移动端swiper scroll简单封装

主要原理：

手指移动时的位置-—手指按下的位置，加上元素按下的位置，就是元素移动过程中的位置

分析需求：

1.取消document默认事件，阻止点透事件，及其他

2.一个函数一个功能，将滑屏的行为，和滑屏之后的操作分开，

3.首先封装滑屏swiper行为

参数初始化，合并配置，初始手指、移动距离、两次移动距离

手指按下，记录手指位置，和移动初始值——执行手指按下的函数

手指移动，当前手指位置-手指按下手指的位置，得到手指移动的距离——将手指移动的距离传入函数，执行手指移动的函数

手指抬起，——执行手指抬起的函数

4.swiper函数传入参数，

传入元素，不同手指状态不同的函数

5.封装滑屏scroll操作——横屏滑动、竖屏滑动

不同的方向滑屏判断

x、y、xy

6.滑动的速度

记录手指移动过程中两次手指移动的距离

手指抬起，将两次手指移动距离值当做参数传入手指抬起的函数中

手指抬起的函数中执行以下的内容

运动的t、b、c、d

t变化的时间

b开始的位置

c总的位置

d总的时间

注意当t>d的时候结束运动

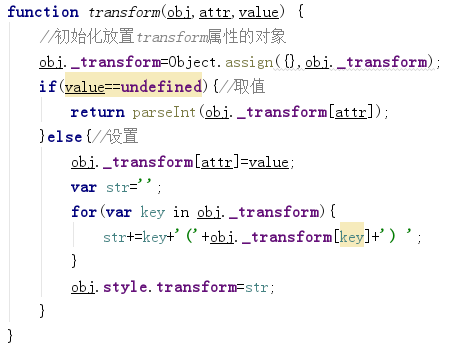
运动的函数，计算出位置

7.手指按下函数，停止运动

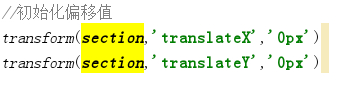
*1.取消document默认事件*



*2.transform封装*



3. 初始化transform值



*4.封装swiper行为*

*将行为和操作分开，swiper就封装行为*

**function** *swiper*(option) {

*//合并配置，初始化参数，参数当大于3个的时候，用对象传入*

**var** opt=Object.assign({

**elm**:**document**,

onStart:**function** () {***console***.log(**'开始按下'**)},

onMove:**function** () {***console***.log(**'开始移动'**)},

onEnd:**function** () {***console***.log(**'结束'**)}

},option)

*//手指的位置*

**var** option={

**x**:0,

**y**:0

};

*//手指移动的距离*

**var** dis={

**disX**:0,

**disY**:0

};

*/\*按下，记录手指的位置和移动的位置\*/*

opt.**elm**.addEventListener(**'touchstart'**,**function** (e) {

*//手指按下记录当前手指的位置*

option={

**x**:e.changedTouches[0].**pageX**,

**y**:e.changedTouches[0].**pageY**

};

opt.onStart();*//按下手指的回调函数*

})

*/\*移动，手指移动的距离\*/*

opt.**elm**.addEventListener(**'touchmove'**,**function** (e) {

*//手指移动的距离*

dis={

**x**:e.changedTouches[0].**pageX**-option.**x**,

**y**:e.changedTouches[0].**pageY**-option.**y**

};

*//将手指移动的距离当做参数传到回调函数中*

opt.onMove(dis);*//移动手指的回调函数，并将dis当做参数传入回调函数*

})

*/\*手指抬起\*/*

opt.**elm**.addEventListener(**'touchend'**,**function** () {

opt.onEnd()*//手指抬起时，触发的函数*

})

}