**1.小程序中常用的标签**

小程序中的常用标签也是就是我们常用的这些

view（视图容器）、rich-text（富文本）、swiper（滑块视图容器）、icon（图标）、text（文字）、progress（进度条）、button（按钮）、form（表单）、input（输入框）、checkbox（多项选择器）、radio（单项选择器）、picker（列表选择器）、slider（滚动选择器）、switch（开关选择器）、textarea（多行输入框）、label（标签）、navigator（应用链接）、audio（音频）、image（图片）、video（视频）、camera（系统相机）、map（地图）、scroll-view（可滚动视图容器）、picker-view（内嵌列表选择器）、canvas（画布）movable-area（可移动区域）、movable-view（可移动的视图容器）、cover-view（覆盖视图）、cover-image（覆盖图片）、functional-page-navigator（跳转到插件功能页）、live-player（实时音视频播放）、live-pusher（实时音视频录制）



1、view  
div和view都是盒模型，默认display:block。  
盒模型在布局过程中，一般推荐display:flex的写法，配合justify-content:center;align-items:center;的定义实现盒模型在横向和纵向的居中。  
2、text  
除了text文本节点以外的其他节点都无法长按选中。。  
截止到0.10.102800的开发者工具text支持嵌套text，不过有class的text会被面板过滤，样式不影响。  
3、 icon  
icon可以直接用微信组件默认的图标，默认是iconfont格式的，从WeUI那边沿袭过来的一种做法。  
自定义的icon推荐svg sprite格式或者iconfont。  
目前来看，市面上还没有很好的自动合并单个svg为svg sprite的工具，需要手动拼图。  
4、input  
input 的类型，有效值：text, number, idcard, digit, time, date 。  
input不需要设置line-height或padding来纵向居中，默认placeholder的文字是居中的。  
小程序把checkbox和radio都单独做成了组件，默认的input只支持输入文本。  
上传文件在小程序里需要调用chooseImage事件完成;  
5、picker  
picker默认支持普通、日期和时间三种选择器。  
picker通过bindchange事件来调取range中自定义的数据数据，并展示到页面中，调用的是系统原生的select。  
这里小程序废弃了select组件，考虑到的是这个组件的交互不适合移动场景，最终用picker代替了。  
选择 北京 上海  
6、 navigator  
navigator支持相对路径和绝对路径的跳转，默认是打开新页面，当前页面打开需要加redirect;  
navigator仅支持5级页面的跳转;  
navigator不可跳转到小程序外的链接地址;  
在小程序开发工具里，默认打开新页面，工具左上角有返回按钮。加上redirect，当前页打开，不出现返回按钮。  
7、image  
小程序的image与HTML5的img最大的区别在于：小程序的image是按照background-image来实现的。  
默认image的高宽是320\*240。必须通过样式定义去覆盖这个默认高宽，auto在这里不生效。(开发者说这样设置的原因是：如果设置 auto ，页面布局会因为图片加载的过程有一个闪的现象(例如高度从 0 到 height )，所以要求一定要设置一个宽度和高度。)  
最新的api支持获取图片的高宽。不过这里返回的高宽是px单位，不支持屏幕自适应;  
图片包括三种缩放模式scaleToFill、aspectFit、aspectFill和9种裁剪模式，三种缩放模式的实现原理对应如下：  
  
background-size:100% 100%;//不保持纵横比缩放图片，使图片的宽高完全拉伸至填满 image 元素  
  
background-size:contain;//保持纵横比缩放图片，使图片的长边能完全显示出来。也就是说，可以完整地将图片显示出来。  
  
background-size:cover;//保持纵横比缩放图片，只保证图片的短边能完全显示出来。也就是说，图片通常只在水平或垂直方向是完整的，另一个方向将会发生截取。  
8、button  
额外补充下button的实现方式，button的边框是用:after方式实现的，用户如果在button上定义边框会出现两条线，需用:after的方式去覆盖默认值。不过这个应该会在最近的开发者工具中修复。  
  
小程序不支持button:active这种样式的写法，button的点击态通过.button-hover{}的样式覆盖，也可修改hover-class为自定义的样式名。  
9、css3动画  
最新版的开发工具已经支持transition和keyframes动画，不用js苦哈哈的写动画队列了。  
除了官方公布的小程序的组件之外，有一些标签比如，span、em、strong、b

也是支持的，只是官方并不推荐使用。

# 2.微信小程序图片上传组件

<https://www.jianshu.com/p/01eac095cf95>

# 3. [微信小程序页面路由](https://www.cnblogs.com/home-/p/11607040.html)

<https://www.cnblogs.com/home-/p/11607040.html>

标签跳转<navigator></navigator>:组件-->导航-->navigator

navigator标签中的属性:(不支持跳转tabbar页面需要设置跳转方式)

url:当前小程序内的跳转链接

open-type:跳转方式(有很多合法值)switchTab只支持跳转tabbar页面

<navigator url='/pages/index/index' open-type='switchTab'>跳转</navigator>

open-type中的合法值navigate只支持跳转非tabbar页面

<navigator url='/pages/list/list' open-type='navigate'>跳转</navigator>

open-type中的合法值reLaunch可以跳转任何页面

wx.navigateTo({

url: 'test?id=1',

success: function(res) {

// 通过eventChannel向被打开页面传送数据

res.eventChannel.emit('acceptDataFromOpenerPage', { data: 'test' })

}

})

wx.navigateTo({

url: 'test?id=1',

events: {

// 为指定事件添加一个监听器，获取被打开页面传送到当前页面的数据

acceptDataFromOpenedPage: function(data) {

console.log(data)

},

someEvent: function(data) {

console.log(data)

}

...

},

success: function(res) {

// 通过eventChannel向被打开页面传送数据

res.eventChannel.emit('acceptDataFromOpenerPage', { data: 'test' })

}

})

# 4.选项卡

HTML中

<view class="menu\_box">

    <view class='menu1 {{menuTapCurrent=="0"?"borders":""}}' data-current="0" catchtap="menuTap">托运方</view>

    <view class='menu2 {{menuTapCurrent=="1"?"borders":""}}' data-current="1" catchtap="menuTap">承运方</view>

 </view>

 <view class="tab1" hidden="{{menuTapCurrent!='0'}}">

    11111

 </view>

 <view class="tab1" hidden="{{menuTapCurrent!='1'}}">

    22222

 </view>

Js中

 data: {

    menuTapCurrent:0,

  },

  menuTap: function (e) {

    var current = e.currentTarget.dataset.current;//获取到绑定的数据

    //改变menuTapCurrent的值为当前选中的menu所绑定的数据

    this.setData({

      menuTapCurrent: current

    });

  },

Css中

.menu\_box{

 display: flex;

 height: 80rpx;

 position: absolute;

 top: 0;

}

.menu1,.menu2,.menu3,.menu4,.menu5{

 flex: 1;

 font-size:30rpx;

 line-height: 80rpx;

 text-align: center;

 display: block;

 width: 130rpx;

}

.borders{

 border-bottom: 4rpx solid #f00;

 color: #f00;

}

.tab1,.tab2,.tab3,.tab4,.tab5{

  position: absolute;

  top: 90rpx;

 height: 300rpx;

 width: 100%;

 background: #23bff3;

 font-size: 28rpx;

 overflow: hidden;

}

.tab2{

 background: #ccc;

}

.tab3{

 background: rgb(62, 216, 144);

}

# 5.页面配置

{

  "navigationBarBackgroundColor": "#ffffff",

  "navigationBarTextStyle": "black",

  "navigationBarTitleText": "微信接口功能演示",

  "backgroundColor": "#eeeeee",

  "backgroundTextStyle": "light"

}

# 6. 微信小程序之picker组件（选择器）

<https://blog.csdn.net/xiangzhihong8/article/details/78106016>

**html中**

<picker class="goodsinfo" bindchange="changetype" value="{{index}}" range="{{typearr}}">

    <text>货物类型</text>

    <text style="text-align:right;width: 500rpx;display: inline-block;">{{typearr[index]}}</text>

    <text class="iconfont icon-jiantou jiantou"></text>

  </picker>

**Js中**

data: {

    typearr: ['请选择反馈类型', '商品相关', '物流状况', '客户服务', '优惠活动', '功能异常', '产品建议', '其他'],

    index: 0,

  },

  changetype:function (e) {

    console.log('picker发送选择改变，携带值为', e.detail.value)

    this.setData({

      index: e.detail.value

    })

  },

# 7.获取节点

<https://blog.csdn.net/weixin_34248849/article/details/88069879>

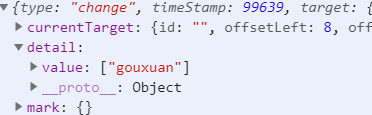
# 8.获取checkbox是否勾选

<checkbox-group bindchange="checkedvalue">

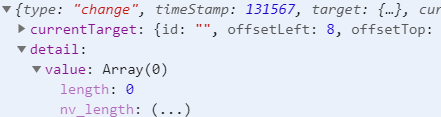
      <checkbox value="gouxuan"></checkbox>

    </checkbox-group>

如勾选：



如未勾选：



以此判断是否勾选，勾选的是那个。

# 9.向e.target.dataset传数据

<https://www.cnblogs.com/dawenyang/p/9039495.html>

用data-\*属性 传输数据

# 10.wx.for的使用

<https://www.jianshu.com/p/ee6b297eb6c3>

<view wx:for="{{arr}}" wx:key="index"></view>

# 11.wx.if的使用

wx:if="{{item.state==1}}"

# 11.input中的值

Bindinput事件

键盘输入时触发，event.detail = {value, cursor, keyCode}，keyCode 为键值，2.1.0 起支持，处理函数可以直接 return 一个字符串，将替换输入框的内容。

# 12.组件使用方法

<https://www.jianshu.com/p/a6c456d68f42>

组件component的写法

<https://www.jianshu.com/p/3aec33a54d8d>

# 13.小程序和组件之间数据传输

<https://blog.csdn.net/qq_33237207/article/details/84235443>

# 14.从本地上传图片

1.点击事件调用上传图片组件

2.设置data：{img:””}

3.this.setData{

Img:res.

}

Eg:

  unload:function(){

    var that = this;

    wx.chooseImage({

      count: 1,

      success:function(res){

        console.log(res)

        let tempimgurl= res.tempFilePaths;

        that.setData({

          imgurl:tempimgurl,

        })

      }

    })

    console.log(this.data.imgurl)

  },

上传图片展示：用

<image class="img" src="{{imgurl}}"></image>