 文本

font-size: 30px; //字体大小

6. 其它(列表、内容、表格...)

PS：样式选择器 与 CSS 选择器相同，部分不可用