1.scrollBehavior (to, from, savedPosition) {

        console.log(to) // to：要进入的目标路由对象，到哪里去

        console.log(from) // from：离开的路由对象，哪里来

        console.log(savePosition) // savePosition：会记录滚动条的坐标，点击前进/后退的时候记录值{x:?,y:?}

     if (savedPosition) {

            console.log(savedPosition)

            return savedPosition

        } else {

            if (from.meta.keepAlive) {//如果keepAlive的话，保存停留的位置：

                from.meta.savedPosition = document.body.scrollTop;

            }

         return { x: 0, y: to.meta.savedPosition || 0 }

        }

    }

2、async的意思是"异步"，[顾名思义](https://www.baidu.com/s?wd=%E9%A1%BE%E5%90%8D%E6%80%9D%E4%B9%89&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)是有关异步操作有关的关键字。  申明async方法比较简单，只需要在普通的函数前加上"async"关键字即可。我们执行下这个函数，发现并没有返回字符串"helloAsync",而是通过Promise.resolved()将字符串封装成了一个Promise对象返回。  
async function helloAsync(){

return "helloAsync";

}

helloAsync().then(v=>{

console.log(v);*//"helloAsync"*

})

3、XLink，即 XML 链接语言，用于在 XML 文件上创建内部和外部链接，以及与这些链接相关联的元数据。

属性（xlink:href）值（*URL*） 描述（要连接的url）

4、**svg 修改image的xmlns:xlink及图片的显隐**

**5、<aside> 标签定义其所处内容之外的内容。aside 的内容应该与附近的内容相关。**

**有两种使用方法**

1. 最典型的是侧边栏，其中的内容可以使友情链接，博客中的其它文章列表、广告单元等。
2. 被包含在article元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、名次解释，等等。

6、**v-load-more：**

**vue自定义指令v-load-more实现上拉加载**

7、ellipse标签可用来创建椭圆 椭圆与圆很相似。不同之处在于椭圆有不同的 x 和 y 半径，而圆的 x 和 y 半径是相同的。

**代码解释：**

* cx 属性定义圆点的 x 坐标
* cy 属性定义圆点的 y 坐标
* rx 属性定义水平半径
* ry 属性定义垂直半径

8、[beforeDestroy](https://segmentfault.com/q/1010000012216800)：vue的生命周期

假设有三个页面，分别为A、B、C页面，其中B页面定义了beforeDestroy生命周期函数。页面执行顺序是A -> B -> C,当从B到C时，beforeDestroy 函数会执行，但由B点击浏览器返回按钮时返回A时，beforeDestroy 不触发。

9、[Vue路由的$router.back(-1)回退时如何判断有没有上一个路由](https://segmentfault.com/q/1010000010714863)

现在每个页面的左上角有一个返回按钮<  
点击时的代码是this.$router.back(-1)，返回上一个路由  
但是我们现在有这样一个需求，把其中某一页分享出去，用户打开时并没有上一条路由的历史记录，所以点击<按钮时没有反应。  
所以应该怎么判断有没有上一条路由的历史记录。  
代码：  
routerback: function () {  
this.$router.back(-1)  
},

* 10、vue中transition组建标签 起到一个过度的作用 条件渲染 (使用 v-if)
* 条件展示 (使用 v-show) 隐藏显示的时候会有一个过度的效果 必须用transition包起来 具体看官网https://cn.vuejs.org/v2/guide/transitions.html

11、同时生效的进入和离开的过渡不能满足所有要求，

Vue 提供了 过渡模式   
in-out：新元素先进行过渡，完成之后当前元素过渡离开。   
out-in：当前元素先进行过渡，完成之后新元素过渡进入。

12、form: {  
…id && {id: id},  
}

三个点的意思是扩展运算符，是ES6的语法

意思是： 如果id存在，不为undefined，就用后面的对象填充进form对象中

13、SVG <circle>

**<circle> 标签可用来创建一个圆。**

**代码解释：**

cx 和 cy 属性定义圆点的 x 和 y 坐标。如果省略 cx 和 cy，圆的中心会被设置为 (0, 0)

r 属性定义圆的半径。

14、**@include指令**

使用include指令可以方便的把多个页面中重复显的内容抽取出来，大大的减少代码的重复量，方便我们对重复内容的维护

15、**Async**

async. 默认是true，即为异步方式，$.Ajax执行后，会继续执行ajax后面的脚本，直到服务器端返回数据后，触发$.Ajax里的success方法，这时候执行的是两个线程。若要将其设置为false，则所有的请求均为同步请求，在没有返回值之前，同步请求将锁住浏览器，用户其它操作必须等待请求完成才可以执行。

* 16、**scrollBehavior滚动行为**

使用前端路由，当切换到新路由时，想要页面滚到顶部，或者是保持原先的滚动位置，就像重新加载页面那样。 vue-router 能做到，而且更好，它让你可以自定义路由切换时页面如何滚动。接收to和from两个路由对象

scrollBehavior (to, from, savedPosition) {

// return 期望滚动到哪个的位置

17、混入 (mixins) 是一种分发 Vue 组件中可复用功能的非常灵活的方式。混入对象可以包含任意组件选项。当组件使用混入对象时，所有混入对象的选项将被混入该组件本身的选项。

例子：

|  |
| --- |
| // 定义一个混入对象 var myMixin = {  created: function () {  this.hello()  },  methods: {  hello: function () {  console.log('hello from mixin!')  }  } }  // 定义一个使用混入对象的组件 var Component = Vue.extend({  mixins: [myMixin] })  var component = new Component() // => "hello from mixin!" |

18、**什么是async、await？**

async顾名思义是“异步”的意思，async用于声明一个函数是异步的。而await从字面意思上是“等待”的意思，就是用于等待异步完成。并且await只能在async函数中使用

通常async、await都是跟随Promise一起使用的。为什么这么说呢？因为async返回的都是一个Promise对象同时async适用于任何类型的函数上。这样await得到的就是一个Promise对象(如果不是Promise对象的话那async返回的是什么 就是什么)；

await得到Promise对象之后就等待Promise接下来的resolve或者reject。

1 async function testSync() {

2 const response = await new Promise(resolve => {

3 setTimeout(() => {

4 resolve("async await test...");

5 }, 1000);

6 });

7 console.log(response);

8 }

9 testSync();//async await test...

一个简单的async、await异步使用async、await完成异步操作代码可读与写法上更像是同步的