



## 第二版：Vue 39 道

### 目录

第二版：Vue 39 道	1
1、JS 中判断数据类型的方法有几种？	2
2、vue 与 angular 的区别？	3
3、说说你对 angular 脏检查理解？	3
4、active-class 是哪个组件的属性？	3
5、嵌套路由怎么定义？	3
6、怎么定义 vue-router 的动态路由？怎么获取传过来的动态参数？	5
7、vue-router 有哪几种导航钩子？	5
8、scss 是什么？在 vue.cli 中的安装使用步骤是？有哪几大特性？	6
1、可以用变量，例如（\$变量名称=值）；	6
2、可以用混合器，例如（）	6
3、可以嵌套	6
9、mint-ui 是什么？怎么使用？说出至少三个组件使用方法？	6
10、v-model 是什么？怎么使用？vue 中标签怎么绑定事件？	7
11、iframe 的优缺点？	7
1、解决加载缓慢的第三方内容如图标和广告等的加载问题	7
2、Security sandbox	7
3、并行加载脚本	7
4、方便制作导航栏	7
1、iframe 会阻塞主页面的 Onload 事件	7
2、即时内容为空，加载也需要时间	7
3、没有语义	7
12、简述一下 Sass、Less，且说明区别？	7
13、axios 是什么？怎么使用？描述使用它实现登录功能的流程？	8
14、axios+tp5 进阶中，调用 axios.post( 'api/user' )是进行的什么操作？	8
axios.put( 'api/user/8')呢？	8
15、vuex 是什么？怎么使用？哪种功能场景使用它？	8
16、mvvm 框架是什么？它和其它框架（jquery）的区别是什么？哪些场景适合？	8



17、自定义指令 (v-check、v-focus) 的方法有哪些？它有哪些钩子函数？还有哪些钩子函数参数？	8
18、说出至少 4 种 vue 当中的指令和它的用法？	9
19、vue-router 是什么？它有哪些组件？	9
20、导航钩子有哪些？它们有哪些参数？	9
21、Vue 的双向数据绑定原理是什么？	9
22、请详细说下你对 vue 生命周期的理解？	10
23、请说下封装 vue 组件的过程？	11
24、你是怎么认识 vuex 的？	11
25、vue-loader 是什么？使用它的用途有哪些？	11
26、请说出 vue.cli 项目中 src 目录每个文件夹和文件的用法？	12
27、vue.cli 中怎样使用自定义的组件？有遇到过哪些问题吗？	12
28、聊聊你对 Vue.js 的 template 编译的理解？	12
29、vue 的历史记录	13
30、vuejs 与 angularjs 以及 react 的区别？	13
31、与 AngularJS 的区别	13
32.与 React 的区别	14
33、什么是 vue 生命周期？	14
34、vue 生命周期的作用是什么？	14
35、vue 生命周期总共有几个阶段？	15
36、第一次页面加载会触发哪几个钩子？	15
37、DOM 渲染在 哪个周期中就已经完成？	15
38、简单描述每个周期具体适合哪些场景？	15
39、cancas 和 SVG 的是什么以及区别	15
Canvas 与 SVG 的比较	16

我们的网站: <https://tech.souyunku.com>

关注我们的公众号: **搜云库技术团队**, 回复以下关键字

回复: **进群** 邀请您进 **「技术架构分享群」**

微信搜一搜

搜云库技术团队



回复:【内推】即可进: 北京, 上海, 广州, 深圳, 杭州, 成都, 武汉, 南京,

郑州, 西安, 长沙「程序员工作内推群」

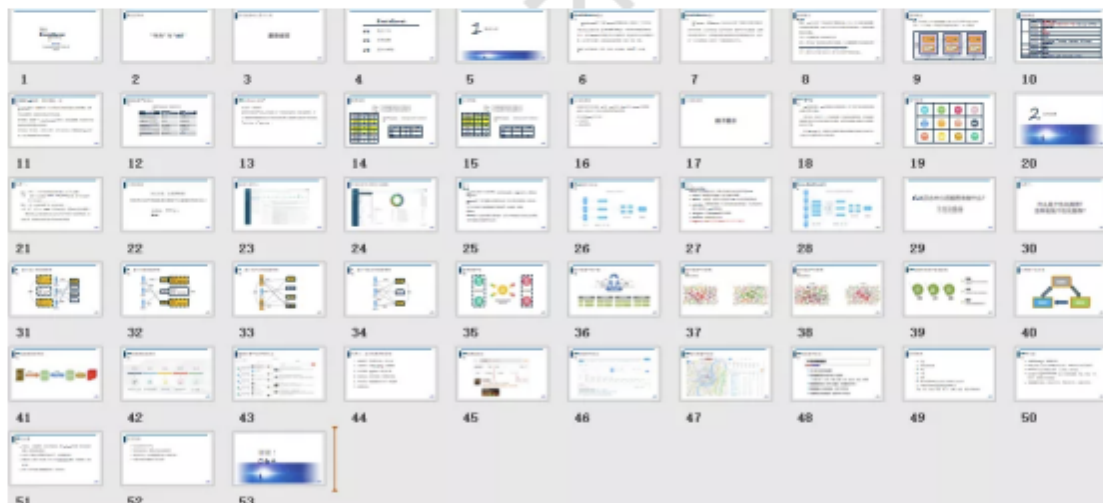
回复【1024】送 4000G 最新架构师视频

回复【PPT】即可无套路获取, 以下最新整理调优 PPT!

## 46 页《JVM 深度调优, 演讲 PPT》



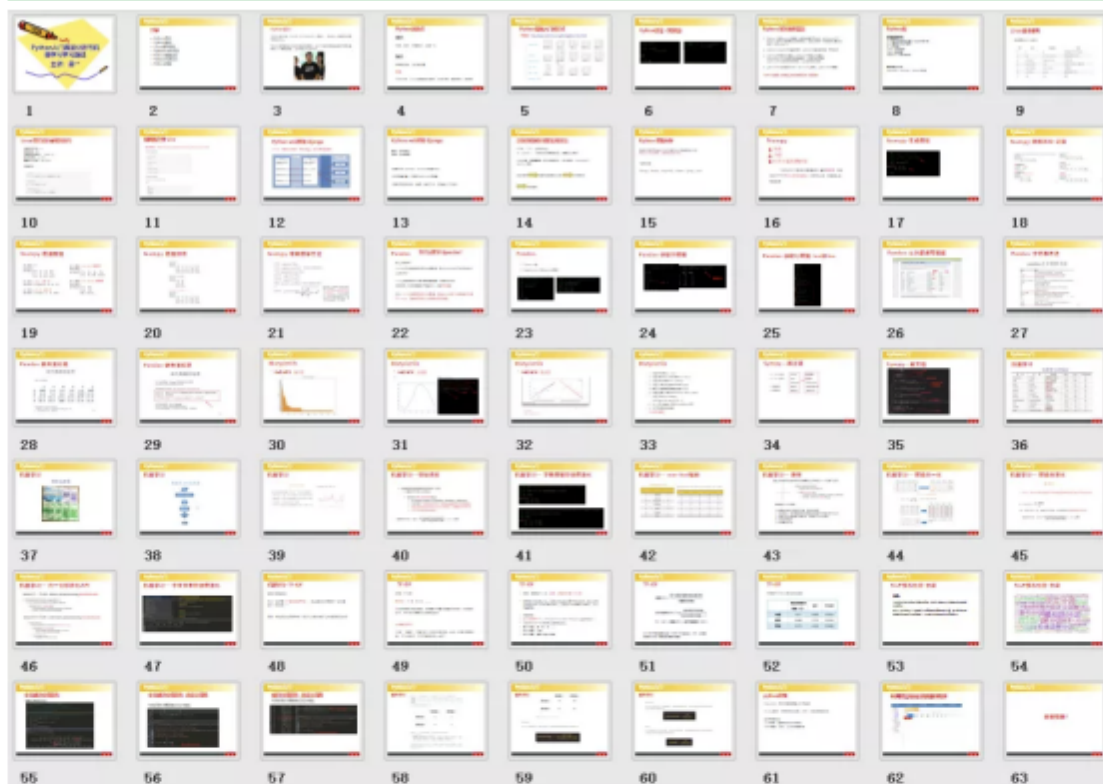
## 53 页《Elasticsearch 调优演讲 PPT》



## 63 页《Python 数据分析入门 PPT》

微信搜一搜

搜云库技术团队



微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推  
专注于分享最有价值的互联网技术干货文章

## 1、JS 中判断数据类型的方法有几种？



最常见的判断方法：typeof

判断已知对象类型的方法： instanceof

根据对象的 constructor 判断： constructor

无敌万能的方法：jquery.type()

## 2、vue 与 angular 的区别？

- 1、vue 仅仅是 mvvm 中的 view 层，只是一个如 jquery 般的工具库，而不是框架，而 angular 而是 mvvm 框架。
- 2、vue 的双向绑定是基于 ES5 中的 3.getter/setter 来实现的，而 angular 而是由自己实现一套模版编译规则，需要进行所谓的“脏”检查，vue 则不需要。因此，vue 在性能上更高效，但是代价是对于 ie9 以下的浏览器无法支持。
- 4、vue 需要提供一个 el 对象进行实例化，后续的所有作用范围也是在 el 对象之下，而 angular 而是整个 html 页面。一个页面，可以有多个 vue 实例，而 angular 好像不是这么玩的。
- 5、vue 真的很容易上手，学习成本相对低，不过可以参考的资料不是很丰富，官方文档比较简单，缺少全面的使用案例。高级的用法，需要自己去研究源码，至少目前是这样。

## 3、说说你对 angular 脏检查理解？

在 angular 中你无法判断你的数据是否做了更改，所以它设置了一些条件，当你触发这些条件之后，它就执行一个检测来遍历所有的数据，对比你更改的地方，然后执行变化。

这个检查很不科学。而且效率不高，有很多多余的地方，所以官方称为 脏检查。





## 4、active-class 是哪个组件的属性？

vue-router 模块的 router-link 组件。

## 5、嵌套路由怎么定义？

在实际项目中我们会碰到多层嵌套的组件组合而成，但是我们如何实现嵌套路由呢？因此我们需要在 VueRouter 的参数中使用 children 配置，这样就可以很好的实现路由嵌套。

index.html，只有一个路由出口

```
<div id="app">
  <!-- router-view 路由出口, 路由匹配到的组件将渲染在这里 -->
  <router-view> </router-view>
</div>
```

main.js，路由的重定向，就会在页面一加载的时候，就会将 home 组件显示出来，因为重定向指向了 home 组件，redirect 的指向与 path 的必须一致。children 里面是子路由，当然子路由里面还可以继续嵌套子路由。

```
import Vue from 'vue'
import VueRouter from 'vue-router'
Vue.use(VueRouter)

//引入两个组件

import home from './home.vue'
import game from './game.vue'
//定义路由
```



```
const routes = [
  { path: "/", redirect: "/home" },//重定向,指向了 home 组件
  {
    path: "/home", component: home,
    children: [
      { path: "/home/game", component: game }
    ]
  }
]
//创建路由实例
const router = new VueRouter({routes})

new Vue({
  el: '#app',
  data: {
  },
  methods: {
  },
  router
})
```

home.vue, 点击显示就会将子路由显示在出来, 子路由的出口必须在父路由里面, 否则子路由无法显示。

```
<template>
  <div>
    <h3>首页</h3>
    <router-link to="/home/game">
      <button>显示</button>
    </router-link>
    <router-view></router-view>
```



```
</div>
</template>
```

game.vue

```
<template>
  <h3>游戏</h3>
</template>
```

## 6、怎么定义 vue-router 的动态路由？怎么获取传过来的动态参数？

在 router 目录下的 index.js 文件中，对 path 属性加上/:id。  
使用 router 对象的 params.id。

## 7、vue-router 有哪几种导航钩子？

三种，

第一种：是全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。

第二种：组件内的钩子

第三种：单独路由独享组件

## 8、scss 是什么？在 vue.cli 中的安装使用步骤是？有哪几大特性？





css 的预编译。

使用步骤：

第一步：用 npm 下三个 loader (sass-loader、css-loader、node-sass)

第二步：在 build 目录找到 webpack.base.config.js，在那个 extends 属性中加一个拓展.scss

第三步：还是在同一个文件，配置一个 module 属性

第四步：然后在组件的 style 标签加上 lang 属性，例如：lang="scss"

有哪几大特性：

- 1、可以用变量，例如（\$变量名称=值）；
- 2、可以用混合器，例如（）
- 3、可以嵌套

## 9、mint-ui 是什么？怎么使用？说出至少三个组件使用方法？

基于 vue 的前端组件库。npm 安装，然后 import 样式和 js，vue.use (mintUi) 全局引入。在单个组件局部引入：import {Toast} from 'mint-ui'。

组件一：Toast( '登录成功' );

组件二：mint-header;

组件三：mint-swiper

## 10、v-model 是什么？怎么使用？ vue 中标签怎么绑定事件？



可以实现双向绑定, 指令 (v-class、v-for、v-if、v-show、v-on)。vue 的 model 层的 data 属性。绑定事件: `<input @click=doLog()/>`

## 11、iframe 的优缺点?

iframe 也称作嵌入式框架, 嵌入式框架和框架网页类似, 它可以把一个网页的框架和内容嵌入在现有的网页中。

优点:

- 1、 解决加载缓慢的第三方内容如图标和广告等的加载问题
- 2、 Security sandbox
- 3、 并行加载脚本
- 4、 方便制作导航栏

缺点:

- 1、 iframe 会阻塞主页面的 Onload 事件
- 2、 即时内容为空, 加载也需要时间
- 3、 没有语意

## 12、简述一下 Sass、Less, 且说明区别?

他们是动态的样式语言, 是 CSS 预处理器,CSS 上的一种抽象层。他们是一种特殊的语法/语言而编译成 CSS。

变量符不一样, less 是@, 而 Sass 是\$;

Sass 支持条件语句, 可以使用 if{}else{},for{}循环等等。而 Less 不支持;

Sass 是基于 Ruby 的, 是在服务端处理的, 而 Less 是需要引入 less.js 来处理 Less 代码输出 Css 到浏览器



### 13、axios 是什么？怎么使用？描述使用它实现登录功能的流程？

请求后台资源的模块。npm install axios -S 装好，然后发送的是跨域，需在配置文件中 config/index.js 进行设置。后台如果是 Tp5 则定义一个资源路由。js 中使用 import 进来，然后.get 或.post。返回在.then 函数中如果成功，失败则是在.catch 函数中

### 14、axios+tp5 进阶中，调用 axios.post( 'api/user' )是进行的什么操作？ axios.put( 'api/user/8')呢？

跨域，添加用户操作，更新操作。

### 15、vuex 是什么？怎么使用？哪种功能场景使用它？

vue 框架中状态管理。在 main.js 引入 store，注入。新建了一个目录 store，..... export 。场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

### 16、mvvm 框架是什么？它和其它框架 (jquery) 的区别是什么？哪些场景适合？

一个 model+view+viewModel 框架，数据模型 model，viewModel 连接两个



区别：vue 数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多的场景，更加便捷

## 17、自定义指令 (v-check、v-focus) 的方法有哪些？它有哪些钩子函数？还有哪些钩子函数参数？

全局定义指令：在 vue 对象的 directive 方法里面有两个参数，一个是指令名称，另外一个函数。组件内定义指令：directives

钩子函数：bind（绑定事件触发）、inserted（节点插入的时候触发）、update（组件内相关更新）

钩子函数参数：el、binding

## 18、说出至少 4 种 vue 当中的指令和它的用法？

v-if：判断是否隐藏；v-for：数据循环出来；v-bind:class：绑定一个属性；v-model：实现双向绑定

## 19、vue-router 是什么？它有哪些组件？

vue 用来写路由一个插件。router-link、router-view

## 20、导航钩子有哪些？它们有哪些参数？



导航钩子有：

a/全局钩子和组件内独享的钩子。b/beforeRouteEnter、afterEnter、beforeRouterUpdate、beforeRouteLeave

参数：

有 to（去的那个路由）、from（离开的路由）、next（一定要用这个函数才能去到下一个路由，如果不用就拦截）最常用就这几种

## 21、Vue 的双向数据绑定原理是什么？

vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过 Object.defineProperty()来劫持各个属性的 setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

具体步骤：

**第一步：**需要 observe 的数据对象进行递归遍历，包括子属性对象的属性，都加上 setter 和 getter

这样的话，给这个对象的某个值赋值，就会触发 setter，那么就能监听到了数据变化

**第二步：**compile 解析模板指令，将模板中的变量替换成数据，然后初始化渲染页面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图

**第三步：**Watcher 订阅者是 Observer 和 Compile 之间通信的桥梁，主要做的事情是：



- 1、在自身实例化时往属性订阅器(dep)里面添加自己
- 2、自身必须有一个 update()方法
- 3、待属性变动 dep.notice()通知时,能调用自身的 update()方法,并触发 Compile 中绑定的回调,则功成身退。

**第四步：**MVVM 作为数据绑定的入口，整合 Observer、Compile 和 Watcher 三者，通过 Observer 来监听自己的 model 数据变化，通过 Compile 来解析编译模板指令，最终利用 Watcher 搭起 Observer 和 Compile 之间的通信桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据 model 变更的双向绑定效果。

## 22、请详细说下你对 vue 生命周期的理解？

总共分为 8 个阶段创建前/后，载入前/后，更新前/后，销毁前/后

**创建前/后：**在 beforeCreated 阶段,vue 实例的挂载元素 \$el 和数据对象 data 都为 undefined，还未初始化。在 created 阶段,vue 实例的数据对象 data 有了，\$el 还没有。

**载入前/后：**在 beforeMount 阶段,vue 实例的 \$el 和 data 都初始化了，但还是挂载之前为虚拟的 dom 节点，data.message 还未替换。在 mounted 阶段,vue 实例挂载完成，data.message 成功渲染。

**更新前/后：**当 data 变化时，会触发 beforeUpdate 和 updated 方法。

**销毁前/后：**在执行 destroy 方法后，对 data 的改变不会再触发周期函数，说明此时 vue 实例已经解除了事件监听以及和 dom 的绑定，但是 dom 结构依然存在





## 23、请说下封装 vue 组件的过程？

首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用 `Vue.extend` 方法创建一个组件，然后使用 `Vue.component` 方法注册组件。子组件需要数据，可以在 `props` 中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用 `emit` 方法。

## 24、你是怎么认识 vuex 的？

vuex 可以理解为一种开发模式或框架。比如 PHP 有 thinkphp，java 有 spring 等。

通过状态（数据源）集中管理驱动组件的变化（好比 spring 的 IOC 容器对 bean 进行集中管理）。

应用级的状态集中放在 `store` 中；改变状态的方式是提交 `mutations`，这是个同步的事物；异步逻辑应该封装在 `action` 中。

## 25、vue-loader 是什么？使用它的用途有哪些？

解析 `.vue` 文件的一个加载器，跟 `template/js/style` 转换成 `js` 模块。

用途：`js` 可以写 `es6`、`style` 样式可以 `scss` 或 `less`、`template` 可以加 `jade` 等

## 26、请说出 vue.cli 项目中 src 目录每个文件夹和文件的用法？



assets 文件夹是放静态资源；components 是放组件；router 是定义路由相关的配置；view 视图；app.vue 是一个应用主组件；main.js 是入口文件

## 27、vue.cli 中怎样使用自定义的组件？有遇到过哪些问题吗？

第一步：在 components 目录新建你的组件文件（smithButton.vue），script 一定要 export default {

第二步：在需要用的页面（组件）中导入：import smithButton from  
'../components/smithButton.vue'

第三步：注入到 vue 的子组件的 components 属性上面，components:{smithButton}

第四步：在 template 视图 view 中使用，<smith-button> </smith-button>  
问题有：smithButton 命名，使用的时候则 smith-button。

## 28、聊聊你对 Vue.js 的 template 编译的理解？

简而言之，就是先转化成 AST 树，再得到的 render 函数返回 VNode（Vue 的虚拟 DOM 节点）

详情步骤：

首先，通过 compile 编译器把 template 编译成 AST 语法树（abstract syntax tree 即 源代码的抽象语法结构的树状表现形式），compile 是 createCompiler 的返回值，createCompiler 是用以创建编译器的。另外 compile 还负责合并 option。



然后，AST 会经过 generate（将 AST 语法树转化成 render funtion 字符串的过程）得到 render 函数，render 的返回值是 VNode，VNode 是 Vue 的虚拟 DOM 节点，里面有（标签名、子节点、文本等等）

## 29、vue 的历史记录

history 记录中向前或者后退多少步

## 30、vuejs 与 angularjs 以及 react 的区别？

## 31、与 AngularJS 的区别

相同点：

都支持指令：内置指令和自定义指令。

都支持过滤器：内置过滤器和自定义过滤器。

都支持双向数据绑定。

都不支持低端浏览器。

不同点：

1、AngularJS 的学习成本高，比如增加了 Dependency Injection 特性，而 Vue.js 本身提供的 API 都比较简单、直观。

2、在性能上，AngularJS 依赖对数据做脏检查，所以 Watcher 越多越慢。



Vue.js 使用基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。

对于庞大的应用来说，这个优化差异还是比较明显的。

## 32.与 React 的区别

相同点：

React 采用特殊的 JSX 语法，Vue.js 在组件开发中也推崇编写 .vue 特殊文件格式，对文件内容都有一些约定，两者都需要编译后使用。

中心思想相同：一切都是组件，组件实例之间可以嵌套。

都提供合理的钩子函数，可以让开发者定制化地去处理需求。

都不内置列数 AJAX，Route 等功能到核心包，而是以插件的方式加载。

在组件开发中都支持 mixins 的特性。

不同点：

React 依赖 Virtual DOM，而 Vue.js 使用的是 DOM 模板。React 采用的 Virtual DOM 会对渲染出来的结果做脏检查。

Vue.js 在模板中提供了指令，过滤器等，可以非常方便，快捷地操作 DOM。

## 33、什么是 vue 生命周期？



Vue 实例从创建到销毁的过程，就是生命周期。也就是从开始创建、初始化数据、编译模板、挂载 Dom→渲染、更新→渲染、卸载等一系列过程，我们称这是 Vue 的生命周期。

### 34、vue 生命周期的作用是什么？

它的生命周期中有多个事件钩子，让我们在控制整个 Vue 实例的过程时更容易形成好的逻辑。

### 35、vue 生命周期总共有几个阶段？

它可以总共分为 8 个阶段：创建前/后，载入前/后，更新前/后，销毁前/销毁后

### 36、第一次页面加载会触发哪几个钩子？

第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

### 37、DOM 渲染在 哪个周期中就已经完成？

DOM 渲染在 mounted 中就已经完成了

### 38、简单描述每个周期具体适合哪些场景？



生命周期钩子的一些使用方法：  
 beforecreate：可以在这加个 loading 事件，在加载实例时触发  
 created：初始化完成时的事件写在这里，如在这结束 loading 事件，异步请求也适宜在这里调用  
 mounted：挂载元素，获取到 DOM 节点  
 updated：如果对数据统一处理，在这里写上相应函数  
 beforeDestroy：可以做一个确认停止事件的确认框  
 nextTick：更新数据后立即操作 dom

arguments 是一个伪数组，没有遍历接口，不能遍历

### 39、cancas 和 SVG 的什么以及区别

#### SVG

SVG 是一种使用 XML 描述 2D 图形的语言。

SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。您可以为某个元素附加 JavaScript 事件处理器。

在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形。

#### Canvas

Canvas 通过 JavaScript 来绘制 2D 图形。

Canvas 是逐像素进行渲染的。

在 canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象。

#### Canvas 与 SVG 的比较

#### Canvas





依赖分辨率

不支持事件处理器

弱的文本渲染能力

能够以 .png 或 .jpg 格式保存结果图像

最适合图像密集型的游戏，其中的许多对象会被频繁重绘

## SVG

不依赖分辨率

支持事件处理器

最适合带有大型渲染区域的应用程序（比如谷歌地图）

复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）

不适合游戏应用