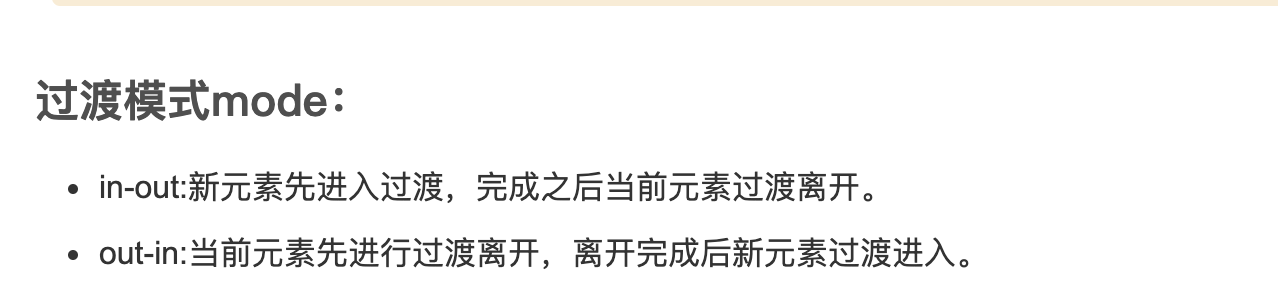
阶段总结和关于留声机项目的总结

在通过我写‘时代留声机’的两个页面，得到组内的反馈是有一些我的代码中的问题。比如  
一、样式 布局写的不行，很烂，于是我在项目结束后查阅资料学习了下很多项目实战中的有关于flex布局的问题，  
总结了一些比较常用的知识点当父容器开启flex布局后，可以设置  
1.justify-content属性，这个属性是定义子容器如何沿着主轴方向排列  
属性值有flex-start 起始端对齐，flex-end末尾对其，center就是居中，space-around就是均匀分布，但是位于首位两端的子容器到父容器的距离是子容器间距的一半，也算是比较常用，space-between也是子容器沿着主轴均匀分布但是首尾两端的子容器和父容器相切。  
2.align-items属性，是定义子容器沿着交叉轴排列间距，属性值有flex-start，flex-end，center，baseline就是沿着基线对齐，确定基线的方式就是，与起点到基线距离最大的子容器与起点相切，来确定基线。stretch就是，子容器沿交叉轴拉伸，填满。  
3.子容器的flex属性，以确定各个子容器的伸缩比例。  
4.子容器的align-self属性，这是用来单独确定哪个子容器的怎么沿交叉轴排列，属性值依旧是有flex-start，flex-end，center，baseline，stretch。  
5.flex-direction 设置主轴的方向 row向右 column向下 row-reverse向左 column-reverse向上。 当主轴确定了，交叉轴就也确定了。  
6.flex-wrap 决定子容器是否换行排列。nowrap 不换行 wrap换行 wrap-reverse逆序换行，也就是沿着交叉轴的反方向换行。  
7.flex-flow就是  flex-direction 与 flex-wrap 的结合，可单独设置任一个也可以同时设置两者。  
8.align-content就是设置有多行时，行与行之间怎么排列。属性值依旧是flex-start、flex-end、center、space-between、space-around、stretch。  
9.子容器的flex-basis，不伸缩的情况下，子容器的原始尺寸，主轴为横向（向右向左）代表宽度，主轴为纵向时（向上向下），就代表高度。  
10.子容器flex-grow，当子容器没填满这一行时，剩下的空间以flex-grow的值的比例分配给各个子容器。  
11.子容器flex-shrink，设置收缩比例，同理当子容器排列超出这一行时，就要把超出部分按比例值减去。  
12.子容器order，决定子容器的排列顺序，可以为负值，值越小排列越靠前。  
Flex布局，我觉得很关键，看了看前辈们写的代码里也有用到了很多，以后我也尽量多用flex布局。  
二、还在空余时间了解了下http缓存相关的知识  
http缓存的定义是：客户端向服务器请求资源时，会先抵达浏览器缓存，如果浏览器有“要请求资源”的副本，就可以直接从浏览器缓存中提取而不是从原始服务器中提取这个资源。  
根据定义可知，从缓存里拿资源都是在第二次请求开始的，我之前一直以为是第一次。   按是否要重新向服务器发请求的方式去分类的话，有强缓存和协商缓存。  
强缓存就是，缓存没失效的情况下就不会向浏览器发送请求，直接拿缓存数据，http状态码是200。header属性里有，按决定强缓存优先级排列，Pragma(存在于http1.0中)、Cache-Control、Expires。  
其中Cache-Control的值有     1、no-cache不直接使用缓存，我可不可以理解为如果有正确的ETag，当配置了no-cache时，客户端会先发出请求向后端验证资源是否发生更改，如果资源未更改，则拿缓存，也即是向服务器端验证。2、no-store不使用缓存  
3、max-age是缓存时长4、public/private是否是单个用户使用5、must-revalidate，本地资源过期之前可以拿缓存，如果过期了之后就要向服务器端验证。  
Expires，表示缓存时长存在时间，如果是和max-age同时存在就被覆盖。但是这样的话如果客户端和服务器端的时间不一致就很容易出现问题，还有如果缓存失效，服务器端资源更改，拿不到最新的资源。  
协商缓存，第一请求的响应里没有Cache-Control和Expires，或者这俩都过期，或者Cache-Control是no-cache，客户端就要请求服务器，和服务器交互，header属性里相关的有ETag对应的if-none-match和last-modified对应的 If-Modified-Since。etag我理解的是一个hash值，第二次请求头中带的 If-None-Match是第一次返回的header中的etag，会做比对，如果发生了变化就把资源返回给浏览器  ，http状态码是200，如果没有变化就给浏览器返回一个304未修改。这样通过控制浏览器端的缓存，可以节省服务器的带宽，因为服务器不需要每次都把全量数据返回给客户端。  
Etag优先级高一些。  
三、看了看之前的项目代码，总结了一些知识点，还有项目中需要第一次遇到的。  
some函数  
every函数  
every()是对数组中每一项运行给定函数，如果该函数对每一项返回true,则返回true。  
some()是对数组中每一项运行给定函数，如果该函数对任一项返回true，则返回true。  
var arr = [ 1, 2, 3, 4, 5, 6 ];   
   
console.log( arr.some( function( item, index, array ){   
    console.log( &apos;item=&apos; + item + &apos;,index=&apos;+index+&apos;,array=&apos;+array );   
    return item > 3;   
}));   
   
console.log( arr.every( function( item, index, array ){   
    console.log( &apos;item=&apos; + item + &apos;,index=&apos;+index+&apos;,array=&apos;+array );   
    return item > 3;   
}));  
  
ES6为Array增加了find()，findIndex函数。  
find()函数用来查找目标元素，找到就返回该元素，找不到返回undefined。  
findIndex()函数也是查找目标元素，找到就返回元素的位置，找不到就返回-1。  
他们的都是一个查找回调函数。  
  
es6 Object.is()方法  
与严格比较运算符（===）的行为基本一致。不同之处只有两个：一是+0不等于-0，二是NaN等于自身。  
  
路由切换时的动画。  


  
fill函数  
this.$route.params 和this.$route.query的区别  
query可以通过name或path来引入路由  
params只能通过name来引入路由，path会undefined  
  
  
传递参数时的区别  
  
  
  
  
还接触了npm的一些模块，如throttle-debounce，还有jsonp，我们公司对于jsonp模块的封装使用都也以理解，还安装并调试好了charles，也学会了抓包，以上知识也都运用在了项目中。  
还有其他的知识，很多都记在本子上了。留声机算是我人生中参与的第一个上线的项目，给那么多人点进去使用，虽然我的代码只占了很少很少，但是能通过哥哥们的指导，熟悉项目开发规范，并且学知识，感觉非常荣幸。