技术面试流程--解答技巧

* 自我介绍

<解答参考>：

这个流程通常按照姓名 → 年龄 → 毕业学院 → 工作经历（包括几个有代表性的作品） → 自己所擅长的领域 → 来公司面试的岗位的基本流程进行。

在自我介绍的时候注意观察面试官的表情。时间不要太短，也不宜太长，这个过程中要控制在1分钟左右。切忌过渡自信，当然也不宜唯唯诺诺显得没有能力。

* 为什么离开上一家公司

<解答参考>：

这个问题是人事的惯用问题，一方面可以通过你的言行来测试你是否真有工作经验，一方面又可以测试你是否能大用。

要杜绝从工薪低、个人偏好不符或压力过大等方面回答。

推荐从公司经营不善裁员、家庭地址搬迁。也可以是些有故事性的更生动理由。

原则是不要被牵着走，也要适当地去引导面试官的询问思路。

如果是已经有行业工作经验，还保留一份真实的上家公司的“离职证明”，但是离这次面试的时间比较久，就可以说是期间和朋友进行了创业，但是由于资金或推广上遇到了问题，再加上期间关键核心人员的退出，公司最终也没有开创起来。从中也发现了自己的一些不足，还需要一些岁月来进行磨练。

* 对我们公司有多少了解

<解答参考>：

“知己知彼，百战不殆”，如果在面试前就对该公司有一定了解，会让别人认为你很有诚意，而不是那种简历“海投”的抽签性质求职，并表示对这个公司很有兴趣，愿意让自己的价值在这里实现。这样会很大程度地获得面试官的好感。

* 简单谈一下你Web前端所擅长的领域

<解答参考>：

这个地方通常会由技术面试官来问，是看看你的核心能力是否是公司还继续需要或项目正缺的一个领域。这样的问题尽量不要答某一两个普通的技术名（如：CSS、JavaScript、jQuery等），因为这是前端在简历里经常都出现“精通”二字描述的技术，同质化太严重，很难让自己脱颖而出。

要回答这样的问题，除了擅长的基本技术外，尽量从稍微广一点、深一点的角度出发。如：

* 对前端整体的架构有一定研究（项目设计之初对项目文件结构以及代码结构的编排组合的可管理维护性）
* 对前端性能的优化有一定研究（通过浏览器开发者工具测试浏览器在加载页面期间的各种能耗分析和测试，图片体积的压缩和图片类型的选用、查询器的存储复用、函数和方法的封装复用、循环语句减少DOM的查询、使用数据驱动的思想等）
* 对前端自动化构建有一定研究（通过package.json快速创建项目依赖，或通过命令行创建所需依赖、以及多文件的打包压缩的管理和分类）
* 拥有MVC思想（模型、视图、控制器分割管理代码的思想）
* 代码的自动化编译和管理（自动转Less/SASS、ES6 → ES5、图片转base64或hash名的图片）等等

总之，在这个环节就要让面试官感觉出你的“等级层次”，和对新兴技术的把握和学习能力，不是一般的泛泛之辈。

* 会不会PS？掌握水平怎么样？谈谈怎么在一个PSD内切出前端开发所需的图形？

<解答参考>：

对PS的使用基本可以满足项目的需求，会使用PS对美工提供的PSD文件进行分析和最佳形式的切图、能制作简单的UI图形或利用不错的素材网提供的素材经过简单设计制作出符合项目风格所需或所缺的界面图形

也可以提到，最新版本的PS使用起来比较简单，以前需要较多步骤才能完成的工作，现在只需要很少几步就能完成。（如右键图层面板后的快速导出功能）

对页面内所需的图形素材能在保证不失真和体积最小之间找到尽可能好的平衡点（使用Ctrl+Alt+Shift+S的功能，选择格式为“jpeg”后通过双联对比来调整图像质量）

* png、jpg、gif这些图片格式有什么区别，什么时候用？怎么优化图像？

<解答参考>：

png常用的有8/16/24/32位，位数越多细节的表现越好（PS内提供了8和24两种基本位数的导出格式）。这种图像格式的优点在于支持alpha通道（透明像素通道，“RGBA”中的“A”即是指“alpha”），能制作出半透明或部分全透明的图片。缺点在于这种图像格式的压缩空间比较有限。这种图像格式通常用于一些对形状细节要求较高的地方，或用于替代老版本IE浏览器不支持CSS3的alpha通道的问题

jpg（jpeg）是一种可压缩图形格式，正式由于其可压缩的特性，在网页中和其它设备中（如手机、单反的照片）被广泛使用，这种格式可以根据需要调整出更精美还是体积更小的图像文件。这种图像格式的缺点在于不支持alpha通道。

gif是这三种图像格式中体积最小的，但确是颜色深度和细节表现最差的一种，支持alpha通道。但正是因为其体积较小的关系，常用于制作动态图片（通过打开PS可以发现，一张动态gif是有多张静态的gif做成的，再通过一定的编码运算排列在一起），当然也有用jpg来制作动态图的，不过一般体积会比较大，是用于制作更为精美的动态图使用，通常是用于演示而不是传媒。除了用于制作动态图，在不使用字体图标的情况下，用来制作单/双色图标也是最好的选择

* 细节技术讨论

1. HTML、CSS、JS之间的关系

<解答参考>：

这里从前端的三层结构的角度出发作答

HTML相当于是一个页面的骨架结构（结构层）

CSS是用于调整这些骨架结构和为这个骨架“添砖加瓦”的存在（表现层）

而JavaScript不仅可以用于控制他们的行为（行为层）

在现在的前端开发中，行为层变得多元化，并且占据着越来越重要的位置，如React使用的JSX语法，基本上将HTML、CSS都写进了JS文件里。

1. 有哪些主流浏览器，内核是什么？

<解答参考>：

谷歌浏览器（Google Chrome） —— 原Webkit，现Blink，JS使用的是V8引擎驱动

火狐浏览器（Mozilla FireFox） —— Cecko，JS引擎不做了解（SpiderMonkey）

Opera —— 原Presto（废弃），现Blink，现在基本和谷歌一样了

Safari —— 原Webkit（KHTML的分支），现Webcore。Windows系统上已废弃

IE（Internet Explorer） —— Trident，OSX/iOS上不可用

Edge（Microsoft Edge） —— EdgeHTML，OSX/iOS上不可用

1. CSS怎么确定样式选择器的优先级，各自权重是多少？

<解答参考>：

这里一定要向面试官确定是CSS引用优先级还是CSS选择器优先级。

如果是引用优先级，则是：行内 > 内嵌（内联）> 外链 > 导入

如果是选择器优先级，基本的是同权重，后出现的覆盖先出现的，常用的几种选择器优先级权重值如下：

0：行内的样式属性，位于HTML标签属性上，如：hidden这个HTML5属性，还有就是通用选择器“\*”，又叫做“通配符”。

1：标签名，位于CSS文件，如：div、p、span、header、main……

10：类名（class）、属性、伪类，通常位于CSS文件或HTML的内嵌样式标签（<style>）内，名称自定义，如：.theme、.bgc、.dialog-maklayer……。属性选择器，如：[src]、[type=”text”]等、伪类如：:hover、:active、:focus等

100：ID，通常位于CSS文件或HTML的内嵌样式标签内，名称自定义：如：#username、#userInfo、#email……

1000：行内样式，位于HTML标签的style属性值内，表现形式和CSS一致。不过不能设置伪类元素样式、

1. 在网页中使用奇数像素的字体好还是偶数像素字体好？为什么？怎么在Chrome浏览器中使用小于12像素的字体。

<解答参考>：

通常来讲，使用偶数字体好，因为计算机计算的最小单位是位，是一种偶数的计算方式，所以现在显示器的显示分辨率都是以偶数作为单位的，偶数的字体像素单位使得字体会更加的“平滑”，而不容易出现锯齿，渲染速度也更快。

但对于现在的浏览器来讲，这个因素的影响正在逐渐减小，由于计算机硬件的进步，浏览器渲染算法的革新，都已经有不错的防锯齿处理机制了，这使得字体具体是奇数还是偶数已经渐渐地不再那么重要了。

Chrome不支持小于12像素的字体，小于12像素的字体都会被固定为12像素（如果字体像素为0的时候，字体消失，也不再占有文本显示的空间），如果要设置小于12像素的字体，可以用CSS3的transform属性的scale值来进行缩放，在不支持CSS3的浏览器中，只有使用图片来展示文本了。

1. 当相邻几个元素的display属性设置为inline-block的时候在什么情况下元素间会出现间隙，可以用哪些方法来消除间隙（当字体系列不确定的时候）？

<解答参考>：

当这些元素间产生空格或出现换行符的时候，这些元素间就会出现间隙，这些元素间的间隙受字体系列和字体的大小影响。

解决元素之间的间隙主要有两种方案（包括这两种方案的分支方案）：

第一，通过CSS将这些元素的父级元素的字体大小设置为0像素，再根据需要给这些元素设置指定的字体大小（因为父级字体大小会继承）

第二，将每个元素的开始标签紧贴上一个元素的结束标签（但是这样会让标签结构的可读性收到一定的破坏）

1. 怎么解决父元素内首个子元素margin-top会把父级元素强制移位的问题？

<解答参考>：

主要有以下解决方案：

* 给父级元素设置overflow:hidden属性
* 给父级元素设置一个1像素的透明边框（border-top:1px solid transparent）
* 用父元素的padding-top值来替代子元素的margin-top值
* 给父级元素设置CSS3的伪元素：

parentElement:before {

content: "";

display: table;

}

* 父级元素设置float:left或者使用绝对定位（不推荐）

1. 使用外链CSS有什么优势？

<解答参考>：

详情参考课件<http://aulence.com/>中CSS分类课程第一章“CSS简述及引用”中的最后一节“通过<link>标签引用CSS文件”。

1. 知道CSS Sprites吗？这样做优缺点是什么？

<解答参考>：

是一种多图标整合为同一张图片的技术手段。

优点：减少HTTP请求，减少图片总传输体积和磁盘占用体积，但在HTTP2.0普及之后，HTTP请求的问题将不再是问题。

缺点：创建、维护和修改起来不便，增加开发的时间成本

可通过命令行下载“cnpm i -g jqueryui”,下载完成后打开“index.html”文件查看示例文件，并打开“images”文件夹查看那些多个图标组成的图片，这个就是CSS Sprites。可通过在线或一些客户端生成，也可以利用PS等图形工具手动去获取CSS Sprites内的图形backgroud-positon的坐标和尺寸。

1. 怎样去处理浏览器的兼容性问题？

<解答参考>：

这是一个很难一概而论的问题，需要对不同情况分别作答：

HTML中：要有基本的文档声明，标签仍然使用HTML5的语义化标签，但像mark、meter、progress、code这一类带有效果或功能性标签，则需要配合CSS做一些[Hack](http://blog.csdn.net/freshlover/article/details/12132801)，对于用于功能的，还得需要通过条件注释（“Hack”提供超链接的文章内有详细讲解）引用单独的JS文件来实现等价的功能。考虑到XHTML（严格模式的HTML，另外还有个DHTML，叫做动态的HTML，其实指的就是通过AJAX手段实现的动态HTML）项目，一般单标签需要在标签结束前加上“/”，如<meta />、<input />、<img />等，<link>标签需要加上“rel”和”type”属性指明文件类型（<script>标签只需加上“type”属性）。如果是中文网站，<html>标签还需加上值为“zh-CN”的“lang”属性，<meta charset=”UTF-8”/>也是必要的。

CSS中：首先就是对网页内会用到的HTML标签做和其表现一致的样式初始化，像Bootstrap那样，设置header/main/footer/nav等标签为“block”，而多媒体标签video/audio等标签为“inline-block”（本来都是“inline”类型的）。对于一些不影响功能而只影响外观的，仍然可以使用CSS3的代码做外观，如圆角、阴影等，对于背景色过渡则使用1像素的图片替代，做X或Y轴的线性重复。如果是布局上的问题，仍然使用CSS Hack来调整差异。

JS中：现在的流行库（jQuery）和三大主流JS框架（vue/react/angular，但他们对IE的支持都很有限，IE8之前更是一概不支持）都做好了兼容性处理，所以一般使用框架开发都不必太担心兼容性问题，如果使用ES6（ECMAScript 2015）开发，则可以用Babel来搞定。如果一定要用原生JS来开发，需要处理兼容性的话常用的手段有：

* 逻辑表达式赋值，如：var event = window.event || event;
* 三元表达式赋值：如：var Array.reduce = Array.reduce ? Array.reduce : Array.reduce(function(callback){ ... callback(); ... });
* 条件判断语句处理，用于处理当前浏览器支持情况提示，是一种比较无奈时的选择： if(!window.localStrage) { alert(“对不起！您当前的浏览器版本太久，导致网站功能无法正常运行，请更换谷歌或火狐等主流浏览器”); return false;}

另外，也可以通过String.indexOf(“Mobile”)、String.indexOf(“iPhone”)、String.indexOf(“Android”)等方法，配合浏览器的“navigator.userAgent”属性来判断是PC端还是移动端，移动端是什么系统的设备、以及浏览器的版本好来做一些兼容性处理和提示。，代码实例请访问[HTML5在线课件](https://www.aulence.com/)JavaScript分类课程“JavaScript BOM”一章的“navigator对象”一节的内容。

1. 在项目需求不确定的时候，怎么去自主确定兼容模式？

<解答参考>：

这需要看项目的类型，如果项目是需要体现出科技感，对视觉交互要求较高的展示型页面，并且用户都是一些专业用户时，可以选择一些较新版本的技术（如HTML5、CSS3或者现代JS框架），如果是移动端，也是一样的，因为移动端现在对新技术的支持都非常好。

而在开发一些大众型的项目，也就是面向广大网民的项目时，就需要选用一些能兼容低版本浏览器的技术和使用一些支持低版本浏览器的UI库和JS库来提升项目的开发效率，如Bootstrap 3、jQuery 1.x等。

1. 分别讲讲HTML、CSS和JS的代码优化，除此之外还需要做哪些额外优化？

<解答参考>：

HTML：减少标签的数量、避免空标签和不必要的嵌套，使用HTML5语义化标签，设置详细的meta信息来完善SEO。当页面内容过多时，可以考虑“懒加载”技术（配置事件驱动AJAX的异步加载显示技术）。如果是不考虑兼容性，还可以使用“[Link Prefetching API](http://blog.csdn.net/typename/article/details/38358765)”来进行指定的预加载（通过<link rel="prefech" href=URL>来实现的预加载）

CSS：代码分主次、分层次的归类型管理（可以利用Less/SASS的一些特性），在对页面中大量出现的样式和背景效果使用群组选择器管理，不同的选择器哪怕使用同一张图片，如果不用群组选择器管理也会触发多次HTTP请求。另外，就是遵循低权重原则，提高代码的可维护性。在不使用第三方UI库的情况下，如果条件允许，使用Less/SASS，对公共的CSS文件甚至可以分离为CSS初始化、CSS元件样式、CSS细节处理、CSS媒体查询、CSS动画为几个不同的.less/.sass文件，并导入（@import）到入口文件（如“common.less”），通过打包工具（主流的webpack）+编译工具（编辑器自带、“考拉”等）+资源镜像（source-map）的方式去实现细致化的管理。上线前开启代码压缩（如果已经搭建好了以上所提到的自动化开发环境，则可以一开始就让代码压缩，省去后面重构URL的工作）

JS：利用MVC（MVVM）思想，按功能职责划分JS文件，提高代码的可复用性。对于项目中大量会使用到的功能，可以考虑封装为简单的方法或插件，便于后续或其它项目使用（如弹出框插件、日历插件、数据有效性验证、搜索框插件等）。尽可能地避免对DOM的重排（回流）和重绘（重构）。对于代码的模块化管理，仍然可以使用Webpack去实现，并用其实现代码的压缩。当然，如果单纯地实现压缩，使用YUICompress插件安装进IDE或Gulp（Grunt有点老了，相比Gulp劣势明显）会更方便一些。当然，如果团队对这些新技术不了解，项目时间上又不允许去进行学习和熟悉，那就应该在项目初期就要约定好一些规范。如：不允许出现全局变量，除非经过协商一致同意的少量变量，否则都应该有自己的“命名空间”，分类管理全局空间的对象变量。使用“面向对象编程”（OOP）思想将功能抽象为对象个体的方式去进行管理，每个对象个体推荐使用构造函数的形式去进行实例化的封装和继承，以提升代码的可维护性和可扩展性。

除此之外，还需要做HTTP请求数量的优化，尽量将多个请求合并为单个请求，如将多个CSS文件、JS文件合并为一个文件，使用CSS Sprites合并页面上的图形，服务器端配置好文件的缓存，使用CDN（内容分发网络）来减少自己公司服务器的负担。配合AJAX和scroll事件实现图像资源的懒加载。

1. 简谈CSS的盒子模型

<解答参考>：

CSS盒子模型主要分为两种：

* content-box，属于现在浏览器采用的标准盒子模型，在CSS中设置的width只表示其内容（content）的宽度，padding和border会增加额外的实际宽度
* border-box，属于IE6的盒子模型，但被推荐转换为这种模式使用，因为CSS设置的width就是其本身实际的宽度，设置的border和padding都会“压缩”内容显示的空间，这种模式有利于做一些精确的像素或百分比布局，而不需额外去计算padding和border的线宽。现今主流的UI库（主要是PC）都是用这种模式，如Bootstrap

CSS3中还出现了弹性盒子模型“flex-box”，不过在分类上稍许不同，但却可以忽略以上两种盒子模型的设定，按照空间划分来实现布局。现今的主流移动端UI库大多使用这种模式。

1. 为什么要将样式文件（CSS）放在head标签内，而将脚本文件（JavaScript）文件放在页尾的body结束标签前？

<解答参考>：

因为浏览器的默认加载执行顺序是从上到下的（谷歌等现代浏览器下载的时候是并行下载），将样式文件放在head标签内是可以让样式表首先被加载，防止用户先看到散乱布局的界面。而将脚本文件放置在body结束标签之前（而不是body结束标签之后）一方面是W3C规范的要求，另外一方面是脚本加载和解释执行比较耗时，如果放在整个文档之前（如也放在head标签内部），脚本的解释执行会阻塞后续文档内容的加载渲染，这样会让页面出现很长的“白屏”期，影响用户体验。

1. CSS中的em和rem这两个单位有什么异同？实际项目开发中你会选择哪个？为什么？

<解答参考>：

em和rem都主要是用于字体衡量的单位，区别在于em的大小是由最邻近含有字体大小（font-size）设置的父级元素决定的。而rem的大小是有设置在html标签上的字体大小决定的，如果html标签没有设置字体大小，则由浏览器的默认字体大小决定，现代浏览器的默认字体大小通常为16像素。

在实际开发中我更偏向于使用rem这个单位，尤其是在移动端，但如果项目需要兼容IE浏览器系列或者一些其它低版本的浏览器，则会选择em作为单位，因为rem不支持IE 10、FireFox 9及以下版本的浏览器

1. 定位（position）主要有几种方式，各自的特点是什么？

<解答参考>：

主要有三种（不能省掉“主要”二字）。

默认定位方式为“static”，采用默认文档流布局

relative：相对定位，不会脱离文档流，相对当前的位置做偏移定位

absolute：绝对定位，会脱离文档流，以最邻近采用非static定位的父级元素为参考定位，若父级中没有非static定位的元素，则是相对于整个document文档进行定位（不是body，因为body有时会受自身或其第一个子元素的margin-top属性影响开始位置）

fixed：固定定位，会脱离文档流，当对于浏览器显示窗口定位，位置与其所在的元素父级没有关系，唯一的参考物就是浏览器窗口，所以它不会随文档内容的滚动而滚动。

1. 什么时候需要清除浮动影响？都有些什么清除浮动的办法

<解答参考>：

当采用浮动布局的元素的父级元素没有高度，并且影响了其后元素的布局的时候需要做清除浮动的影响。

详情可到在线课件CSS分类课程“CSS显示与定位”一章中“清除浮动副作用”一节中查看。

总体来说，最简便的方式就是给父级元素添加overflow属性，值为“hidden”(推荐)或“auto”均为，但这种方式不能用于带有隐藏菜单的元素。而通过设置:after来替代一个空标签来进行清除的手段虽说麻烦，但能应付的情况是比较多的，唯一的缺陷就是对IE老版本兼容问题。实际使用起来可以通过CSS Hack或JS判断去实现带有清除浮动样式的空标签的插入。

1. 让一个元素在不干扰文档流的情况下在浏览器窗口内水平和垂直居中

<解答参考>：

同样都使用“position: fixed”(或“absolute”)，后续代码分为几种情况：

* 已知元素宽高，方法一：使用{ top: 50%; left: 50%; margin-top -1/2px; margin-left: -1/2px; }（“-1/2px”为元素实际宽或高的一半）
* 已知元素宽高，方法二：使用{ left: 0; top: 0; bottom: 0; right: 0; margin: auto;}
* 未知元素宽高，使用CSS3的方式{ top: 50%; left: 50%; transform: translate(-50%,-50%);}
* 未知元素宽高，兼容老版本IE，{ top: 50%; left: 50%;}，然后通过通过JS获取负值的outerWidth和outerHeight的一半的像素值，作为“margin-left”和“margin-top”的值。

1. display主要有哪些属性值，他们有什么区别

<解答参考>：

参考在线课件CSS分类课程“CSS显示与定位”一章中“元素的显示方式display”一节。

回答重点在于它们在文档流中的布局占用方式。

1. JS有哪些数据类型？null、NaN、undefined的区别是什么？

<解答参考>：

基本数据类型：

number、string、boolean、null、NaN（这里注意大小写）、undefined

引用数据类型（复合型、地址型）：

object

null：表示一个对象为空（document的标签节点也是一个对象）

NaN：是尝试将非number的字面量、变量或表达式转换为数字的时候返回的值

undefined：当一个变量已经声明，但是未赋值的时候返回的值。（<注>如果是未声明变量就直接带入表达式，则会产生未定义的报错）

这种题还有一个问法，就是问typeof()之后的结果，或者是JavaScript的数据值类型有哪些。如下：

number、string、boolean、object、function、undefined

要注意区分

1. JS函数内的this是什么，有哪些情况？怎么改变this的指向？

<解答参考>：

* this出现在一个全局函数内部，this就是window对象。
* this出现在一个事件的匿名函数内部，那this指向的就是触发该事件的这个元素节点（如果该元素含有多层子节点，this同样指向绑定该事件的那个节点，如果想获取实际点击的子节点，应该用事件对象e.target去做判断，“e”为该匿名函数的事件对象别名）
* this出现在一个对象的普通属性内，this的指向仍然是window
* this出现在一个对象的方法内，this的指向是该对象本身
* this出现在一个构造函数内，那this指向是该构造函数被通过new关键字实例化后的那个对象，否则仍然指向window

改变this的指向，可以通过function的原型链提供的三个方法：call/apply/bind来改变，详情参看[《JavaScript标准参考教程（alpha）》](http://javascript.ruanyifeng.com/oop/this.html)的“this关键字”一章，提到这三个方法的在第4节。

1. JS里面事件监听和普通事件的区别

<解答参考>：

事件监听 —— Element.addEventListener("handle", function(){...}, boolean)可以为同一个元素节点（Element）添加同一个事件，而不会覆盖之前的同名事件。

普通事件 —— Element.onHandle(function() {...})如果重复绑定，那按照执行顺序，后绑定的会覆盖先绑定的同名事件。

除此之外，有些事件只能通过事件监听来绑定，如火狐的“DOMMouseScroll”事件，这个事件在其它浏览器（包括IE）里叫做“onmousewheel”，是鼠标滚轮触发的事件，和onscroll的区别在于，“onmosewheel”事件不管是否浏览器出现了滚动条都会执行，而且可以通过事件对象的e.wheelDelta返回的正负值判断滚动方向（上还是下），各浏览器存在具体数值和正负的差异，详情[参考](http://www.aseoe.com/show-10-464-1.html)。

1. JS中的eval方法有什么用？它存在什么问题？

<解答参考>：

eval方法能将字符串编译为JavaScript表达式来执行，但是由于以下原因，该方法不推荐使用：

* eval执行的“字符串代码”很难被调试
* 旧版浏览器对其执行的效率极其低下
* 被压缩后的代码会让他在执行的时候报错
* 容易被执行XSS攻击，安全性没有保障

1. 怎么用原生JS实现一个事件代理（当页面事件元素是后面动态生成的时候）

<解答参考>：

var tableID = document.getElementById("tableID");

tableID.onclick = function(e) {

// 这里的tagName也可以是id或className等元素标识符

if(e.target.tagName === "TD") {

// 做要做的事

}

}

1. 怎么阻止事件冒泡和浏览器默认事件

<解答参考>：

阻止冒泡：

// W3C标准

event.stopPropagation();

// IE浏览器

event.cancelBubble = true；

阻止默认事件：

// W3C标准

event.preventDefault();

// IE浏览器

event.returnValue = false;

1. JS里的prototype是什么，有什么用？

<解答参考>：

prototype是JS里的原型链对象，JS里的所有方法都是通过原型链的Object继承而来，除了JS内置的那些方法，我们还可以通过原型链来扩展一些属性和方法，而且这些方法还可以通过prototype之前的类型（JS的六种数据值类型）来限制方法的类型使用范围。

扩展

prototype主要是用于扩展构造函数的

\_\_prop\_\_主要用于扩展对象的，当然也能扩展函数

1. 什么叫做闭包，它有什么特点？闭包存在什么问题？

<解答参考>：

从结构上来讲，闭包就函数内包含函数的结构

从表现上来讲，闭包函数在被赋值执行后，其内部的变量将不会被销毁。每次调用闭包函数，其内部的变量都会保留上一次执行的结果。

由于闭包对象会保留在内存中，不会被浏览器垃圾回收机制回收，在项目中大量的使用会造成性能问题，多者会造成内存泄露，产生严重的程序错误。

1. 怎么用JS实现一个类，应该怎么实例化这个类？

<解答参考>：

如例：

function RoleInfo(name,skill) {

this.name = name;

this.skill = skill;

this.showInfo = function() {

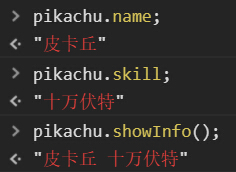
return this.name + " " + this.skill;

}

}

var pikachu = new RoleInfo("皮卡丘","十万伏特");

运行结果如图：



如果改成ES6的Class的形式话，应该写成：

Class RoleInfo(name,skill) {

constructor(name , skill ) {

this.name = name;

this.skill = skill;

}

showInfoMethod() {

return this.name + " " + this.skill;

}

get showInfo() {

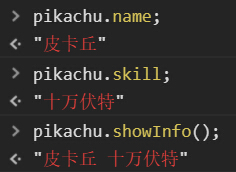
retrun this.showInfoMethod();

}

}

// 实例化的方式和调用的方式则是一样的

var pikachu = new RoleInfo("皮卡丘","十万伏特");



1. 知道Web Storage吗？谈谈cookie、SessionStorage和LocalStorage的异同，它们都有一些什么用途？

<解答参考>：

Web Storage是HTML5后对Window对象新增的API，是用于实现浏览器的数据持久化的技术，在浏览器开发者工具里面可以查看到具体的数据键值对。

cookie是存储在浏览器端，并且随浏览器的请求一起发送到服务器端的，它有一定的过期时间，到了过期时间自动会消失。一般只有5kb左右的浏览器存储空间。

SessionStorage和LocalStorage都能对数据进行存储，但是它们都只能以字符串的方式进行存储，需要用JSON.parse和JSON.stringify这两个方法来进行转换操作。

不同的域名有不同的Web Storage空间，这个空间根据浏览器的不同各有差别，以谷歌浏览器为例，大概的存储空间是5MB左右。(www.baidu.com、zhidao.baidu.com、music.baidu.com等认为是同一个域名，所以公用一个Storage空间）

它们的区别在于，SessionStorage存储的数据在关闭浏览器之后就会消失，而浏览器的刷新和历史记录的变化都不会对数据造成影响，

而LocalStorage哪怕在浏览器关闭后数据仍然保留，除非用clear方法或在浏览器开发者工具内手动清除。

SessionStorage用于页面间的传值非常方便，可以记录一些交互数据、计数器等，也可以用来记录一些用户状态信息。

LocalStorage可以用来记录一些用户的非敏感信息，如用户昵称。（天猫商城就是用LocalStorage来记录用户昵称，这样下次用户在未登录的情况下也能向用户显示类似“亲爱的xxx，欢迎您，请先登录”这样的欢迎词）

<localStorage存储空间检测>

if(window.localStorage) {

var remainder = 1024 \* 1024 \* 5 - unescape(encodeURIComponent(JSON.stringify(localStorage))).length;

console.log("空间为：" + remainder + "个字节");

console.log("约等于：" + Math.round(remainder/1024) + "kb");

}

1. 谈谈你对DOM中重构（重绘）、回流（重排）的理解

<解答参考>：

浏览器的重构指的是改变每个元素外观时所触发的浏览器行为，比如颜色，背景等样式发生了改变而进行的重新构造新外观的过程。重构不会引发页面的重新布局，不一定伴随着回流，回流指的是浏览器为了重新渲染页面的需要而进行的重新计算元素的几何大小和位置的，他的开销是非常大的，回流可以理解为渲染树需要重新进行计算，一般最好触发元素的重构，避免元素的回流；比如通过通过添加类来添加css样式，而不是直接在DOM上设置，当需要操作某一块元素时候，最好使其脱离文档流，这样就不会引起回流了，比如设置position：absolute或者fixed，或者display：none，等操作结束后在显示。

1. 知道XSS和CSRF吗，怎么防御？

<解答参考>：

XSS攻击（又称CSS，Cross SiteScript，跨域脚本攻击）。其原理是攻击者向有XSS漏洞的网站中输入(传入)恶意的HTML代码，当其它用户浏览该网站时，这段HTML代码会自动执行，从而达到攻击的目的。如，盗取用户Cookie、破坏页面结构、重定向到其它网站等。

防止XSS攻击的方案主要就是用正则表达式严格过滤用户的输入。其它的就交给服务器端那面处理了。

CSRF（Cross-site request forgery，跨域请求伪造）。CSRF能够做的事情包括：以你名义发送邮件，发消息，盗取你的账号，甚至于购买商品，虚拟货币转账......造成的问题包括：个人隐私泄露以及财产安全。

防止CSRF主要的手段是使用图形验证码（不过这些都是服务器端干的事）

* ES6相关的问题

1. 我们一直用来声明变量的var和现在ES6中的let、const有什么区别？

<解答参考>：

var声明一个变量和想let、const相比最大的特定就是，var可以无限次的声明已经存在的变量，而let和const在声明已经存在的变量会报错，另外一个特点的就是var有变量提升，而let和const没有，let和const不能提升变量而造成的变量后定义报错这种现象叫做“临时死区”。

let用于声明变量，其值可以被修改。

const用于声明一个常量，其值不能被修改。但是复合型的数据内部的数据发生变化JS引擎并不认为是该数据发生变化。

2. 在常规的数据声明中，let和const哪个推荐更多的使用？为什么？

<解答参考>：

const被更多的使用。因为const在很多不希望被修改的数据在被修改时JS都会抛出错误，可以有效确保数据的安全性。对象实例基本都使用const来声明的。

3. ES6中的箭头函数和以前的普通函数有什么区别？

<解答参考>：

箭头函数和普通函数最大的区别就是，箭头函数并不会改变整个父级作用域内的this指向，而普通函数的this指向会根据语境有所差异。

4. 怎么用ES6里的class；来定义一个类？

<解答参考>：

参考地址：

[http://es6.ruanyifeng.com/#docs/class](http://es6.ruanyifeng.com/" \l "docs/class)

5. Promise是什么，它能处理什么样的问题？

<解答参考>：

参考地址：

1、<https://segmentfault.com/a/1190000006708151>

2、<http://es6.ruanyifeng.com/#docs/promise>

* 与AJAX相关的问题？

1. AJAX是什么？有什么用？原生JS怎么实现一次完整是AJAX请求

<解答参考>：

AJAX（Asynchronous JavaScript and XML，异步的JavaScript和XML）不是新的编程语言，而是一种使用现有标准的新方法。它最大的优点是在不重新加载整个页面的情况下，可以与服务器交换数据并更新部分网页内容。

详情参考[HTML5在线课件](https://www.aulence.com/)JavaScript课程分类“JavaScript AJAX”一章，注意理清基本的四个步骤。

1. 怎么处理多个AJAX请求异步中代码执行的次序

<解答参考>：

传统的做法是在一个AJAX请求回调函数成功的判断里嵌入接下来要做的AJAX，以此来对AJAX请求的顺序进行请求执行的排序，而在ES6之后可以使用promise对象来进行处理。而现在的主流的框架和库都采用了这种模式，而其中比较著名的AJAX插件[Axios](https://www.awesomes.cn/repo/mzabriskie/axios)正是这种promise模式的代表作。

如果使用的是Axios，如果被追问为什么使用它，它的亮点是什么，可以从这几点回答：

* + 较高的社区活跃度（github上星数和angular不相上下）
  + 无任何依赖（不需对任何库和框架进行依赖，任何Web项目都能直接使用）
  + 完善的功能（包含创建和修改日期等实用信息）
  + 支持并发请求（同时请求几个接口，然后依次处理）
  + 和现在主流的AJAX插件一样，支持promise模式（也就是异步的依次处理）
  + GitHub上完善的使用文档
  + 受到技术大牛的推荐（如Vue.js作者尤雨溪）

1. AJAX中的GET和POST两种方式有什么区别

<解答参考>：

参考[HTML5在线课件](https://www.aulence.com/)JavaScript分类课程“JavaScript AJAX”一章的“GET和POST对比”一节的内容

1. 什么叫做“浏览器的同源策略”？什么叫做跨域？怎么解决AJAX请求跨域的问题。

<解答参考>：

浏览器的同源策略是浏览器的一种安全策略，指数据的请求访问都只能通过同一个域下进行，不同的域则不能直接进行数据的请求访问。

以下四点的不同，都已经表示当前的URL产生了跨域：

* + 协议不一致（http/https/file）
  + 端口号不一致（:8080和8081是两个不同端口号）
  + 域名不一致，包括主域名和子域名不一致（www.baidu.com和www.google.com是不一样的域名，www.baidu.com和zhidao.baidu.com也不是一样的）
  + 域名格式（又称：域名类型）不一致（如：.com、.cn、.org、.io等都不是一个类型）

而解决跨域的方案现在比较多，最为主流的有两种，请参考[这篇文章](https://segmentfault.com/a/1190000005082202)。

而比较推荐的是CORS这种方案。这种方案只需要服务器端手动指定域名的白名单即可，前端不需要做任何多余的操作，属于效率最高的一种方案。

1. 在向服务器端请求数据时，readyStatus返回的200+、300+、400+和500+状态码分别代表什么？

<解答参考>：

200+：请求成功

300+：重定向（进入到一个URL后发生的二次跳转）

400+：请求失败（可能是路径名错误或者权限不满足）

500+：服务器内部错误（包括网路问题）

* 谈谈对HTML5的理解。知道什么叫做XHTML，什么叫做DHTML吗？

<解答参考>：

HTML5并不是一个新的技术，也不是一次“改革”，而是一次标准的升级，它带来的并不只是HTML里面那些简单的标签，它还带来一系列的API（Application Programming Interface，译为：应用程序编程接口），可以通过[HTML5在线课件](https://www.aulence.com/)“JavaScript”分类课程“HTML5 API”中查看简要的API介绍。

这些API很大程度地扩充了JavaScript的功能性，HTML5的出现让前端开发者能做出不逊色于原生的应用。传统的即时通讯现在用Websocket和WebRTC这些API也能办到。

这里的提问最主要的考核点是试探你是否了解HTML5的基本概念，因为很多人（包括知识面狭窄的前端开发者）认为HTML5只是新增了一些语义化和功能标签而已。

XHTML就是指严格的HTML，严格的HTML标签必须是闭合的，单标签也应该在结束标签前加上“/”，如：<meta />、<img />、<input />等，link标签和script标签都必须有type属性指定文件类型，script标签还必须有lang属性来指定所用脚本语言。

DHTML是Dynamic HTML的简称，就是动态的html，是相对传统的静态的html而言的一种制作网页的概念。所谓动态HTML，其实并不是一门新的语言，它只是HTML、CSS和客户端脚本的一种集成，即一个页面中包括html+css+javascript（或其它客户端脚本），其中css和客户端脚本是直接在页面上写而不是链接上相关文件。DHTML不是一种技术、标准或规范，只是一种将目前已有的网页技术、语言标准整合运用，制作出能在下载后仍然能实时变换页面元素效果的网页设计概念。

* 如果不使用任何框架，怎么去做响应式的布局

<解答参考>：

使用媒体查询（@media）+流式布局（即百分比布局）的形式实现多分辨率的响应式布局。

如果只是小规模的响应式调整，可以直接在CSS文件里通过将”@media { /\* 需要覆盖的选择器及需要修改的属性 \*/ }”放在文件末尾（这样无需特意的去提高权重）来实现响应式布局

但如果网站的需要不同的分辨率端（PC、Pad、Mobile）实现几乎完全不一样的布局及风格，那则需要利用<link>标签的“media”属性来实现（见文件内附带的“example”示例演示）

如果需要文字能根据分辨率也进行调整，比较好的一种做法就是给body设置一个固定像素，而和其有继承关系（表单元素不会继承body的字体）的字体都采用“em”作为字体大小的单位。这样一来，当网站需要整体调整字号的时候，只需调整body的字体像素，那整个网站的非表单元素的字体大小也会进行等比调整。

* 掌握哪些CSS和JS框架的使用？

CSS框架：

1. Bootstrap

<解答参考>：

Bootstrap可以说是UI库中最具有统治地位的CSS框架，它对之前已经出现的UI库和其后出现的UI库都产生了非常大的影响，它改变了开发者们对CSS的思考方式，可以说是一款意义非常大的CSS框架。而且现在Bootstrap 4.0的预览版已经发布，它改用SASS编写（原Less），在设计上不断地去附和人们的赏美观，同时也在做着一些引领，而且还在不断进步中。

参与Bootstrap设计或赞助的，不乏很多前端领域世界知名的人物，如尼古拉斯·泽卡斯，整个Bootstrap对于CSS样式初始化部分的代码都是他写的，单独使用则是一个名为[normalize.css](http://necolas.github.io/normalize.css/)的文件，可以说这个初始化文件已经成为Web前端开发者对于样式初始化的一个标本了。

1. ionic

<解答参考>：

它和Bootstrap一样，通过UI库（CSS部分）加上对AngularJS二次封装的JS部分组成了一个CSS框架，其设计思维借鉴于Bootstrap，但它本身是一个纯粹针对移动端的CSS框架，没有PC端的版本，虽然AngularJS现在已经逐渐走下神坛，最新的4.0版本在性能、易用性以及学习成本上相比Vue和React都处于下风，但是用ionic的UI库来开发手机端的界面也是一种不错的选择。[<详见>](http://www.ionic.wang/css_doc-index.html)

1. mui

<解答参考>：

是国内DCloud公司旗下的知名前端IDE HBuilder内置的一款手机端的CSS框架，它和现有的移动端CSS框架大同小异。由于国别的关系，学习成本有所下降。如果使用的IDE是HBuilder，那使用mui来进行手机端的应用开发可谓是“行云流水”。[<详见>](http://dev.dcloud.net.cn/mui/ui/)

1. AmazeUI

<解答参考>：

AmazeUI同样是一个国内的CSS框架，它分为PC端（Web）和移动端（Touch）两个版本，它的官方其中一条标语是“站在巨人的肩膀上”，可见其设计模式和现在主流的UI库也没有太大程度的差别。要说它的特点，个人认为它的美观度还是比较不错的。

它的PC端版本（[<详见>](http://amazeui.org/getting-started)）依赖jQuery来组成，而移动端（[<详见>](http://t.amazeui.org/" \l "/?_k=icwoae)）则是基于React去实现。

1. ElementUI

<解答参考>：

Element UI 是一套采用 Vue 2.0 作为基础框架实现的CSS组件库，它面向企业级的后台应用，能够帮助你快速地搭建网站，极大地减少研发的人力与时间成本。它是国内新兴知名企业“饿了么”的Web前端团队开发的。

它和之前提到的CSS框架有所区别，因为它是基于Vue的组件化的基础上进行封装的，其UI库部分的表现非常精美，如果对Vue.js比较了解的话，这个组件库一旦熟悉使用起来的确会给人一种“神清气爽”的感觉，也是一个非常有潜力的CSS组件库。[<详见>](http://element.eleme.io/" \l "/zh-CN/guide/design)

1. jQueryUI

<解答参考>：

这是一个还未“火过”就已经离开的PC端UI库，虽然说他是jQuery的“官方作品”，但由于其本身的一些缺点，如：体积过大，维护困难，性能低下，美感欠缺等问题，jQuery官方团队已经在好几年前就已经停止了对其更新，不过在jQuery的官方链接上仍能看到它：<https://jqueryui.com/>

也可通过命令行下载“cnpm i -g jqueryui”,下载完成后打开“index.html”文件查看示例文件

1. jQuery Mobile

<解答参考>：

一个同样比较失败的jQuery官方出品，和jQuery UI一样，同样存在着体积过大的问题（用现在的话来说，就是jQuery Mobile里面存在着很多“过度设计”），关键是这样的体积并没有带来等比的功能实现，甚至可以说是简陋（当然，这里说的是功能而不是外观）。但不可否认的是，它的确是HTML5概念中从PC走向移动端的“先河”，很多早期就接触移动端开发的互联网公司对jQuery Mobile都不会陌生。到现在，仍然还有部分公司还在使用它。

但如果是现在新启动一个项目，还去使用jQuery Mobile的话，就显得有些冥顽不灵了。除非等到它脱胎换骨的那一天。

CSS预处理语言：

<解答参考>：

指的就是Less、Sass/SCSS，还有国内不太常用的Stylus，它们的设计思想都在互相借鉴，之间的语法差异也越来越趋同，所以只要熟练使用其中之一就好。

这里就拿Less来说，它的核心特征有：

* + 变量：和JS一样，有全局和局部变量之分，作用域的理论也基本和JS一样
  + 混合：就是将一组样式进行混合（可理解为样式集或样式打包），然后供其它选择器调用，还可以支持参数的设置
  + 嵌套：就是将有父子关系的元素通过嵌套的结构来进行表现和设置
  + 函数运算：如一些简单的算术运算，颜色值的运算等
  + 导入：和CSS不同，Less/SASS的@import导入文件发生在开发阶段，而真正使用的CSS文件是在编译阶段之后，所以说不会有CSS使用@import那样的执行时间顺序问题。使用导入功能，可以让样式表文件分离管理，降低耦合，从而提升代码的可维护性

另外，需要知道SCSS是Sass的最新版本第三版的语法，它更易用更“先进”一些，所包含的函数和一些处理机制都更为丰富，算是同类CSS预处理语言中特性最为丰富的一个了。甚至有知名组织已经在建议将SCSS列为浏览器端默认支持的样式设置语言，Sass中文官方地址：<https://www.sass.hk/>

JS框架：

1. jQuery

<解答参考>：

前端必会、必精的一个JavaScript类库，它的本质是对原生JavaScript的操作进行更加简化和易用的封装，让书写的代码更少，更利于阅读理解，在很大程度上也减少了代码的复杂性。

1. AngularJS和Angular

<解答参考>：

AngularJS是Angular的第一版本，诞生于2009年，是由一个Java开发者创建的（所以AngularJS有很多Java的编程思想），由于其新颖的语法模式和结构，以及模块化开发方案产生的高可维护性等特征，其后被Google正式收购，并在2010年10月正式发布。是一种MVC（其表现更像是MVVM）模式的框架，其模块化方案、模版插值、标签指令、双向数据绑定、依赖注入的特点被后面很多JS框架效仿。

AngularJS使用脏值检测机制，使得watcher越来越多的时候，整个程序都会变得越来越慢。这迫使开发者为了解决其性能问题不得不使用一些更“高深”的技术来处理这个问题，这使得代码的逻辑性变得复杂，代码量明显增多，开发时间成本也等比增加。因为这一系列的原因，Google在2014年9月停止了对AngularJS的开发，并发布了Angular2.0（注意Angular命名后面没有JS了）。

Angular是Google从第2版本开始的命名，目前最新版本号是5，（Google在2018年3月份发布第6版），他们几乎将AngularJS进行了重写，其核心内容的修改达幅度到80%以上，并且采用微软的TypeScript作为开发语言，并且可以使用npm来快速搭建脚手架，几乎成为了一个新的JS语言框架。受当时React的启发，Angular的核心概念也是“组件化”，且删除了AngularJS其最核心的$scope以及ng-controller指令。2.0的依赖注入也发生了演变，和Java的Spring框架中的依赖注入非常相似了。

另外Angular2.0开始，可以使用Webpack作为构建工具了。

1. React

<解答参考>：

React 起源于 Facebook 的内部项目，因为该公司对市场上所有 JavaScript MVC 框架，都不满意，就决定自己写一套，用来架设Instagram 的网站。做出来以后，发现这套东西很好用，就在2013年5月开源了。

React并不是一款MV\*模式的框架，它只关注视图层（view）的东西，它使用了一种叫做JSX的语法，就是一种将HTML/CSS/JavaScript写在一起的语法，这种语法满足软件设计模式“高内聚、低耦合”的要求。它通过对Virtual DOM进行操作，并将操作结果使用diff算法渲染到视图上，所以性能比较高。

它所采用的开发模式是flux模式，就是一种单向数据流的模式。区别于现在三大主流框架的另外两个Angular和Vue，它不支持双向数据绑定。现在出现了一种Flux数据操作模式的升级优化版的实现——Redux，它使得Flux模式变得更简单。

组件化编程思维是其核心的编程思维模式。

1. Vue

<解答参考>：

vue.js是由国内的大牛尤雨溪为主要开发的，它起初是个人项目，在2014年初开源就受到广泛关注。起初尤大神把它定位是一个视图库而非框架，但随着vue-router，vue-resource（现已废弃，推荐使用axios.js）等第三方库的发展，Vue.js逐渐成长为一个框架。Vue.js真正火热起来，并井喷式的发展是在2016年5月的2.0版本发布之后。

是一个标准的MVVM框架，其核心仍然也只是关注视图层，区别于传统JS的事件驱动模式，和Angular、React一样都是采用数据驱动模式，并且和Angular一样支持双向数据绑定，其核心开发模式也是采用组件化编程思维。

由于是“后起之秀”，Vue尤其是2.0之后，解决了很多Angular和React存在的缺陷，并且它非常的轻量化，在使用gzip压缩之后只有22kb，从知名网站多方面的性能测试结果来看，其运行速率也是三大主流框架之首。Less is more（大道至简）是Vue.js所追求的目标。

* 区别何为插件，何为库，何为框架

<解答参考>：

* 插件通常为完成单一功能的实现，如图片轮播插件、日历插件、弹出框插件等
* 库是对原生JavaScript进行功能简化或通过现有属性或方法的组合封装，其目的是通过更简洁、更易读、更易懂的代码来完成页面功能，其代码形式比较自由
* 框架按照一定的语法模式对原生JavaScript“重新设计”，开发者使用框架进行开发必须遵循起语法规则和设计思想，在此基础上去完成功能的实现。相比库而言，其代码更为严格，自由度有限，其目的是让代码更加严谨，更利于管理维护
* 对MVC、MVVM、Flux/Redux模式有没有了解，它们都有什么样的特点？

<解答参考>：

* MVC（Model View Controller），是模型－视图－控制器单词的缩写，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。
* Model是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。
* View是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。
* Controller是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。
* MVVM是Model-View-ViewModel的简写。其中的Model和View和MVC模式中的“MV”是一样的，而后面的VM是一个单词表示的是“视图模型”，是一种可以通过视图来控制逻辑的模式。
* Flux模式也是一种软件架构的思想，是一种数据“单向流动”的特殊模式，所以也被称作“单向数据流模式”。可参考这一篇[博客](http://sentsin.com/web/1011.html)帮助理解这种模式。而Redux是一种更简洁友好的Flux模式的实现，现在React通常就是使用这种模式去实现状态管理，前后端来共建数据的。Redux并非一种独立的模式，可以说是对Flux模式的一种进化的实现。
* 会不会使用NodeJS？它有一些什么特点？

<解答参考>：

nodeJS有很多技术模块如用于搭建服务器的Express模块，用于包管理的NPM模块，用于文件流操作的fs模块。只需如实回答自己所了解和已经实际使用过的即可。

* 使用过哪些代码版本管理工具（如：svn、git等）？知道集中式和分布式管理的区别吗？能谈谈使用Git怎么创建一个分支和合并分支吗？

<解答参考>：

有使用过svn（subversion）开发团队项目的经验，其特点是集中式版本管理。

用Git来管理自己的代码版本，使用GitHub来分享自己的作品。其特点是分布式版本管理。

集中式和分布式管理的区别在于：集中式管理只有唯一的一个版本库，而分布式在不同的终端和服务器端都可能存在不同的版本。

使用git创建并切换到分支的的命令是：

git checkout -b dev（相当于是先git branch dev然后再git checkout dev的简写）

使用git何必分支的流程应该是：

git checkout master

git merge dev

<注>dev只是一个分支名，可以用其它合法名称替代。

* 怎么避免代码出现冲突，当代码冲突出现时怎么去解决？

<解答参考>：

避免冲突应该从这几个方面下功夫：

* 使用模块化开发或使用组件化的开发模式（组件化开发是模块化开发的一种很好的实践）
* 合理的分工安排，尽可能避免不同的开发者有机会去修改同一份文件
* 公共代码文件、配置文件应该交由专人维护，其它开发者要修改其内容得经过该文件的维护人的许可。
* 编写一份项目目录结构文档，文档中应该包含每个目录/文件的功能及责任人，当出现新建、删除文件时，该文档也应该尽可能早的实时更新，并知会其他组员
* 减少每次版本合并的时间间隔，一般情况下每次版本的整合的时间间隔不应超过一天

解决冲突：

* 如果产生冲突的两份代码有明显的正误之分（如bug的解决，空行或空格，变量命名纠正，代码格式调整等），可由该文件的负责人直接解决并提交到主分支、主仓库上
* 如果产生冲突的地方较多，可以集合参与该文件修改的相关人员一起核对应该保留的代码，冲突解决后再合并到主仓库上，并让相关人员及时更新到最新版本
* 使用过哪些架构工具？（如：npm、bower、yarn、grunt、gulp、webpack、browserify、Babel等）

<解答参考>：

如实回答即可，千万不要滥竽充数。因为如果面试官恰好挑出一个你不会的东西来，你却一个字都答不上来，那你的整篇简历和面试过程都会受到怀疑。

只要写在简历上的东西，要不就是你会的，要不就是你对齐基本理论的概述能较流利的口述出来（虽然有可能你并不会这个技术）。

* 谈谈Grunt/Gulp和Webpack/Browserify的区别？

<解答参考>：

首先Grunt和Gulp属于一类，Gulp的使用更加直观渐变，比Grunt更优，更新，Webpack和Browserify又属于一类，Webpack现在基本成为了主流，而Browserify配置过于复杂死板老旧，已经不能满足现在的构建要求了。

Gulp这类的工具主要是对工作流程的优化，如压缩、合并代码、样式预处理等，并可以生成一份构建完成后新的文件目录。在用于一些中小型的静态页面展示的项目会比较方便。

而Webpack这类工具的核心是模块化的实现，而压缩代码、合并文件这些都更像是附带的功能，而像CSS预处理语言、ES版本转换、特殊文件（如.vue）的编译都需要依靠单独依赖loader去实现。像html-webpack-plugin这些插件更是为模块化开发模式下的产物组件化开发提供了实现的可能性。

* PC端（桌面）和移动端开发有些什么区别？开发中需要注意一些什么问题？

<解答参考>：

PC端和移动端开发从编码角度上来讲基本上是没有多大的区别的，但是从界面设计和用户体验角度上来讲区别却是比较大的。

PC端采用960~1200像素居中的布局，用户的视觉焦点位于屏幕居中偏左上的位置，是使用鼠标操作，需要用CSS的hover的来提升操作体验，尤其是对于一些无文字的图标按钮都需要在标签上设置title属性来对功能进行补充说明。

PC端当页面较多的时候，导航栏会用二级导航，甚至三级导航栏来补充。大多数用户不太喜欢使用浏览器自带的浏览记录回退按钮，需要较好的引导，如使用“路径导航”（又称“面包屑导航”）来标明页面的位置。

PC端存在多个厂商且不同版本的浏览器，而且PC主机的使用年限一般较长，需要进行很多兼容性的处理。

而移动端的浏览通常是全屏的，标题在顶部，而导航栏在下方。页面的总体设计通常要有适当的留白，以保证交互体验。

移动端导航按钮一般不会太多，多是通过流程化的引导，和左上角的回退来实现多页面的访问的。

移动端的情况相对较好，移动端的浏览器基本上都是基于webkit的内核进行架构的，对HTML5和CSS3以及ES6的支持情况普通较好。手机使用周期短，更新换代速度快，对新技术的支持情况非常的好。

另外需要注意的是，对于移动端Web开发的屏幕分辨率问题，一般要手机厂商给出的分辨率的基础上宽和高都除以2，才能得到真实的显示分辨率。如iPhone8给出的屏幕分辨率是1334 x 750，那它的Web浏览器中的实际分辨率应该是1334/2 x 750/2，也就是667 x 375。因为在移动端，屏幕分辨率和浏览器分辨率并不是一样的。

* 有自己的技术博客（或作品网站）吗？

<解答参考>：

如实回答即可，就算没有也最好要有一个github来存放自己的作品。能有一个线上地址来展示自己的作品就更好了，如果有一个已备案域名的个人服务器来罗列自己的优秀作品，那就是上佳了。

* 使用何种IDE（集成开发环境，指代码编辑器、配套技术和软件，泛指编辑器），优点是什么？

<解答参考>：

问这样的问题很大原因是该公司团队已经都在使用一个约定好的编辑器，这样一定程度的便于团队的相互协作，不过无外非就Webstorm、Sublime Text 3、HBuilder、Visual Studio Code（VSCode）等几种主流的Web前端主流编辑器。

这里只需要说明自己使用的编辑器，并简单概括该编辑器的优势是什么，并稍微说一下不足之处。

当然，在有的公司已经有现成的后台语言开发好的项目，要运行代码也只有用相关的编辑器，如Eclipse/myEclipse系列，Visual Studio等。不过对于我们前端开发而言，需要编写的代码也无非就是HTML、CSS和JavaScript这三种类型的文件，使用自己喜好的编辑器完成功能，再去使用指定的编辑器运行启动检查效果即可。

* 你做过让你最满意的项目是什么？为什么？

<解答参考>：

一般能让自己满意的项目都主要是可能因为这几点：

* 视觉效果、交互动态制作得非常漂亮，用户体验极佳
* 项目具有较高难度，并且都被一一攻克（自己起到了关键作用的）
* 运用了较新的流行技术，或者是一种自己从未尝试过的编程模式，并且做出成型项目
* 这个项目产生了很高的价值（公益价值、商业价值、品牌价值、口碑价值等）
* 这个项目有着对个人的特殊意义（这个因人而异）

以上要点能选出其中一至两条回答即可，如果能有条理性的依次说出3条以上，那就比较完美了。

* 在你做过的项目中遇到过最大的问题或者难点是什么？后来是怎么解决的？

<解答参考>：

这种问题一般很少由技术面试官来问，而是人事的问题，是一个常见的问题，其考核点是你是否会去常常总结自己所做的项目和收获的知识，是验证一个人的学习方法或者说是验证一个人是否会快速的进步。

所以在回答这个问题的时候的要点就是解决问题的途径以及对这个问题的总结记录。

举个例子：团队在第一次尝试用Github来做项目的版本管理的初期，经常会出现大量的代码冲突，甚至代码被对方覆盖的情况，对git使用的流程也总是很混乱，经常会花费大量时间去解决这些冲突，甚至耗费在这上面的时间多余项目开发本身。

解决问题的回答应该是体现出自己价值的回答，就可以说成是：后来我就提议列出项目文件分工的明细，精确到人头，并使用XXX工具（如PPT、XMind等）画出了git代码更新提交的流程图，git的使用按照这个流程严格执行。要求被采纳之后，大家都按照这个方式去做，代码冲突就很少会发生了。

遇到的问题很普通，但解决过程很有条理，流程也很正规，能体现出个人解决问题的能力。当然，这只是一个例子。实际上项目中遇到的难点的问题其实是非常多的，关键是要加以总结，并使用笔记工具记录在案。防止以后遇到同样的问题的时候还需要花费大量时间去进行“二次”解决。

* 平时是通过什么途径学习前端技术的

<解答参考>：

技术论坛、手机App、专业技术网站、技术牛人博客文献、该技术的官方文档、线上线下技术交流、Github开源项目、现成的优秀Demo、专业知识问答平台（SegmentFault、Stack Overflow、知乎等）都可以作为回答点

回答的关键就是要体现出自己的学习能力和对技术点多样化的检索能力

* 有没有掌握过或考虑过学习其它方向的技术？比如Java、PHP、Python等

<解答参考>：

根据自己的真实意愿，分两种情况回答。

愿意的情况：

如果公司项目不紧急的时候，还是愿意抽时间去多学一门技术的，反正“技多不压身”嘛，也想通过了解学习其它的语言来提升自己的逻辑和架构思维。

不愿意的情况：

现在前端发展的速度比较快，很多新技术点和新模式不断的出现，这在几年前都是不可想象的。我认为现在前端还有很多东西都还需要我去学习，不想“顾此失彼”，所以还是打算先将Web前端领域的技术掌握得好一些再去考虑其它方向的事。

总之，无论是哪种回答，都尽量不要把话说得太死，这里要稍微表现得圆滑一点，避免面试官对你产生成见。

* 怎么理解团队

<解答参考>：

回答要点主要有：

* “八仙过海，各显神通”—— 各有所长，充分发挥每个人的专长
* “三人行必有我师”—— 互相学习，共同进步。为日后的项目打下更好基础，让产出只会越来越好
* “一根筷子容易折断，一捆筷子则不然”—— 对风险的抵御能力。孤军作战一旦遇到难关要想渡过并不是那么容易的，而一群人在一起就不一样了
* “三个臭皮匠赛过一个诸葛亮”—— 滴水汇成汪洋。当哪怕每个个体都很弱小的时候，一旦汇聚在一起都会产生惊人的能量
* 谈谈您在之前的公司团队中做出过最突出的贡献是什么

<解答参考>：

这个问题的回答方式比较多元化，但回答的要点必须要体现出这件事情的价值。譬如完成了一个XXXX项目，修改了多少个bug这些回答都是较次的回答方式。就算是干了这些事，也要体现出做这些事的价值所在，如：

* 利用私人时间为公司开发了一套OA（自动化办公）系统，解决了公司长久以来都用纸笔来统计员工业绩的问题
* 对公司的核心产品进行了大幅的性能优化，使得产品在启动速度和核心模块的响应速度上，经过多次测试取平均值分别提高了300和150个百分点。（测试方案使用的是console.time方法加浏览器开发者工具中的Performance和Network工具）
* 为开发团队带来了新的开发模式。以前他们开发都对库和框架的使用都是直接引用方式，代码也从来没有压缩和合并，团队的代码管理工具也使用的是FTP的方式。开发效率低下，程序可维护性很低，项目的加载速度很慢，安全性也非常低。后来教会他们使用SVN、Git等工具来管理代码，又教他们怎么用Gulp/Grunt来优化代码开发流程、压缩合并代码，提高项目代码的安全性

等等...，总之要注意的强调的不是自己所做的事，而是所做的事情产生的结果，需要确定这个结果对于团队而言是有价值的。

* 怎样看待加班

<解答参考>：

没人喜欢加班，但是有这样一个道理：如果公司做不好，那我们也好不到哪去。但如果做好了我们价值和利益都可以得到满足。如果加班是为了目的，那这样的加班也是义不容辞的。

当然在这里也可以表露一下自己的态度：我听说一些公司，明明没有事做，但偏偏要强行让员工加班到晚上九点十点，这样真没有什么意义。

表露自己态度的目的是刺探对方是不是这样一个公司，如果对方就是这样一个公司，那这样的公司就没有什么必要进去了。

* 如果你工作干得比同一个团队的同事好，但待遇却不如他，你会怎样去看待这个问题？

<解答参考>：

这里的回答其实很简单，要点是从自己身上找原因即可。

公司属于社会，在社会里任何时候做人都是最重要的。“事在人为”，如果人对了，那做的事也不会差在哪里去。如果人没有做好，得不到别人的好感，那自我的发展也差不多到此为止了。

* 您未来三年（或者五年）的职业规划是什么样的呢？

<解答参考>：

这是人事常爱问的问题，其发问目的是是判断你这个人对职业有没有规划，是不是一个有计划有目标的人，从而判断你是否能适应公司长期发展的需要。

如果是做基础开发岗位，大概的回答模版就是：

在未来的一到两年内，继续沉淀自己的技术，通过多方面的途径学习提升，多认识这个领域的大牛们，学习他们身上优秀的东西。在此期间开发出比较有价值的项目来作为一个阶段性的成果，并在此基础上继续自我的突破。争取在三年内能达到一个合格项目/产品经理的要求，在接下来的时间内也会朝着技术总监、架构师方面努力。

如果是面试管理位置，大概的回答应该是：

在进入公司两年内，能给公司带出一个出色技术和高度凝聚力的开发团队，制定出一套完善合理的团队管理制度，并且在此基础上根据公司的定位，给公司开发出一个有较高市场价值的项目产品，等等。

* 能谈谈你所认知的产品开发流程吗？

<解答参考>：

见PPT演示文档《产品开发流程-version》

总结：

精确的回答面试官的每一个问题，并且能突出回答的要点，一方面会使面试成绩几率直接提升，另外一方面，在谈判薪资的时候也能更加的主动。不要小看薪资多500块和少500块的差距，毕竟这样下来一年就能多6000块钱的薪水，加上年终奖（通常是n份底薪）差不多也是1万左右了，而1万块钱能干些什么相信大家心里也明白吧，反正凑个国产车的首付是差不多了。