1. async 函数返回一个promise 对象， 可以使用then 方法添加回调函数。当函数执行的时候，一旦遇到await就会先返回，到触发的异步操作完成，再接着执行函数体内后面的语句。
2. v-model.lazy当添加.lazy修饰符之后，改变input框中的内容并不会使得span中的内容发生变化，此时当输入框失去焦点后触发[change事件](https://www.baidu.com/s?wd=change%E4%BA%8B%E4%BB%B6&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd).控制台中输出相应内容；
3. click.prevent 阻止事件的默认行为
4. xlink:href属性像参考<IRI>一样定义到了资源的链接。该链接的确切含义取决于使用它的每个元素的上下文
5. ellipse 标签可用来创建椭圆。

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；

1. transition是一个过渡，如果里面写入v-for 编辑器就会出现红线但不会影响效果
2. Vue.nextTick() 在下次 DOM 更新循环结束之后执行延迟回调。在修改数据之后立即使用这个方法，获取更新后的 DOM
3. clearTimeout() 方法可取消由 setTimeout() 方法设置的 timeout；

clearTimeout() 方法的参数必须是由 setTimeout() 返回的 ID 值

1. refresh() 函数刷新当前卡片，如果刷新成功，则返回一个空字符串。如果失败，则返回一个非空字符串
2. getBoundingClientRect() 用于获取某个元素相对于视窗的位置集合 top,lef,right,bottom,width,height;
3. tap，zepto的tap事件, 利用touchstart和touchend来模拟click事件；缺点: 点击穿透;
4. minxin使用及介绍

vue中提供了一种混合机制--mixins，用来更高效的实现组件内容的复用。最开始我一度认为这个和组件好像没啥区别。。后来发现错了。下面我们来看看mixins和普通情况下引入组件有什么区别？

组件在引用之后相当于在父组件内开辟了一块单独的空间，来根据父组件props过来的值进行相应的操作，单本质上两者还是泾渭分明，相对独立。

而mixins则是在引入组件之后，则是将组件内部的内容如data等方法、method等属性与父组件相应内容进行合并。相当于在引入后，父组件的各种属性方法都被扩充了。

单纯组件引用：

父组件 + 子组件 >>> 父组件 + 子组件

mixins：

父组件 + 子组件 >>> new父组件

作用：多个组件可以共享数据和方法，在使用mixin的组件中引入后，mixin中的方法和属性也就并入到该组件中，可以直接使用。钩子函数会两个都被调用，mixin中的钩子首先执行。

下面给大家介绍vue mixin的用法，具体介绍如下所示：

~定义一个 js 文件(mixin.js)

export default {

data() {

return {

name: 'mixin'

}

},

created() {

console.log('mixin...', this.name);

},

mounted() {},

methods: {}

}

~~在vue文件中使用mixin

import '@/mixin'; // 引入mixin文件

export default {

mixins: [mixin]

}

详见网址：<https://blog.csdn.net/qq_36838191/article/details/81004590>

1. svg：既然资源是svg, 那么最好是使用svg标签去引用, 这样可以对svg调整大小, 设置填充色等一系列的svg特性都可以使用。具体怎么引用要看使用什么loader了:

html:

<use xmlns:xlink="http://www.w3.org/1999/xlink" :xlink:href="require('../svg/icon-header.svg')"></use>

webpack:配置：

{

test: /\.svg$/,

loader: 'svg-sprite?' + JSON.stringify({

name: '[name]\_[hash]',

prefixize: true

})

}

详见网址：<https://segmentfault.com/q/1010000009942523/a-1020000009947681>

1. go back与return有什么区别

go back回去,回到某一个地方；return是从a地到b地,然后强调了又从b地回到a地；  
 而且return还有一个用法就是指借了东西,归还回去

1. 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

}

1. svg 里面的 **<circle>**[SVG](https://developer.mozilla.org/en-US/docs/Web/SVG) 元素是一个SVG的基本形状，用来创建圆,基于一个圆心和一个半径。

实例 ：

<?xml version="1.0"?>

<svg viewBox="0 0 120 120" version="1.1" xmlns="http://www.w3.org/2000/svg">

<circle cx="60" cy="60" r="50"/>

</svg>

Svg 里面的 <line> 标签用来创建线条

实例 ：

<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"

"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="100%" height="100%" version="1.1"

xmlns="http://www.w3.org/2000/svg">

<line x1="0" y1="0" x2="300" y2="300"

style="stroke:rgb(99,99,99);stroke-width:2"/>

</svg>

1. <figure> 标签 用作文档中插图的图像 是html新增标签

<figure> 标签规定独立的流内容（图像、图表、照片、代码等等）。

figure 元素的内容应该与主内容相关，但如果被删除，则不应对文档流产生影响。

实例:

<figure>

<p>黄浦江上的的卢浦大桥</p>

<img src="shanghai\_lupu\_bridge.jpg" width="350" height="234" />

</figure>

<figcaption> 标签用作文档中插图的图像，带有一个标题；

<figcaption> 标签定义 [figure](http://www.w3school.com.cn/tags/tag_figure.asp) 元素的标题（caption）。

"figcaption" 元素应该被置于 "figure" 元素的第一个或最后一个子元素的位置

1. decodeURIComponent() 函数可对 encodeURIComponent() 函数编码的 URI 进行解码。

实例:

<script type="text/javascript">

var test1="http://www.w3school.com.cn/My first/"

document.write(encodeURIComponent(test1)+ "<br />")

document.write(decodeURIComponent(test1))

</script>

返回值: URIstring 的副本，其中的十六进制转义序列将被它们表示的字符替换。

1. [Vue scrollBehavior 滚动行为](https://www.cnblogs.com/sophie_wang/p/7880261.html)

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。

注意: 这个功能只在 HTML5 history 模式下可用。

详细地址见: <https://www.cnblogs.com/sophie_wang/p/7880261.html>