## 一、Ejs

什么是模板引擎，为什么要用模板引擎

* 循环遍历数据显示
* 实现数据与视图分离（即html结构与数据分离)

## 二、HTML

<em> 表示强调，默认用斜体表示

<strong> 表示更强烈的强调， 用粗体表示

<code>/<pre>加入代码标签

<caption>为表格添加标题摘要

<a>标签属性target

mailto



<img>

1、src：标识图像的位置；

2、alt：指定图像的描述性文本，当图像不可见时（下载不成功时），可看到该属性指定的文本；

3、title：提供在图像可见时对图像的描述(鼠标滑过图片时显示的文本)；

4、图像可以是GIF，PNG，JPEG格式的图像文件。

<form>表单属性

1、action ：浏览者输入的数据被传送到的地方,比如一个PHP页(save.php)。

2、method ： 数据传送的方式（get/post）

3、所有表单控件（文本框、文本域、按钮、单选框、复选框等）都必须放在 <form></form> 标签之间（否则用户输入的信息可提交不到服务器上）

<input>属性type="radio" 时，控件为单选框

"checkbox" 时，控件为复选框

value：提交数据到服务器的值（后台程序PHP使用）

name：为控件命名，以备后台程序 ASP、PHP 使用

checked：当设置 checked="checked" 时，该选项被默认选

<select>属性：value：提交值

multiple="multiple" 多选

<option> selected="selected" 默认选中

<input>属性：type：只有当type值设置为submit时，按钮才有提交作用

只有当type值设置为reset时，按钮才有重置作用

value：按钮上显示的文字

<label>当用户单击选中该label标签时，浏览器就会自动将焦点转到和标签相关的表单控件上

## 三、CSS

子选择器:即大于符号(>),用于选择指定标签元素的第一代子元素.

包含选择器:即加入空格,用于选择指定标签元素下的后辈元素。

常用的块状元素有：

<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>

常用的内联元素有：

<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>

常用的内联块状元素有：

<img>、<input>

2d/3d转换

<transform>

rotate()旋转角度

translate()根据x，y轴移动

scale()x，y宽度放大倍数

skew() x,y倾斜角度

transition：width 1s 过渡时间

动画，当动画完成时，会变回初始的样式。

animation：动画名称 time

@keyframes 动画名称{

from：{}

to：{}

// 也可0-100%

}

grid布局

## 五、前端性能

1. 减少http请求（第一次访问有效减少HTTP请求）
2. 图片地图：超链接关联到图片上，一个图片上可以关联多个URL，减少HTTP请求。但是图片区域不好划分。
3. CSS Script：多个图片合并到一个单独的图片上，合并后的图片比分离图片的总和小。Data，用于内联图片，也可用在需要指定URL的地方。
4. 合并脚本和样式表：每个文件都要一个HTTP请求。
5. 使用内容发布网络（CDN）

应用程序web服务器离用户更近，一个HTTP请求的响应时间会缩短；组件web服务器离用户更近，多个HTTP请求响应时间会缩短。依赖CDN，响应时间可能会受其他网站流量的影响，无法直接控制组件服务器，如，修改HTTP响应头必须通过服务器提供商完成。

CDN用于发布静态内容，更容易储存且有较少的依赖性。

1. 添加Expires头， （长久的）Expires头：使用缓存减少HTTP请求数量，并减小HTTP响应大小。
2. Cache-Control Max-Age和mod\_expires：缓存时间
3. 空缓存VS完整缓存
4. 长久的Expires应不仅是图片，应包含任何不经常变化的组件（脚本、样式表、flash组件）
5. 修订文件名，确保用户获取组件的最新版本
6. 压缩组件
7. 使用gzip编码压缩HTTP响应包，减少网络响应时间。
8. Accept-Encoding
9. Content-Encoding

将样式表放在顶部

脚本放在底部

避免CSS表达式

## 六、HTML5

### 新特性

HTML5 中的一些有趣的新特性：

* 用于绘画的 canvas 元素
* 用于媒介回放的 video 和 audio 元素
* 对本地离线存储的更好的支持
* 新的特殊内容元素，比如 article、footer、header、nav、section
* 新的表单控件，比如 calendar、date、time、email、url、search

新增元素20个

H5兼容：

IE：Shiv 解决方案

<!--if lt IE 9>

<script src=””></script>

<![endif]-->

### （一）媒介

视频：<vedio src=”” cotrol=”control”>

<sourse src=”” type=”vedio/ogg”>

</vedio>

方法paly() pause()

音频:<audio>

### （二）拖放（Drag 和 Drop）

1.拖动元素draggable 属性设置为 true

2.拖动时ondragstart 属性调用drag(event)，dataTransfer.setData() 方法设置被拖数据的数据类型和值

3放置元素ondragover事件，调用event.preventDefault() 方法设置允许放置，因为浏览器默认不允许

4.放置被拖数据时，ondrop 属性会调用 drop(event)

<canvas>

颜色，坐标，路径，文本，渐变，画像

Svg与convase比较

### （三）获取地理位置

watchPosition() - 返回用户的当前位置，并继续返回用户移动时的更新位置（就像汽车上的 GPS）。

clearWatch() - 停止 watchPosition() 方法

### （四）web存储

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储

应用缓存

HTML标签属性manifest属性，manifest 文件的建议的文件扩展名是：".appcache"

manifest 文件需要配置正确的 MIME-type，即 "text/cache-manifest"。必须在 web 服务器上进行配置

浏览器对缓存数据的容量限制可能不太一样（某些浏览器设置的限制是每个站点 5MB）

Web worker（IE不支持）

web worker 是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能。

web worker脚本postMessage() 方法 - 它用于向 HTML 页面传回一段消息

从 HTML 页面调用web worker脚本，创建新的 web worker 对象，运行脚本， "onmessage" 事件监听器。

终止web worker，terminate() 方法。

由于 web worker 位于外部文件中，它们无法访问下例 JavaScript 对象：window 对象、

document 对象、parent 对象

HTML5 服务器发送事件（server-sent event）允许网页获得来自服务器的更新。

### 表单

表单元素input的13个新类型：

color/data/datatime/datatime-local/time/range/week/number/email/tel/url/mounth/search

新增表单元素：datalist/keygon/output

### （六）缓存

cache manifest

#日期 版本号

CACHE MANIFEST: 第一次加载后缓存

NETWORK:\* 需要联网 ，\*表示其他所有文件

FALLBACK：/html/ /offline.html

如果无法建立因特网连接，则用 "offline.html" 替代 /html5/ 目录中的所有文件

清除缓存：用户自动清，manifest 文件被修改（更改#里面的额日期版本号），程序更新

### web worker

new Worker(文件url)

typeof(Worker)

terminate();终止

但是无法访问

window 对象

document 对象

parent 对象

postMessage（）

## 七、VUE



项目通过运行脚本npm run build来执行打包

main.js:项目入口文件，主要作用是初始化vue实例并引入需要的插件。其中App.vue是我们的主组件，所有页面都是在App.vue下进行切换的

axios服务调用:

是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。

功能：

从浏览器中创建 XMLHttpRequests

从 node.js 创建 http 请求

支持 Promise API

拦截请求和响应

转换请求数据和响应数据

取消请求

自动转换 JSON 数据

客户端支持防御 XSRF

#### 2018/10/17 基础知识

Nuxt，weex

基础知识：js、 ES6、 webpack、 npm：node.js包管理工具

1.Vue不兼容IE8及以下的浏览器

2.不对DOM操作，对数据，数据变了页面就变了 MVVM

MVVM模式，M层 V层 VM层--ES5 API，虚拟DOM

MVP模式：model：数据层；presenter：呈现层；view:视图层

3.todoList

标签指令：

v-for 循环

V-on 绑定事件(@)

V-model 数据双向绑定（适用于input）

4.前端组件化：页面的一个区域

Vue.component 创建全局组件的方法

组件名：大驼峰

props:：子组件接收父组件传到的值

template：组件的html

V-bind 属性值（:）

局部组件需要在new Vue实例里注册，components

#### 2018/10/18 父子组件传值

1.子组件向父组件传值

子组件绑定方法，把绑定的内容传给父组件，

$emit()向外触发事件

在绑定一个监听这个事件

在父组件里定义监听事件后的方法

1. 一个组件就是一个Vue实例，带$的是实力属性或者实例方法

$data 实例属性

$destroy()，销毁实例的方法

3.Vue实例生命周期钩子

4.模板语法：

v-text：与插值表达式作用相同{{}}

v-html：

{{}}和v-XX:=””,等于号后的引号内可以是js表达式

#### 2018/10/21 计算属性

1. 计算属性

减少数据冗余。

是内置缓存的，如果依赖的值没有改变，就不会重新计算一次

一种方式：计算属性 computed；另一种方式：方法methods

方法不如计算属性能高，因为没有缓存机制

watch监听方法也能实现计算属性的功能，也存在缓存机制，但是代码比较复杂。

1. 计算属性的setter和getter

get：被计算的值变了，计算属性的值也会变

set：计算属性的值变了，被计算的值也会变,改变关联的值

1. Vue中样式绑定

借助class的对象绑定

Class数组绑定-变量，可以有很多变量，class为变量

1. Vue中的条件渲染

v-if，true或false，是否在页面上存在

v-else，使用v-if和v-else的标签紧贴使用

v-show：true或false，是否在页面上展示

是否展示--v-show性能高一些

1. 给DOM元素标签加一个key值，跟别的元素key值不一样，就不会复用同样的内容
2. Vue中的列表渲染

vue里操作数组 pop push shift unshift splice sort reverse

Template只做包裹，页面上不会渲染

对象做循环

#### 2018/10/28 组件

1. 组件：

table、select、ul is=”组件名”

data只能是个函数，不能是个对象，在根组件只用一次，在子组件会多次调用，因此在子组件，data:function(){ return{} }

1. 在Vue中操作DOM ：标签上ref=””获取DOM节点

this.$refs

组件获取的是子组件的引用

2018/10/30

父组件通过属性的形式向子组件传值，传递参数，子组件不能反过来修改父组件传递过来的内容。因为可能影响其他的子组件

:count="0" 有冒号等号后面是一个js表达式，就是数字。没有冒号就是字符串

<div id="root">

<counter :count="0"></counter>

<counter :count="1"></counter>

</div>

<script>

//父组件通过属性的形式向子组件传值

var counter = {

template:'<div>0</div>'

}

var vm = new Vue({

el:'#root',

components:{

counter:counter

}

})

</script>

props：子组件接收父组件传递的数据

单向数据流：父组件可随意向子组件传递参数，但是子组件不能随意修改父组件传递过来的参数。

子组件像父组件传递数据：向外触发事件 $emit()

#### 2018/11/27组件参数校验

组件参数校验与非props特性：

子组件对父组件传递的数据进行校验（比如：类型校验，自定义校验等）

非props特性：没有声明获取的内容就不能使用；属性会展示在子组件最外层的标签属性里

props特性：父组件通过属性向子组件传值，子组件中props声明对父组件传值的接收

特点：

2018/11/28

在组件上绑定的是自定义事件，在元素上绑定的是原生事件

想要在组件直接绑定原生事件，@click.native='handleChildClick'

非父子组件传值：第三层组件与第一层数据传递，第三层之间传值

1. Vue官方网站提供的数据层框架 Vuex ）
2. Bus/总线机制/发布订阅模式/观察者模式

Vue.prototype.bus = new Vue()

#### 2019/1/27 插槽

插槽：使用场景 父组件向子组件传递DOM结构

<slot> 标签放插槽内容，可以直接在标签内写默认值

作用域插槽：当子组件循环，某一部分DOM结构由外部传递，父组件接收子组件传递的数据， 要用<template>标签包裹起来

动态组件：根据:is数据的变化动态加载组件

v-once指令（性能好）

#### 2019/3/10 Vue的CSS原理

过渡的动画效果：通过某一时刻增加和删除class (transition)

在标签<transition>加属性name=”fade” (.fade-enter .fade-enter-activ .fade-leave-to

.fade-leave-active) 没有name 默认为v-XXX

keyframe

自定义class名称 enter-active-class=””

可引入animate.css

动画执行时长是以过渡效果还是keyframes，type=“transition”;自定义 :duration=””1000,可设置入场动画时长和出场动画时长

2019/3/11 动画钩子 velocity.js动画库

入场动画 before-enter动画执行前，enter在before-enter执行完执行,done执行完执行after-enter

出场动画

vue会复用DOM，不复用加key值

transition属性mode="in-out"，多个元素先进入后隐藏 mode="out-in"先隐藏，后进入

列表过渡<transition-group>

#### 2019/4/11 VUE中的动画封装

#### 2019/4/15 VUE创建工程代码上传GitHub

动态过渡

1. 安装node.js
2. 打开cmd，输入:node -v 出现版本号
3. npm -v 包管理工具
4. 码云网站 ，注册账号，代码管理，创建项目，通过Git关联
5. 本地安装Git，下载安装
6. Git安装完成打开cmd，输入git --version 看安装好没
7. 将本地代码与git关联 生成ssh公钥
8. 桌面右键可以看见git bash，打开可以执行Linux命令
9. ssh-keygen -t rsa -C “邮箱”
10. 再执行 一句cat ~/.ssh/id\_rsa.pub 生成公钥
11. 线上创建好的项目下载到本地（选ssh协议）,复制地址
12. 在执行Linux指令的窗口，先找到要放代码的文件夹然后执行git clone 复制的地址
13. 在文件夹里创建Vue的项目，先安装Vue的CLI脚本架
14. Cmd里运行npm install --global vue-cli
15. 切换到文件夹，vue webpack my -project（文件夹名称）
16. 编译形式选第一个，路由选Y，。。。。各种选完之后，等着下载完
17. npm run dev 出现地址
18. 浏览器里输入地址
19. 打开码云，本地上传到线上 Linux里git status
20. git add .
21. git commit -m ‘project initialize’
22. git push

文档

webpack前端打包工具

<https://github.com/linsem123/demo.git>

http://localhost:8080

git add -A && git commit -m "update" && git pull && git push

git status



#### 2019/4/16 单文件组件和路由

路由：根据网址的不同，返回不同的内容给用户

<router-view/> 显示的是当前路由地址所对应的内容

文件路径@/的@表示src文件夹

#### 2019/4/17 多页面和单页面应用

多页面应用：每次页面跳转都是一个新的html，首屏时间快，SEO效果好，会给html一个好的排名，页面切换慢

单页应用：Vue是单页面应用，原理是js感知url变化，把原来的清除掉，渲染新的页面，优势是页面切换快，首屏时间慢SEO查搜索引擎优化差，搜索引擎只能识别html不识别js。

页面跳转<router-link>

在template暴露在外面的只能一个元素，包裹其他元素

ctrl+shift+` 打开vscode终端控制台

移动端网页-项目初始化

Reset.css 保证不同终端浏览器初始样式一致

项目入口文件main.js

build文件夹放的是webpack相关的

config生产和开发环境的配置参数

node

移动端1像素边框，分辨率比较大，多倍像素

某些机型300ms点击延迟，解决这种亲情况引入fastclick

Iconfont

#### 2019/4/18 首页开发

npm install stylus --save 安装stylus样式依赖包

手机端：meta标签添加不可以页面缩放，reset样式统一，border.css一像素边框问题，手机分辨率，fast-click 解决click事件延迟执行

局部化样式：对style样式做限制只用于当前组件有效 <style>加 scoped

Stylus iconfont

代码优化：

1. 背景颜色，写成变量 ：$+变量名，@import引用文件 ~@
2. 改了webpack的文件内容需要重启服务器

GitHub创建分支：

1. git branch -a(查看所有分支包括本地分支和远程分支)

     git branch -r(查看远程分支)

1. git checkout -b branchname(直接新建一个分支然后切换至新创建的分支).就是创建加切换分支
2. git checkout branchname(是切换分支名)
3. 将新分支推送至GitHub，git push origin branchname

分支怎么上传 合并 ？

先git checkout master，

Git merge branchname

Git push

首页轮播图 vue-awesome-swiper插件

Google安装vue.devtools插件有助于代码编译

占位，宽高比

height:0

overflow:hidden

padding-bottom :50%

forEach()遍历数组

超出文字...展示

overflow: hidden

white-space: nowrap

text-overflow: ellipsis

#### 2019/5/9 ajax

Axios

不想上传的文件：把文件路径加在gitignore

接口路径：本地的模拟报文跟实际上的不一样, api替换成本地路径

Config/index.js 文件proxyTable ，接口联调更改proxyTable下target的路径

父子组件数据传递：属性props

<router-link to=”URL”> 路由 页面跳转内置组件

宽高：content+padding

box-sizing:border-box

Class “border-topbottom” 上下边框，在移动端处理1px

.border-topbottom

&:before

border-color:#ccc

&:after

border-color:#ccc

better-scroll 是iscroll的封装

Vue 通过ref属性获取dom，变化的:ref

Watch 监听器

兄弟组件传值：其中一个子组件向外触发$emit()传值，父组件监听这个组件接收，再通过props传给另一个子组件，该子组件通过监听值的变化触发事件

打包上线：

Npm run build

Dist 文件是上线内容

App.js文件很大的时候使用异步组件优化代码

在app.vue文件 去掉import，components：（）=>import(‘url’)

其他组件里面也可以这样

#### 2019/5/12 Vuex数据共享

数据层框架

把数据放在公共的存储区，如果一个组件改变数据，其他组件也能感知到

State表示公共数据，组件想要改变得调用Actions来进行数据处理和批量的同步操作->调用Mutations来改变公共数据，

可以越过Actions但是注意使用commit方法

getters工具的作用类似于computed ，用于将state数据计算

Module工具代码按模块拆分，拆分出来的包括state，actions，mutations，用module整合

编程式导航

router，页面跳转，一种是结束形式编程式导航，另一种是router-link 生成a标签

Router.push(页面URL)

this.router.push()

Localstorage，使用的时候try{}catch(e){},避免有些浏览器会抛出异常

keep-alive 优化网页性能 新增 activited生命钩子，页面重新显示的时候

路由切换的时候都会发起页面ajax请求，页面会重新渲染

2019/5/14 对全局事件解绑 ，容易出bug

document.documentElement 等同于document.html

Window.addEventListener 事件监听

Created：创建Vue对象时，HTML的dom渲染之前

使用keep-alive才有的钩子

Activated 页面初始化

Deactivated 离开当前页面

使用keep-alive ，Mounted第二次进页面不会被执行因为页面被缓存

递归组件：自己调用自己的组件

打包：npm run build

生成dist文件，里面的内容就是需要的

不在后台根目录下，config -- index.js的build里assetsPublicPath的路径

异步组件

针对App.js，在App.js把所有页面都加载出来，异步组件就是用到哪个页面加载哪一个页面的代码

1.把import引入组件代码去掉换个位置写， component：()=>import(组件路径)

#### 2019/5/16 文档学习

#### 2019/6/17 练习笔记

动态路由 ：<router-link :to=””></router-link> to作为动态的

路由配置 ：path：’/detail/:id’ 表示路径必须是detail/+一个参数 ：id是动态的

## 八、DIV+CSS

### [网页标准化布局(一）](https://www.cnblogs.com/ckyo/p/5871446.html)

#### 优势

表现和内容相分离；代码简洁，提高页面浏览速度；易于维护和改版；提高搜索引擎对网页的索引效率

<span>和<div>标签是没有语义的，是应用样式，所以当样式表失效时它就没有任何的作用

#### 盒模型层次

第1层：盒子的边框(border)  
     第2层：元素的内容(content)、内边距(padding)  
     第3层：背景图(background-image)  
     第4层：背景色(background-color)  
     第5层：盒子的外边距(margin)

#### 水平居中和垂直居中

#### 定位

绝对定位：absolute 相对于最近的具有定位属性的父包含块，若没有则对body

相对定位：relative 相对于以前的位置移动（偏移前的位置保留不动）

固定定位：相对于浏览器窗口

#### 浮动布局

  文档流：浏览器根据元素在html文档中出现的顺序，从左向右，从上到下依次排列。

设置元素的浮动，该元素将脱离文档流，向左或向右移动直到它的外边距碰到父元素的边框或另一个浮动元素的边框为止

行内元素：行内元素的表现形式主要在于填充内容，它默认是没有上下外边距的，只可以左右，不能设置宽和高

#### 清除浮动

浮动溢出，容器没有包围浮动元素，clear 只对块级元素有效，表示如果前一个元素存在左浮动或右浮动，则换行。与浮动框互不干扰，需要对该框应用。

浮动元素后使用一个空元素clear:both;

浮动元素的容器添加overflow:hidden;或overflow:auto;可以清除浮动，另外在 IE6 中还需要触发 hasLayout ，例如为父元素设置容器宽高或设置 zoom:1；

使用CSS的:after伪元素，给浮动元素的容器添加一个clearfix的class，然后给这个class添加一个:after伪元素实现元素末尾添加一个看不见的块元素（Block element）清理浮动，兼容IE触发 hasLayout。

## 九、ajax

jQuery sortable拖拽排序

ajax对象核心XMLHttpRequest

创建XMLRequestHttp对象

open(‘请求方式’ , ’URL’ , 异步还是同步)

请求方式：post/get 异步：true 同步：false

send发送

onload成功 onerror失败 responseText 返回内容 status状态

onreadystatechange：每当 readyState 属性改变时，就会调用该函数

readyState ：存有 XMLHttpRequest 的状态。

从 0 到 4 发生变化。

0: 请求未初始化

1: 服务器连接已建立

2: 请求已接收

3: 请求处理中

4: 请求已完成，且响应已就绪

status：响应状态

200: "OK"  
 404: 未找到页面

## less

#### 嵌套

& 代表父级

a:hover{}

a:{

&：hover{}

}

#### 变量

定义通用的样式

@变量名：值；

变量可以参与计算

作用域 {}，只在作用域内有效

#### 混合

classB继承classA的样式，注意先后顺序，有.classA:after{}，得放在classA内部

.classA{

&:after{}

}

.classB{

.classA

}

传参数

.classA(@color){

border:1px solid @color

}

.classB{

.classA(red);

}

多个参数

用逗号分隔开

默认参数

@arguments 所有参数

模式匹配

@\_ 默认的，首参一定要传

.classA(top,@color){

border-top:1px solid @color

}

.classA(@\_,@color){

width:100px

}

.classB{

.classA(top);

}

math函数

round() 四舍五入

cell() 向上取整

floor() 向下取整

命名空间

关联文件

避免编译 ~’/’

## bootstrap

#### 栅格系统

容器

宽度百分比流体 container-fluid

宽度固定 container 1170-970-750-auto 针对不同分辨率显示不同宽度

固定和流体不要相互嵌套，两种容器都会有padding

阈值：1200-992-768

把页面分成12列 row col

col-lg-1 到col-lg-12

lg 电脑>1200

md >992

sm >768

xs 手机 <768

组合实现

列偏移 col-lg-offset-n 向右偏移n格

排序col-lg-push-n往右col-lg-pull-n往左

嵌套

clearfix 清除浮动的class

响应式工具

针对设备页面上的一些内容展示或隐藏

visible-lg-block显示

hidden-sm 隐藏

打印

visible-lg-print-block

hidden-print

比如导航栏

pull-left左浮动 pull-right右浮动 affix固定定位

字体图标

#### 预定义样式风格

primary首选项

success 成功

info 一般信息

warning 警告

danger 危险

面板 panel

按钮 btn btn-sm btn-default btn-link btn-block

active选中状态 disabled禁用

按钮组 btn-group

#### 下拉菜单

dropdown open dropdown-menu

自定义属性

data-\* 用来操作Js交互的

aria-\* 阅读器

role

#### 选项卡

nav 头部 nav-tabs nav-pills

移动端优先，所以没有移入移出

#### 导航条

nav navbar-\*

## 移动端

viewport

user-scale=no||yes 禁止/允许缩放，ios10没效->preventDefault

minimum-scale

maximum-scale

initial-scale

像素比：devicePixelRatio 把一个像素放大至n个像素显示

x5内核禁止横屏 全屏显示

UC禁止横屏 全屏显示

禁止识别邮箱和电话

href=”tel:phonenumber”

href=”mailto:emailadress”

问题：

a标签 高亮

按钮 圆角 -webkit-

字体 font-family:Helvetica

横竖屏切换字体放大 -webkit-text-size-adjust:100%

font-boosting webkit内核 把没设置高度的文字放大 设置高度或者max-height

fixed

ios 的body overflow：hidden还能横向拖动，用一个div包住对div用overflow：hidden或者->preventDefault

禁止系统的默认滚动条，禁止下拉回弹的效果、禁止长按选中文字和图片、系统默认菜单、点透问题

但是 会导致input聚焦问题->stopPropagation()

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

ev.preventDefault()

})

#### 适配

宽度百分比

viewport js生成meta标签，缩放比例：屏幕尺寸/目标尺寸

rem适配

弹性盒模型

display:flex 兼容性

display：-webkit-box

#### 移动端事件

pc端点击300ms延迟

addEventListener来绑定事件

touchstart

touchend

touchmove

touchEvent

touches 几个手指

targetTouches 当前元素内几个手指

changeTouches 手指数变化

#### 手势操作

ios

两指或两指以上触摸屏幕

gesturestart

两指或两指以上触摸屏幕抬起

gestureend

手指操作图片缩放与旋转

gesturechange

e.scale change时手指距离/start时手指距离

e.rotation 旋转角度差

不兼容安卓

#### transform

-webkit-transform

变换原点（rotate，scale，skew影响，translate不影响）

transform-orign:center,center

执行顺序：先写的后执行

角度转弧度 deg\*Math.PI/180

弧度转角度 dig\*190/Math.PI

transform实际是个矩阵matrix[1,0,0,1,0,0] =>matrix[a,b,c,d,e,f]

translateX:e

translateY:f

scaleX:a\*scale,c\*scale,e\*scale

scaleY:b\*scale,d\*scale,f\*scale

skewY:b=>Math.tan(deg\*Math.PI/180)

skewX:c=>Math.tan(deg\*Math.PI/180)

rotate:

a:Math.cos(deg\*Math.PI/180)

b:Math.sin(deg\*Math.PI/180)

c:Math.sin(deg\*Math.PI/180)

d:Math.cos(deg\*Math.PI/180)

3d变换（rotate translate）

变换元素的父元素设置景深perspective （近大远小）

transform-style元素变换子元素的3d变换是否保留 flat不保留 preserve-3d 保留

backface-visibility：hidden隐藏背面

getComputedStyle(div)[“transform”]