首先可以试着把架构写完 写样式，再看看有什么可以随机改变的量

<style type="text/css">

\*{ margin: 0; padding: 0; }

/\*系统滚动条禁止\*/

html,body{height: 100%;overflow: hidden; }

#index{height: 100%;overflow: hidden; }

#carouse-wrapper ul{position: absolute; list-style: none; overflow: hidden;

/\*width: 500%;\*/}

#carouse-wrapper ul > li{float: left; /\*width: 20%;\*/}

#carouse-wrapper ul > li img{display: block; width: 100%;}

</style>

</head>

<body>

<div id="index">

<div id="carouse-wrapper"></div>

</div>

</body><body>

<div id="index">

<div id="carouse-wrapper"></div>

</div>

</body>

}

//首先先给页面禁止默认行为

<script type="text/javascript">

document.addEventListener('touchstart',function(ev){

ev = ev||event;

ev.preventDefault();

})

/\*

首先整个结构都要先获取

1.首先把所有的图片放在一个数组中,就可以动态生成结构

2.然后获取到这个id 先生成相应的html ul 然后定义一个liText的列表 循环的给它加一li标签,

添加了之后就把这个元素给ul的innerHTNL,在整个ul给这个id

然后给这些图片加样式,设置总的ul的长度,一个li的长度,给这个id容器(获取图片的高度赋值给ul的高度)

\*/

window.onload = function(){

var urls=["img/01.jpg","img/02.jpg","img/03.jpg","img/04.jpg","img/05.jpg",]

var wrap = document.querySelector('#carouse-wrapper');

var list =document.createElement('ul')

var liText = '';

for(var i=0;i<urls.length;i++){

liText+='<li><a href="javascript:;"><img src="'+urls[i]+'" /></a></li>';

}

list.innerHTML = liText;

wrap.appendChild(list);

var imgs = document.querySelectorAll('#carouse-wrapper ul li a img');

setTimeout(function(){

var styleNode = document.createElement('style')

styleNode.innerHTML+="#carouse-wrapper ul{width:"+urls.length+"00%!important}";

styleNode.innerHTML+="#carouse-wrapper ul>li{width:"+(1/urls.length\*100)+"%!important}";

styleNode.innerHTML+="#carouse-wrapper {height:"+imgs[0].offsetHeight+"px!important}";

document.head.appendChild(styleNode)

},100)

/\*滑屏事件\*/

/\*

首先获得第一根手指touchC ，然后定义startX多去当前的手指的位置作为初始位置，

然后定义一个elementX记录元素当前的ul的偏移量

\*/

//记录手指的起始位置

var startX = 0;

//记录当前元素的list的位置

var elementX = 0;

//记录移动时的手指到达的位置

var nowX = 0

//抽象图标的下标

var now =0;

//每次move时手指移动的距离

var disX =0;

wrap.addEventListener('touchstart',function(ev){

ev=ev||event;

var touchC = ev.changedTouches[0];

startX = touchC.clientX;

elementX = list.offsetLeft;

})

/\*

在手指移动的时候 要重新获得手指，当前的now的位置就是手指一动的位置，移动的位置=nowX-startX就是手指

后list就一定disX的位置

\*/

wrap.addEventListener('touchmove',function(ev){

ev=ev||event;

var touchC = ev.changedTouches[0];

//手指移动时的实时位置

var nowX = touchC.clientX;

//每次move时手指移动的距离

disX = nowX - startX;

list.style.left=(elementX+disX)+"px";

})

/\*

根据上面touchmove就知道disX是往那边进行一定 进入判断 如果正数就是右边 负数就是左边 当手指离开的时候就就根据判断让list移动一个

一张图片（屏幕宽度的位置）

\*/

wrap.addEventListener('touchend',function(ev){

ev=ev||event;

var touchC = ev.changedTouches[0];

var now=list.offsetLeft/document.documentElement.clientWidth

now=Math.round(now);

list.style.left = (now\*document.documentElement.clientWidth)+'px'

})

}