手机端事件

目录：

1. 手机端事件

touchstart

touchmove

touchend

touchcancel

封装tap事件，类似pc的click事件，注意this的指向问题

注意：touchend和touchcancel的区别，touch事件和鼠标事件的执行顺序

2. 点透事件，及解决，解决之后存在的问题

3. 事件对象下的3个属性

e.changedTouches——注意不同事件下面情况

e.targetTouches

e.touches

4. transform的获取与设置，封装

5. swiper简单封装

1. 手机端事件

是一类描述手指在触摸平面（触摸屏、触摸板等）的状态变化的事件。

**1.1 输入url之后发生的事情**

输入<https://www.apachefriends.org/zh_cn/index.html>

通过：https://www.apachefriends.org/ 这个部分可以定位某个具体的主机（电脑，服务器）；

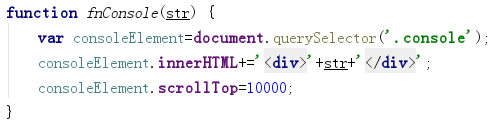
这个主机上的某个软件就会捕获这次请求，分析url，根据url的规则去读取主机上的某个具体的文件内容，或返回某个具体的文件内容。

这种专门用来处理web浏览的软件，web 服务器软件：

比较有名的一些web服务器软件：

apache、nginx、tomcat、node、iis

**1.2 手机模拟console.log**



**1.3 touch 事件列表**

**touchstart** 手指按下——要在元素身上

**touchmove**手指移动——必须有start才有move；触发了start之后，就会move，即时**不在元素**身上，都会触发

**touchend**手指抬起——主动取消，up事件也不用在元素身上触发

**touchcancel**取消手指——被动取消，比如手机离开元素了，比如弹窗，

即时手机没有离开手机，就取消了

注意：touchend和touchcancel的区别，touchend为主动取消，cancel为被动取消。两者只会执行一个。

注意：touch事件在有些浏览器的模拟器上使用on绑定的方式有兼容问题，但是真机是没有问题的，推荐使用addEventListener。

注意：touch事件和鼠标事件的执行顺序：touchstart-touchend-mousedown-mouseup-click

**1.4 封装tap事件**

类似pc的click事件，注意this的指向问题。

需求分析：节点点下，不移动，抬起，这个过程就是点击。



2. 点透事件，及解决，解决之后存在的问题

2.1 点透事件

touch事件和鼠标事件的执行顺序，默认情况下touch和mouse事件会同时触发；mouse事件是在touch事件之后，移动端处理mouse事件的时候，会有延迟。

例子：一个节点有touch事件和mouse事件，这个节点下方有其他节点。当手指点击这个节点时，这个节点触发了touch事件就消失，但是mouse事件会延迟300ms，当300ms到的时候，设备会执行mouse事件，会找发生mouse事件手指的位置的节点，这个节点是消失节点下方的节点，不是节点本身。这个就是点透事件。

2.2 点透事件的解决
移动端一般使用touch，而不去使用mouse事件，mouse事件没有与手指相关的数据。

解决：

因为touch事件触发以后会在300ms以后执行mouse，mouse事件是touch的后续默认动作，所以我们可以在touch事件发生以后，通过阻止默认行为，来阻止后续的mouse事件。



2.3 阻止事件默认行为

2.3.1 好处

阻止事件点透；

阻止图片或文字被选中，导致拖拽卡顿；

解决ios10之后，ios下的safari不在支持viewport的禁止缩放；

解决ios10之后, body的overflow无效的问题。

2.3.2 产生的影响

a标签不能跳转——在touchstart上阻止冒泡

输入框不能输入——在touchstart上阻止冒泡

移动端所有鼠标事件无效（手指长按无法选中文字菜单）

滚动条不能滑动——自定义滚动条，移动端滑屏组建swiper4（模拟滑屏，start，move，end，js来控制）

页面不能回弹——控制left top，tanslate

系统菜单无效

页面不能缩放

3. 事件对象下的3个属性

e.changedTouches——发生改变的触摸点，注意不同事件下面情况

touchstart——新增的手指数

touchmove——移动的手指数

touchend——从屏幕离开的手指数

e.targetTouches——在目标元素的触摸点

e.touches——当前屏幕上的触摸点

注意：3个属性都是列表形式的

this的指向--widow————fn.call()?什么问题

clipboard.png

4. transform的获取与设置，封装

如果我们的js需要去控制（获取）一个元素的transform值，那么最好通过js的方式来操作，当然在js中也不要直接去设置这个元素的tansform，而是封装一个函数，通过这个函数来操作，，因为这样我们可以通过这个函数来记录操作的过程。

5. swiper简单封装