

群面形式

2017年6月28日 14:51

1. 两难问题

是让考生在两种互有利弊的答案中选择其中的一种。

2. 多项选择问题

是让考生在多种备选答案中选择其中有效的几种或对备选答案的重要性进行排序。

3. 资源争夺问题

此类问题适用于指定角色的无领导小组讨论，是让处于同等地位的考生就有限的资源进行分配

★ “荒岛求生” 类

年龄、身份、援救他们所付出的成本、救出去之后能给社会创造多少价值、体力如何能在荒岛上撑多久

★ 互联网公司群面题型——围绕产品

1. 产品推广：电视广告、路边户外广告、电邮EDM、街景送流量、公交车广告、校园推广、明星代言

★ 情景方案类

（1） 题目简介

一般给出一个情景或话题，设计出一个解决方案

（2） 题目例子

- 假如你是CC语音的产品经理，请设计一个宣传推广的方案
- 给出4个人，爸爸（办公）、妈妈（上课、娱乐）、大伯（退休，厌烦传统娱乐方式）和小妹（90后新潮人士），每个人年龄不同，性别不同，生活方式不同，请设计一个满足全家人的移动生活解决方案（不限平台、APP等）

（3） 破题方式

- 仔细审题：确定应用场景
 - a) 地点：确定事件背景
 - b) 人物：确定目标用户
 - c) 要求：确定方案方向
- 提出解决方案类，一般可以将整个讨论分成两个基本讨论模块：载体+内容
 - a) 载体：确定方案展示形式、平台（注意联系移动互联网热点）
 - b) 内容：方案内容，如果小组之前讨论实在没有办法形成统一结论，那么可以将任务进行分期处理，将目标分类成近期较容易实现和近期不易实现，缓和组内矛盾

★ 角色扮演类

（1） 题目简介

故事型题目，给出一个矛盾情况，要求讨论抉择，尽量做到鱼和熊掌兼得。

（2） 题目例子

假设你是可口可乐公司的业务员，现在公司派你去偏远地区销毁一卡车的过期面包（不会致命的，无损于身体健康）。在行进的途中，刚好遇到一群饥饿的难民堵住了去路，因为他们坚信你所坐的卡车里有能吃的东西。这时报到难民动向的记者也刚好赶来。对于难民来说，他们肯定要解决饥饿问题；对于记者来说，他是要报道事实的；对于你业务员来说，你是要销毁面包的。

现在要求你既要解决难民的饥饿问题，让他们吃这些过期的面包（不会致命的，无损于身体健康），以便销毁这些面包，又要不让记者报到过期面包的这一事实？请问你将如何处理？

说明： 面包不会致命。不能贿赂记者。不能损害公司形象。

（3） 破题方式

此类问题一般给出一个看似两难的情况，而目的是尽量鱼和熊掌兼得，看到这样的题目往往会没有办法下手

- 审题：圈出关键字，如过期面包、可口可乐公司等
- 确定矛盾点：
 - a) 饥饿的难民的生命
 - b) 过期面包虽不致命，但记者会报道，对企业的声誉影响
- 确定矛盾优先级：人命关天
- 提出解决方案：
 - a) 方案1：向记者和难民说明情况，说明面包过期，跟公司反映情况，再送一批面包过来
 - b) 方案2：若难民不能等待，那就先救人，后续进行公关维护

（4） 注意事项

- 这类题目的矛盾冲突最为直接，往往是人性的冲击，这时应该保持清晰的头脑，使道德的一方优先级最高
- 方案可以不止一个，提出两个方案，作出选择和解释，缓和矛盾

（5） 一句话思路

审题→确定矛盾双方→道德方优先级提高→提出多个解决方案缓和矛盾

技巧1：矛盾优先级（道德人性>利益声誉）

技巧2：多个解决方案缓和矛盾

技巧

2017年6月28日 14:53

1. 小组成员应该有自己的观点和主见，即使与别人意见一致时，也可以阐述自己的论据，补充别人发言的不足之处，而不要简单地附和说：“某某已经说过了，我与他的看法基本一致。”
 2. 讲话停顿时显得像是在思考的样子，这么做能使你显得是那种想好了再说的人。这种做法在面对面的面试时是可以的，，在电话面试和可视会议系统面试时，不要做思考的停顿，否则会出现死气沉沉的缄默。
 3. 可以用一张便条纸大致画一下各个主试人所坐的位置，并一一写上他们的姓名。在谈话中能够提及主试人的姓名，往往有利于你给对方留下一个好的印象。
 4. 把握说服对方的机会：不要在对方情绪激动的时候力图改变他的观点。理解对方的观点，在此基础上，找出彼此的共同点，可先肯定对方的说法，再转折一下，最后予以否定，引导对方接受自己的观点。
 5. 在论辩中，如果自己处于被动的不利地位，就不要再纠缠于原来的话题，这时可以及时转移话题，或抓住对方的一个弱点。也可以从新的话题上向对方发起进攻，使自己变被动为主动。
- 如果是自己擅长的题目，争取当leader，如果没有人主动担任leader，要有勇气和魄力带领大家前进
 - 如果是不擅长的题目，争取当timer，比较安全，但要注意时间安排如何修正

前期角色争取

抢Leader

- (1) 主动性原则，若大家没动作，则主动提出担任leader；
- (2) 潜移默化：最好的方式还是从讨论入手，根据表现让组员认可，成为事实上的leader，也容易当reporter。
- (3) 混乱情况：主动担当，带领大家前进，之后的陈述也要有霸气的承认自己是leader

抢Timer

- (1) 带一块手表容易抢到timer，增大抢到的几率
- (2) 如果没抢到，也没关系，而且还能降低风险（讨论超时，timer必挂）。协助timer确定讨论时间安排，并时时关注时间。

抢Recorder

个人不是很主动想抢，因为风险较大，回报不高。不过自己一定要记录讨论结果，并

且要有逻辑性，从几次群面的结果来看，往往不是Recorder胜似Recorder。

抢Reporter

风险与机遇并存，记录的东西比较全时，容易受到大家的推举，也可自荐，不过容易引起组员和面试官的不好印象，所以，先等大家推举，实在没有人上才自荐。

面试官你好，下面将由我代表我们的组员，向您陈述我们小组的最终意见。（总起）在讨论之后，我们的排序依次是：大学教授，大学生，经理人，中学教师，小学校长，外科医生，将军，最后是运动员。根据您给的时间，我们进行了以下的安排：（讨论时间安排）5分钟的标准订立，7分钟的分类讨论，最后是2分钟的汇总整理，1分钟完善。在标准订立方面，小组成员的意见中没有太大的分歧，我们将“对社会的贡献程度”作为最重要的标准。（末尾再总结一次）

有人跳了我想要的角色，怎么解决？

- Leader：大部分人不会直接跳，自己要主动争取；若有人跳，则相信他的实力，不要引起争论；退而求其次当个timer，或reporter
- Timer：如果有人先跳出来要当timer，那么没关系，第一准则是不起冲突；可协助timer确定讨论的时间
- Recorder：无论能不能当recorder，都需要记录小组的讨论结果，能够掌握讨论的主动权
- Reporter：如果不是，那么争取在陈述完毕后进行补充，要有亮点，梳理逻辑

“两点一线”思维：把需要达到的目标看成终点，当前现状看成起点，解决问题的方法看成路径（工具）。**聚焦于获取“新路径”。**

- 反驳别人的论点：挖掘新路径
- 被他人反驳：抓住终点反击

“纵向横向”思维：横向分析——把问题切割成很多个阶段，分析每一个阶段。纵向分析——在同一个阶段中，分析每一种原因（因素）。

举个例子：论证滴滴打车软件的出现是注定成功的

分析：

- 1) 横向切割打车过程：等车、上车（空车不拒载）、行驶（司机态度、司机颜值、司机开车技术、司机路况熟悉程度）、结账下车（费用、到达时间、司机态度）。
- 2) 纵向分析每个阶段影响自己满意度的因素。等车时间、等车环境、等车心情、费用、时间.....

答案：从国家、社会、家庭、个人层面展开描述

“聚焦问题”思维：就事论事，把握“终点”

总结问题思维——顺序位于最后的同学

把前面同学提出的建议和方案有条不紊的总结出来，逻辑层次分明，按横纵向归类。

timekeeper：用时间这个理由去中断一些你不想再听的发言。

1. 如果大家不介意的话，我来帮忙记个时。
 2. 非常抱歉打断一下，第一个环节是轮流发言，要不先大家依次说完我们在自由讨论吧？
 3. 我觉得前面 XX 同学说的要细分到一个具体的人群很好，我认为上班族是一个不错的目标人群，有一定消费能力、需求也相对明确，另外，大家的发言我都有记录，回头方便整理汇总。
 4. 能不能针对生理周期的女性用户，为他们提供解决生理疼痛的热饮？
- 当话题开始倾斜之后，快速推动选定方案，尽快让团队走向下一个议题
5. XXX同学你怎么看，要不也说说看法？
 6. 没关系大家别急，我们还有十多分钟的时间，先选一个发言人再把剩下的内容敲定。

突发状况冷静处理，安定团队情绪，严格控制讨论的走向和进度

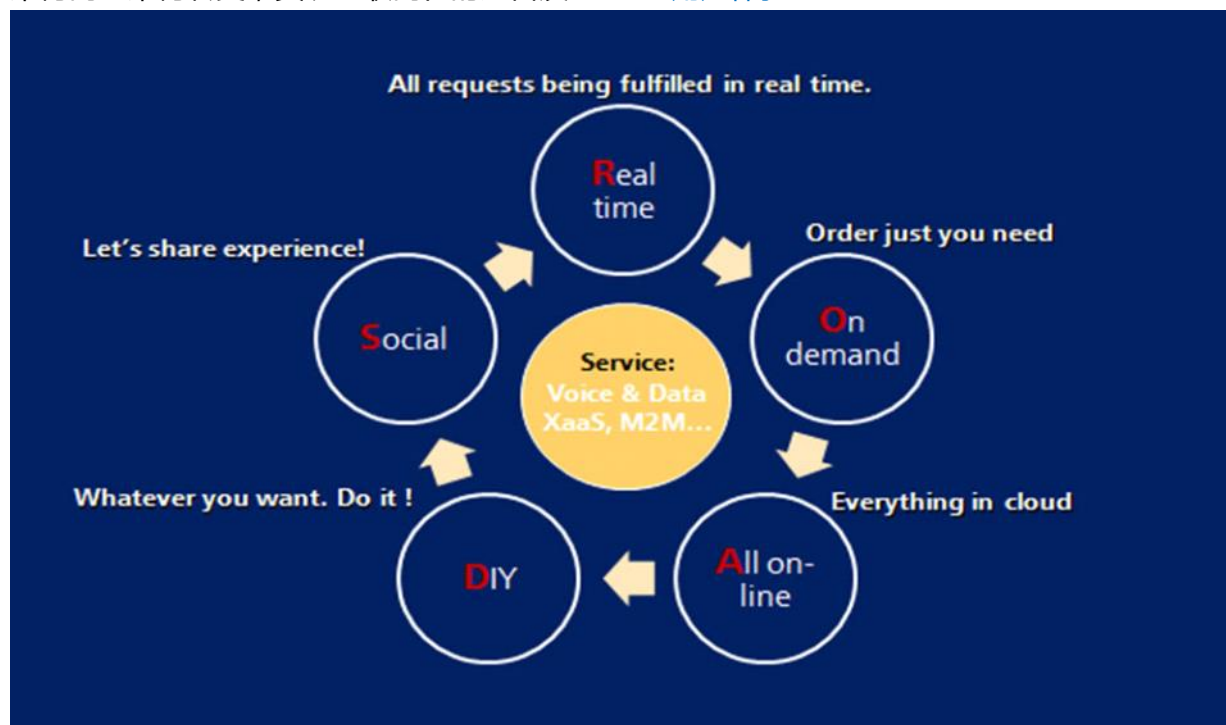
华为

2017年7月2日 20:25

华为CEO徐直军指出，2016年，运营业务在持续做强网络产品与服务的同时，以视频、云化、运营转型等为重大战略方向，做大产业、推动产业发展，在保持高增长的同时，运营资产效率也得到了提升。

华为运营业务BG总裁邹志磊

- ? 对于运营商BG而言，如何围绕OPEN RAN终极用户体验战略，为不同阶段的运营商转型提供全方位的支持就是华为运营业务BG最核心的问题了
- ✍ 全球电信运营商的网络建设逐步由“投资驱动”向“价值驱动”转变，我们要努力让运营商的网络能够覆盖更多的连接，帮助运营商将视频作为基础业务并取得商业成功，引领运营商IT系统由传统技术架构向云架构转变，实现互联网化的运营及ROADS用户体验。



- ✍ 运营商转型的4个阶段（经济红利阶段）
 1. 人口红利
 2. 流量红利
 3. 数据红利：把纯管道升级为智能管道，将管道中充斥的流量进行甄别，通过大数据分析等让数据产生更大价值。
 4. 信息红利：整个社会进入智能和数字化时代

? SoftCOM网络发展战略

汇集以云计算为代表的最新的设计理念和先进技术，构建完全开放的ICT融合架构帮助行业从以前单一

垂直维度创新，到产业链各方各维度Full Mesh自由组合创新，以支撑电信用户体验向ROADS的迈进，迎接未来的机遇和挑战



四个维度进行重构：

1. 架构重构：建立以数据中心为中心的ICT基础架构，适应信息时代的发展要求。
2. 网络重构：软件定义网络（SDN）和网络功能虚拟化，建立弹性和智能的网络架构和产品架构。
3. 业务重构：超越连接，借助云计算商业模式的变革，抓住ICT走向云服务的战略机遇，再造一个电信业。
4. 运营重构：互联网化运营，实现ROADS化的，从内部管控模式转变成外部用户服务的模式。

来自 <<http://www.c114.net/news/126/a885055.html>>

5.

网络故障排除

2017年6月29日 20:36

故障排除基本思路：

- 1、谁出了问题？
- 2、是什么问题？
- 3、问题何时发生的？
- 4、何处发生的故障？

网络故障一般分类：

- 1、连通性问题；
- 2、硬件、媒介、电源故障；
- 3、配置错误；
- 4、设备兼容性问题；
- 5、性能问题；
- 6、网络拥塞；
- 7、到目的地不是最佳路由；
- 8、供电不足；
- 9、路由环路；
- 10、网络不稳定；

主要命令：

ping：检测网络连接，主机是否可达

对每一ping报文的响应情况，如果超时到仍没有收到响应报文，则输出“Request time out”，否则显示响应报文中数据字节数、报文序号、TTL和响应时间等。

Tracert：用于测试数据包从发送主机到目的地所经过的网关，定位配置不当的网络点，分析网络什么地方发生了故障。

display interface：显示接口信息

display current-config& display saved-config：显示当前设备配置文件信息

debug：获得路由器中处理的报文和帧的细节信息

故障排除常用方法

- 1、分层故障排除法；
- 2、分块故障排除法；
- 3、分段故障排除法；

4、替换法；

➤ 分层故障排除法

按照OSI/ISO参考模型，由下到上，逐层进行故障排除。

➤ 分块故障排除法

管理部分（路由器名称、口令、服务、日志等）

端口部分（地址、封装、cost、认证等）

路由协议部分（静态路由、RIP、OSPF、BGP、路由引入等）

策略部分（路由策略、策略路由、安全配置等）

接入部分（主控制台、Telnet登录或哑终端、拨号等）

其他应用部分（语言配置、VPN配置、Qos配置等）

➤ 分段故障排除法

- ①. 开箱即无法使用，检查接口卡或主板上的器件，查看是否器件脱落或被压变形，以及BOOTROM或内存条的插座有无插针无法弹起。检查PCI侧的插针、物理接口（包括电缆）的插针是否有弯针。当没有查到上述硬件故障后，可考虑更换或升级BOOTROM、内存条或主机版本。
- ②. 安装后无法正常使用，线路连接问题，如线路阻抗不匹配、线序连接错误、中间传输设备故障。与其它设备有兼容性问题。接口配置问题。电源或接地不符合要求。在安装过程也要考虑模块接口电缆所支持的最大传输长度、最大速率等因素。
- ③. 使用过程中发生故障，电源、接地和防护方面不符合要求，在有电压漂移或雷击时造成器件损坏。传输线受到干扰。中间传输设备故障。环境的温湿度、洁净度、静电等指标超出使用范围。

网络安全

2017年6月30日 14:19

ARP投毒（中间人欺骗）：中间人欺骗PC机，告诉PC机它是服务器。在欺骗服务器，告诉服务器它就是PC机。以致获取服务器与pc机的会话信息。防御方法：在pc机里静态绑定服务器的ip地址和mac地址，或者是双向的静态绑定。

DoS（Denial of Service）攻击：故意地攻击网络协议实现的缺陷或直接耗尽被攻击对象的资源，目的是让目标计算机或网络无法正常提供服务或资源访问，使目标系统服务系统停止响应甚至崩溃，但此类攻击不包括侵入目标服务器或网络设备。

DDOS攻击（分布式拒绝服务攻击）：多个DoS攻击源一起攻击某台服务器就构成了DDOS攻击。

交换机安全

常见的二层安全攻击为为MAC层攻击、VLAN攻击、欺骗攻击和交换机设备攻击四类，详细攻击分类和攻击方法如下表如示。

CAM表是将这个交换机接口所连的PC的（MAC地址、端口、所属VLAN）去做HASH形成一个确定的数字表

攻击分类	攻击方法	攻击描述	抵御措施
MAC层攻击	MAC地址泛洪	具有唯一且无效源MAC地址的数据帧向交换机泛洪，消耗完交换机的CAM表空间，从而阻止合法主机的MAC地址生成新条目，去往无效主机的流量会向所有端口泛洪	端口安全 MAC地址VLAN访问控制列表
VLAN攻击	VLAN跳转	通过改变Trunk链路中封装的数据包的VLAN ID，攻击设备可以发送或接收不同VLAN中的数据包，而绕过三层安全机制	加强Trunk的配置和未使用端口的协商状态。 把未使用的端口放入公共VLAN
	公共VLAN设备之间的攻击	即使是公共VLAN中的设备，也需要逐一进行保护，尤其是在为多个客户提供设备的服务提供商网段中	实施私有VLAN（PVLAN）
欺骗攻击	DHCP耗竭和DHCP欺骗	攻击设备可以在一段时间内，消耗完DHCP服务器上的可用地址空间，或者在中间人攻击中，把自己伪装成DHCP服务器	DHCP侦听
	生成树欺骗	攻击设备伪装成STP拓扑中的根网桥。若成功了，攻击者就可以看到各种数据帧	主动配置主用和备用根设备 启用根防护

	MAC欺骗	攻击设备伪装成当前CAM表中合法设备的MAC地址，这样交换机就会把去往合法设备的数据帧发到攻击设备上。	DHCP侦听 端口安全
	ARP欺骗	攻击设备故意为合法主机伪造ARP应答。攻击设备的MAC地址就会成为该合法网络设备所发出的数据帧的二层目的地址。	动态ARP检测 DHCP侦听 端口安全
交换机设备安全	CDP修改	通过CDP发送的信息是明文形式且未加密，若攻击者截获CDP消息，就可以获得整个网络拓扑信息	在所有无意义使用的端口上禁用CDP
	SSH和Telnet攻击	Telnet数据包可以以明文形式查看 SSH可以对数据包进行保护，但版本1中仍存在安全问题	使用SSH版本2 使用Telnet结合VTY ACL

■ MAC泛洪攻击

常见的二层攻击或交换机攻击是MAC泛洪，入侵者使用大量无效的源MAC地址，这会导致交换机的CAM表溢出，由于交换机在MAC地址表中找不到目的MAC地址对应的端口，从而导致交换机向其它所有端口定期泛洪数据帧、交换机不会再接收新的合法条目，这会影响到网络中的所有用户的传输速度。攻击者可通过MAC泛洪实现DOS攻击，也可能会获取全网的数据包。

➤ 攻击步骤

- 1、攻击者以不同的源MAC地址发送多个数据包。
- 2、在短时间内，交换机的CAM表被填满，无法再接受新的条目。攻击的行为需要持续不断，否则MAC地址老化后会清空CAM表。
- 3、交换机开始把它接收到的所有数据包，向所有端口泛洪。因此，攻击者可能在任何端口获取网络中的所有数据包。

➤ 防御措施

在交换机端口上限定一个具体的MAC地址或限定MAC地址的数量。可以让交换机自动学习，也可以手动配置MAC地址。

使用粘性MAC地址：交换机端口自动学习MAC地址，然后记住这个MAC地址并将其以静态条目的形式存入运行配置（Running configuration）中，并把这个MAC地址当作端口安全特性允许接入的唯一MAC地址。如果交换机重启用，需要重新学习MAC地址。

科研及项目

2017年6月30日 16:33

data center故障预警和可靠性保障方案研究。根据data center的内存错误数据特性，使用合适的数学模型进行概率统计分析，并采取机器学习中训练数据的方法进行大数据分析处理，归纳出内存错误的特征。在UEFI中部署代码，根据实际需求设置预警值，实现预防服务器内存错误的目的。

★ 基于IEEE802.11P协议的车载无线通信系统

基于车联网技术和IEEE 802.11p协议，MATLAB/SIMULINK软件建模仿真了IEEE 802.11p协议的物理层。本文主要的研究内容是：全面分析IEEE 802.11p物理层的PPDU帧结构以及数据传输的过程，应用正交频分复用、时间交织、卷积编码、纠错编解码、Viterbi译码、信道估计和频域均衡等技术，研究在AWGN信道和Rayleigh信道中IEEE 802.11p物理层的仿真效果，实现3至27Mbps不同的数据率，并能根据信道的信噪比、多普勒频移等情况自适应地选取调制的方式和码率。本文末章总结了IEEE 802.11p协议的特点，并比较分析了IEEE 802.11p和其它无线通讯标准之间的异同点。

HTML

2017年8月9日 9:51

? Doctype作用？标准模式、兼容模式、怪异模式 各有什么区别？

(1)、<!DOCTYPE>声明位于位于HTML文档中的第一行，处于 <html> 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。
标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。

在兼容模式中，页面以宽松的向后兼容的方式显示，模拟老式浏览器的行为以防止站点无法工作。

怪异模式指的是使用浏览器自己的方式解析执行代码。浏览器会模拟Navigator 4和IE5的方式来渲染页面。

? HTML5 为什么只需要写 <!DOCTYPE HTML>

HTML5 不基于 SGML (standard global markup language标准通用标记语言)，因此不需要对DTD (document type definition)进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）

★ XML与SGML、HTML的关系

SGML、HTML是XML的先驱。

- SGML是指“通用标识语言标准” (Standard Generalized Markup Language), 它是国际上定义电子文件结构和内容描述的标准，是一种非常复杂的文档的结构，主要用于大量高度结构化数据的防卫区和其他各种工业领域，利于分类和索引。
- HTML即“HyperText Markup Language”（超文本标识语言），是一种预定义标记语言，它的优点是比较适合web 页面的开发。不能支持特定领域的标记语言，如对数学、化学、音乐等领域的表示支持较少。XHTML可以理解为html+xml，就是用xml的语法来规范html。HTML5是XHTML的下一代产品
- XML是一种元标记语言，所谓“元标记”就是开发者可以根据自己的需要定义自己的标记，比如开发者可以定义如下标记<book> <name>

? 如果页面使用 'application/xhtml+xml' 会有什么问题吗？

IE6，7，8不支持，IE6，7，8支持text/html。

? 行内元素有哪些？块级元素有哪些？空(void)元素有那些？

首先：CSS规范规定，每个元素都有display属性，确定该元素的类型，默认值为“block”，则为“块级”元素；属性值为“inline”，是“行内”元素。

(1) 行内元素有：a, b, span, img, input, strong, select, label, em, button, textarea

(2) 块级元素有：div ul ol li dl dt dd h1 h2 h3 h4...p

(3) 常见的空元素:
 <hr> <input> <link> <meta>

鲜为人知的是: <area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

? 页面导入样式时，使用link和@import有什么区别？

(1) link属于XHTML标签，除了加载CSS外，还能用于定义RSS，定义rel连接属性等作用；而@import是CSS提供的，只能用于加载CSS；

(2) 页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载

(3) @import是CSS2.1 提出的，只在IE5以上才能被识别，而link是XHTML标签，无兼容问题；

? 介绍一下你对浏览器内核的理解？

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)，JS引擎。

1. 渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

2. JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

? 常见的浏览器内核有哪些

Trident内核：IE, MaxThon, TT, The World, 360, 搜狗浏览器等。[又称MSHTML]

Gecko内核：firefox等，基于C++

Presto内核：Opera7及以上。 [Opera内核原为：Presto，现为：Blink;]

Webkit内核：Safari, Chrome等，基于KHTML开发。 [Chrome的：Blink (WebKit的分支)]

? 简述一下你对HTML语义化的理解？

用正确的标签做正确的事情。

html语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析；

即使在没有样式CSS情况下也以一种文档格式显示，并且是容易阅读的；

搜索引擎的爬虫也依赖于HTML标记来确定上下文和各个关键字的权重，利于SEO；

使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

? 如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些？

1. 标签及属性

2. 地理位置 geolocation

Geolocation (地理定位) 用于定位用户的位置，可返回用户位置的经度和纬度

3. 画布 canvas & 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。

- 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）
- 复杂度高能减慢渲染速度（任何过度使用 DOM 的应用都不快）
- 不适合游戏应用

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。

- 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

4. 视频 video

5. 音频 audio

6. 拖放 drag, drop

7. 微数据

8. 应用程序缓存 cache manifest——HTML5 离线存储

离线存储是指：在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。

9. Web 存储 (localStorage, sessionStorage)

10. web workers

Web Workers 是 HTML5 提供的一个 javascript 多线程解决方案，我们可以将一些大计算量的代码交由 web Worker 运行而不冻结用户界面。基本原理就是在当前 javascript 的主线程中，使用 Worker 类加载一个运行在后台的 javascript 文件来开辟一个新的线程，起到互不阻塞执行的效果，并且提供主线程和新线程之间数据交换的接口：

postMessage, onmessage。

11. 服务器发送事件 server-sent event：网页自动获取来自服务器的更新。

? 请描述一下 cookies, sessionStorage 和 localStorage 的区别？

cookie 是什么？

Cookie 实际上是一小段的文本信息。客户端请求服务器，如果服务器需要记录该用户状态，就使用 response 向客户端浏览器颁发一个 Cookie。客户端浏览器会把 Cookie 保存起来。当浏览器再请求该网站时，浏览器把请求的网址连同该 Cookie 一同提交给服务器。服务器检查该 Cookie，以此来辨认用户状态。服务器还可以根据需要修改 Cookie 的内容。

cookie 的作用

HTTP 协议是无状态的协议。一旦数据交换完毕，客户端与服务器端的连接就会关闭，再次交换数据需要建立新

的连接。这就意味着服务器无法跟踪会话。而cookie可以用于跟踪用户的会话。

cookie的特征

- cookie具有不可跨域特性：例如浏览器在访问Google页面时只会携带Google的cookie，而不会携带百度的cookie。虽然网站images.google.com与网站www.google.com同属于Google，但是域名不一样，二者同样不能互相操作彼此的Cookie。
- cookie对中文只支持UTF-8编码
- cookie支持保存二进制图片

cookie的属性

- String name – cookie的名称（不可改）
- Object value – cookie的值
- int maxAge – Cookie失效的时间（second）
如果为正数，则该Cookie在maxAge秒之后失效。
如果为负数，该Cookie为临时Cookie，关闭浏览器即失效。默认-1
如果为0，表示删除该Cookie。
- boolean secure –该Cookie是否仅被使用安全协议传输。安全协议。安全协议有HTTPS，SSL等，在网络上传输数据之前先将数据加密。默认为false。
- String path – cookie路径
- String domain – cookie域名
- int version – Cookie版本号。0表示遵循Netscape的Cookie规范，1表示遵循W3C的RFC 2109规范。

什么是cookie隔离？

在请求资源的时候不携带cookie信息。

解决方法：静态文件不要存储在主域名目录下，这样在使用非主要域名的时候，请求头中就不会带有cookie数据，以降低请求头的大小、请求时间，从而达到降低整体请求延时的目的。同时，这种方法不会将cookie传入Web Server，也减少了Web Server对cookie的处理分析环节，提高了webserver的http请求的解析速度。

cookie用于记录客户端用户信息，session用于记录服务器端的用户信息 cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务器间来回传递。sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小：

cookie数据大小不能超过4k。

sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，>=5M

有限时间：

localStorage	存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；
sessionStorage	数据在当前浏览器窗口关闭后自动删除。
cookie	设置的cookie过期时间（默认是截止到关闭浏览器）之前一直有效，即使窗口或浏览器关闭

session的生命周期

session保存在服务器的内存里。Session在用户第一次访问服务器的时候**自动创建**。Session生成后，只要用户继续访问，服务器就会更新Session的最后访问时间，并维护该Session。Session的超时时间为maxInactiveInterval属性，可以通过对应的getMaxInactiveInterval()获取，通过setMaxInactiveInterval(longinterval)修改。

session的识别

服务器向客户端浏览器发送一个名为JSESSIONID的Cookie，它的值为该Session的id（也就是HttpSession.getId()的返回值）。Session依据该Cookie来识别是否为同一用户。例如用同一台机器的chrome和Firefox浏览器访问服务器时，会生成两个不同的Session。但是用多个chrome的内建窗口访问服务器时生成同一个session。

? cache manifest原理

- 原理：基于一个新建的cache manifest 文件的缓存机制(不是存储技术)，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

- 如何使用：

- 1、页面头部像下面一样加入一个manifest的属性；
- 2、在cache.manifest文件的编写离线存储的资源；

```
CACHE MANIFEST
```

```
#v0.11
```

```
CACHE:
```

```
js/app.js
```

```
css/style.css
```

```
NETWORK:
```

```
resource/logo.png
```

```
FALLBACK:
```

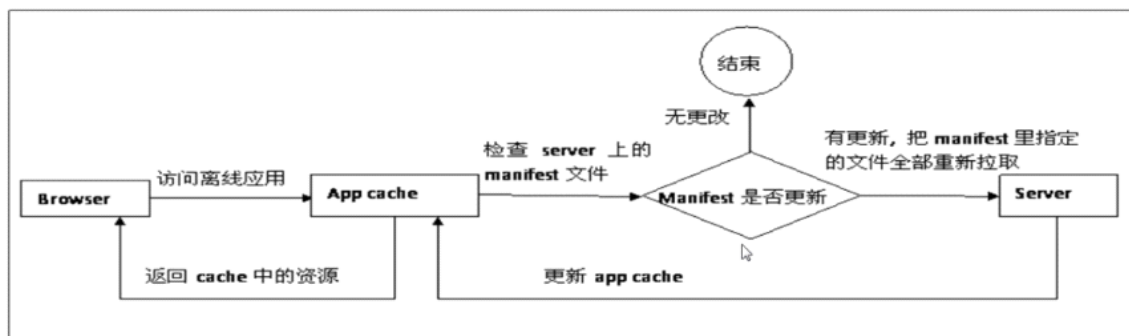
```
/ /offline.html
```

- 3、在离线状态时，操作window.applicationCache进行需求实现。

- 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？

- 1、在线的情况下，浏览器发现html头部有manifest属性，它会请求manifest文件，如果是第一次访问app，那么浏览器就会根据manifest文件的内容下载相应的资源，并且进行离线存储。如果已经访问过app并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。

- 2、离线的情況下，浏览器就直接使用离线存储的资源。



? html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和

HTML5 ?

* HTML5 现在已经不是 SGML 的子集，主要是关于**图像**，**位置**，**存储**，**多任务**等功能的增加。

绘画 canvas;

用于媒介回放的 video 和 audio 元素;

本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失，

sessionStorage 的数据在浏览器关闭后自动删除;

语义化更好的内容元素，比如 article、footer、header、nav、section;

表单控件：calendar、date、time、email、url、search;

新的技术 webworker, websocket, Geolocation;

现很多网站为了实现即时通讯，所用的技术都是轮询(polling)。轮询是在特定的时间间隔（如每1秒），由浏览器对服务器发出HTTP request，然后由服务器返回最新的数据给客户端的浏览器。这种传统的HTTP request 的模式带来很明显的缺点 – 浏览器需要不断的向服务器发出请求，然而HTTP request 的header是非常长的，里面包含的有用数据可能只是一个很小的值，这样会占用很多的带宽。

WebSocket API是下一代客户端-服务器的异步通信方法。该通信取代了单个的TCP套接字，使用ws或wss协议，互相沟通的Header是很小的-大概只有 2 Bytes, 任意的服务器和客户端可以在给定的时间范围内的任意时刻，相互推送信息。服务器不再被动的接收到浏览器的request之后才返回数据，而是在有新数据时就主动推送给浏览器。因为Ajax技术需要客户端发起请求，而WebSocket服务器和客户端可以彼此相互推送信息；Ajax受到域的限制，而WebSocket允许跨域通信。

? websocket如何兼容低浏览器？(阿里)

Adobe Flash Socket、ActiveX HTMLFile (IE)、基于 multipart 编码发送 XHR、基于长轮询的 XHR XMLHttpRequest 对象提供了对 HTTP 协议的完全的访问。XMLHttpRequest 可以同步或异步地返回 Web 服务器的响应，并且能够以文本或者一个 DOM 文档的形式返回内容。

* 移除的元素：

纯表现的元素：acronym, applet, basefont, big, center, dir, font, strike, tt

对可用性产生负面影响的元素：frame, frameset, noframes;

* 支持HTML5新标签：

a. IE8/IE7/IE6支持通过document.createElement方法产生的标签

b. 可以直接使用成熟的框架、比如html5shim;

```
<!--[if lt IE 9]>
```

```
<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>
```

```
<![endif]-->
```

* 如何区分HTML5：DOCTYPE声明\新增的结构元素\功能元素

? iframe有那些缺点？

1、iframe会创建包含另外一个文档的内联框架，会阻塞主页面的Onload事件，搜索引擎的检索程序无法解读这种页面，不利于SEO（搜索引擎优化）。2、iframe和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行

加载

解决方法：通过javascript 动态给iframe添加src属性值（外文档的路径），这样可以避开以上两个问题。

? 如何实现浏览器内多个标签页之间的通信

- 1、webSocket
- 2、sharedWorker: 共享工作线程允许多个页面共享使用，每个页面都是链接到该共享工作线程的某个端口号上。页面通过该端口与共享工作线程进行通信
- 3、LocalStorage: 另一个浏览上下文里被添加、修改或删除时，它都会触发一个事件，我们通过监听事件，控制它的值来进行页面信息通信；
- Cookies

? 在设计和开发多语言网站时，要考虑哪些问题？

1. 字符集的选择：Unicode，非中文页面字符集设定为 'utf-8'
2. 语言书写习惯&导航结构：有些国家的语言书写习惯时从右到左
3. 数据库驱动型网站（客户可以留言并向数据库添加信息）：考虑如何从技术上实现对不同语言数据信息的收集和检索

? 网页加载的完整过程？script和css的位置要求？

1. 首先浏览器从服务器接收到html代码，然后开始解析html
2. 构建DOM树（根据html代码自顶向下进行构建），并且在同时构建渲染树
3. 遇到js文件加载执行，将阻塞DOM树的构建；遇到css文件，将阻塞渲染树的构建

script标签最好放在</body>标签的前面，因为放在所有body中的标签后面就不会出现网页加载时出现空白的情况，可以持续的给用户提供视觉反馈，同时在有些情况下，会降低错误的发生。

css标签应该放在<head></head>标签之间，因为如果放在</body>标签的前面，那么当DOM树构建完成了，渲染树才构建，那么当渲染树构建完成，浏览器不得不再重新渲染整个页面，这样造成了资源的浪费。效率也不高。如果放在<head></head>之间，浏览器边构建边渲染，效率要高的多。

? 浏览器渲染页面和解析加载页面的机制

浏览器的5个常驻线程：

1. 浏览器GUI渲染线程
2. javascript引擎线程
3. 浏览器定时器触发线程（setTimeout）
4. 浏览器事件触发线程
5. 浏览器http异步请求线程（.jpg <link />，外部css文件 这类请求）

注意：这里也涉及到阻塞的现象，当js引擎线程（第二个）进行时，当在进行第二线程的时候，1，3，4，5都会挂起

? 网页搜索全过程解析

当我们在浏览器的地址栏输入网址（譬如：www.linux178.com），然后回车，回车这一瞬间到看到页面到底发生了什么？

域名解析 --> 发起TCP的3次握手 --> 建立TCP连接后发起http请求 --> 服务器响应http请求，浏览器得到html代码 --> 浏览器解析html代码，并请求html代码中的资源（如js、css、图片等） --> 浏览器对页面进行渲染呈现给用户

? 域名是如何解析的呢

1. Chrome浏览器会首先查看（使用 `chrome://net-internals/#dns`）**浏览器自身的DNS缓存**。（缓存时间比较短，大概只有1分钟，且只能容纳1000条缓存）
2. 如果浏览器自身的缓存里面没有找到对应的条目，那么Chrome会搜索（在命令行下使用 `ipconfig /displaydns`）**操作系统自身的DNS缓存**，如果找到且没有过期（服务器是根据什么判断缓存是否还有效呢？有两种方式：ETag（实体标签）、最新修改日期（Last-Modified响应头））则停止搜索解析到此结束。
3. 如果在Windows系统的DNS缓存也没有找到，那么尝试读取**hosts文件**（位于 `C:\Windows\System32\drivers\etc`），看看这里面有没有该域名对应的IP地址，如果有则解析成功。
4. 如果在hosts文件中也没有找到对应的条目，浏览器就会发起一个DNS的系统调用，就会向**本地配置的首选DNS服务器**（一般是电信运营商提供的，也可以使用像Google提供的DNS服务器）发起域名解析请求（通过的是UDP协议向DNS的53端口发起请求，这个请求是**递归的请求**，也就是运营商的DNS服务器必须得提供给我们该域名的IP地址）
5. 运营商的DNS服务器首先查找自身的缓存，找到对应的条目，且没有过期，则解析成功。
6. 如果没有找到对应的条目，则有运营商的DNS代我们的浏览器发起迭代DNS解析请求。
 - ①. 请求根域的DNS的IP地址（这个DNS服务器都内置13台根域的DNS的IP地址）。
根域服务器返回一个顶级域com域的一个域名
 - ②. 请求com域的IP地址
com域这台服务器返回linux178.com这个域的DNS地址（这个一般就是由域名注册商提供的，像万网，新网等）
 - ③. 向linux178.com这个域发起请求对应的IP地址
linux178.com域的DNS服务器返回其IP地址给运营商DNS服务器
 - ④. 运营商DNS服务器将IP返回给Windows系统内核，内核又把结果返回给浏览器，终于浏览器拿到了 `www.linux178.com` 对应的IP地址，该进行一步的动作了。

特殊情况：

7. 操作系统就会查找NetBIOS name Cache，凡是最近一段时间内和我成功通讯的计算机的计算机名和IP地址，就都会存在这个缓存里面。什么情况下该步能解析成功呢？就是该名称正好是几分钟前和我成功通信过，那么这一步就可以成功解析。
8. 如果第7步失败，查询WINS服务器（是NETBIOS名称和IP地址对应的服务器）
9. 如果第8步失败，客户端进行广播查找
10. 如果第9步失败，客户端读取LMHOSTS文件。如果失败，本次域名解析以失败告终。

? 重定向

301：永久重定向，主要用于当网站的域名发生变更之后，告诉搜索引擎域名已经变更了，应该把旧域名的数据和链接数转移到新域名下，从而不会让网站的排名因域名变更而受到影响。

302：临时重定向，主要实现post请求后告知浏览器转移到新的URL。

304：Not Modified，主要用于当浏览器在其缓存中保留了组件的一个副本，同时组件已经过期了，这是浏览器就会生成一个条件GET请求，如果服务器的组件并没有修改过，则会返回304状态码，同时不携带主体，告知浏览器可以重用这个副本，减少响应大小。

重定向应用场景：跟踪内部流量、出站流量

? 前端性能优化

CSS 优化主要是四个方面：

1. 加载性能

主要是从减少文件体积、减少阻塞加载、提高并发方面入手的。

①. 减少HTTP请求（80%~90%时间花在了下载页面中的所有组件进行的HTTP请求上）

图片地图：假设导航栏上有五幅图片，点击每张图片都会进入一个链接，这样五张导航的图片在加载时会产生5个HTTP请求。然而，使用一个图片地图可以提高效率，这样就只需要一个HTTP请求。服务器端图片地图：将所有点击提交到同一个url，同时提交用户点击的x、y坐标，服务器端根据坐标映射响应。客户端图片地图：直接将点击映射到操作

CSS Sprites：通过将多个图片融合到一副图里面，然后通过CSS的一些技术布局到网页上。

②. CDN（内容分发网络）是一组分布在多个不同地理位置的Web服务器，用于更加有效地向用户发布内容。CDN可能选择网络阶跃数最小的服务器，或者具有最短响应时间的服务器。

CDN还可以进行数据备份、扩展存储能力，进行缓存，同时有助于缓和Web流量峰值压力。

③. 添加Expires头

④. 压缩组件

Web客户端可以通过HTTP请求中的Accept-Encoding头来表示对压缩的支持。Web服务器通过响应中的Content-Encoding来通知Web客户端

⑤. 使用外部的JavaScript和css脚本

⑥. 减少DNS查找

⑦. 避免重定向

⑧. 配置Etag

实体标签(EntityTag)是唯一标识了一个组件的一个特定版本的字符串，是web服务器用于确认缓存组件的有效性的机制，通常可以使用组件的某些属性来构造它。

引入Etag的目的：解决last-modified无法解决的问题

2. 选择器性能

selector 的考察更多是规范化和可维护性、健壮性方面。

key selector (选择器最后面的部分, 用于匹配目标元素的部分)

- ①. 如果使用的是ID选择器作为key selector, 那么过滤掉无关的规则, 不要用以下写法 "div#mydiv" 或 "#mydiv.div2"
- ②. 提取项目的通用公有样式, 增强可复用性, 按模块编写组件
- ③. 使用预处理工具 (less, sass)或构建工具 (gulp对css进行语法检查、自动补前缀、打包压缩、自动优雅降级)

优雅降级 (graceful degradation) : 一开始就构建完整的功能, 然后再针对低版本浏览器进行兼容。(Web站点在所有新式浏览器中都能正常工作, 如果用户使用的是老式浏览器, 则代码会检查以确认它们是否能正常工作。由于IE独特的盒模型布局问题, 针对不同版本的IE的hack实践过优雅降级了, 为那些无法支持功能的浏览器增加候选方案, 使之在旧式浏览器上以某种形式降级体验却不至于完全失效。)

```
.transition{
    transition: all .5s;
    -o-transition: all .5s;
    -moz-transition: all .5s;
    -webkit-transition: all .5s;
}
```

渐进增强 (progressive enhancement) : 针对低版本浏览器进行构建页面, 保证最基本的功能, 然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。(从被所有浏览器支持的基本功能开始, 逐步地添加那些只有新式浏览器才支持的功能, 向页面添加无害于基础浏览器的额外样式和功能。当浏览器支持时, 它们会自动地呈现出来并发挥作用。)

“渐进增强”观点认为应关注于内容本身。内容是我们建立网站的诱因。有的网站展示它, 有的则收集它, 有的寻求, 有的操作, 还有的网站甚至会包含以上的种种, 但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。

```
.transition{
    -webkit-transition: all .5s;
    -moz-transition: all .5s;
    -o-transition: all .5s;
    transition: all .5s;
}
```

3. 渲染性能

渲染性能是 CSS 优化最重要的关注对象。页面渲染 junky 过多? 看看是不是大量使用了 text-shadow? 是不是开了字体抗锯齿? CSS 动画怎么实现的? 合理利用 GPU 加速了吗? 什么你用了 Flexible Box Model? 有没有测试换个 layout 策略对 render performance 的影响? 这个方面搜索一下 CSS render performance 或者 CSS animation performance 也会有一堆一堆的资料可供参考。

4. 可维护性、健壮性

◦ 可重用性

一般来说, 一个项目的整体设计风格是一致的, 页面中肯定有几个风格一致但有些许不同的模块, 如何在尽可能多地重用CSS代码, 尽可能少地增加新代码, 这是CSS代码中非常重要的一点。如果CSS代码的可重用性高, 我们可能只需要写一些不一样的地方, 对页面性能和可维护性、提高开发效率都有很大的帮助。

- 可扩展性

如果产品增加了某个功能，我们应该保证新增加的CSS代码不会影响到旧的CSS代码和页面，并且尽可能少地增加新代码而重用旧代码。

- 可修改性

如果某个模块产品经理觉得要修改样式，或者要删掉它，如果没有规划好相应的CSS代码，过了一段时间之后，开发人员可能已经不记得这段代码作用了几个地方，不敢修改或删除它，这样下去CSS代码也就越来越多，影响了页面的性能，还造成了代码的复杂度。



如何在页面上实现一个圆形的可点击区域？

1、map+area：设定作用区域（又称为热点），这样当用户的鼠标移到指定的作用区域点击时，会自动链接到预先设定好的页面

2、svg矢量图

3、border-radius

4、纯js实现 需要求一个点是否在圆上简单算法、获取鼠标坐标等等



HTML data-* 属性

data-* 属性用于存储页面或应用程序的私有自定义数据。

data-* 属性赋予我们在所有 HTML 元素上嵌入自定义 data 属性的能力

```
<ul>
<li data-animal-type="bird">Owl</li>
<li data-animal-type="fish">Salmon</li>
<li data-animal-type="spider">Tarantula</li>
</ul>
```

来自 <<https://www.zhihu.com/question/23250329>>

? 介绍一下标准的CSS的盒子模型？低版本IE的盒子模型有什么不同的？

- (1) 有两种，IE 盒子模型、W3C 盒子模型；
- (2) 盒模型：内容(content)、填充(padding)、边框(border)、边界(margin)；
- (3) 区别：IE的content部分把 border 和 padding计算了进去；

? css选择器

id选择器 (#mydiv)

class选择器 (.mydiv)

标签选择器 (div)

属性选择器 (a[rel = 'external'])

复合选择器 (相邻兄弟 li+a , 普通兄弟 li~a , 后代li a , 子li >a , 并集 h1,h2)

通配符选择器 (*)

伪类选择器 (li: nth-child , a: hover, button: focus)

Css3新增伪类

p:last-of-type, p:first-of-type, p:only-of-type, p:nth-of-type(n), p:nth-last-of-type

p:last-child, p:first-child, p:only-child, p:nth-child(n), p:nth-last-child(n)

: enabled

: checked

: disabled

: enabled

::selection

::after

:empty

? 浏览器怎样解析CSS选择器

样式系统从最右边的选择符开始向左进行匹配规则。只要当前选择符的左边还有其他选择符，样式系统就会继续向左移动，直到找到和规则匹配的元素，或者因为不匹配而退出。

如果你非常在意页面的性能那千万别使用CSS3选择器。实际上，在所有浏览器中，用 class 和 id 来渲染，比那些使用同胞，后代选择器，子选择器 (sibling, descendant and child selectors) 对页面性能的改善更值得关注。

CSS选择器的效率从高到低做了一个排序：

1. id选择器 (#myid)
2. 类选择器 (.myclassname)
3. 标签选择器 (div, h1, p)
4. 相邻选择器 (h1+p)
5. 子选择器 (ul > li)
6. 后代选择器 (li a)
7. 通配符选择器 (*)
8. 属性选择器 (a[rel="external"])
9. 伪类选择器 (a:hover, li:nth-child)

? 哪些属性可以继承？

可继承

字体系列：color, font, font-family, font-size, font-weight, font-style, font-variant, font-stretch, font-size-adjust

文本系列：text-indent, text-align, text-transform, line-height, word-spacing, letter-spacing, direction

列表系列：list-style, list-style-image, list-style-position, list-style-type

表格系列：caption-side, border-collapse, border-spacing, empty-cells, table-layout
声音样式

页面样式：page, page-break-inside, windows, orphans

光标cursor

元素可见性visibility

? CSS3新特性

选择器

@font-face 加载（服务端）字体样式，即便客户端没有安装也可以正常显示

圆角 border-radius

多列布局 multi-column layout <http://www.w3cplus.com/content/css3-multi-columns-span>

弹性盒布局 Flexbox <http://www.w3cplus.com/css3/flexbox-basics.html>

阴影 text-shadow, box-shadow

反射 reflect

文字渲染 text-decoration

线性渐变 gradient

过渡效果 transition

2D转换效果 transform（translate水平移动，rotate旋转，scale伸缩，skew倾斜）

动画效果 animation

? 浏览器兼容性问题？原因，解决方法？ CSS hack？

1. 不同浏览器的标签默认的padding, margin不同

```
*{  
margin:0;  
padding:0;  
}
```

2. 块属性标签float后，又有横行的margin情况下，在IE6显示margin比设置的大,常见症状是IE6中后面的一块被顶到下一行

```
float:left;  
display:inline;
```

3. 设置较小高度标签（一般小于10px），在IE6，IE7，遨游中高度超出自己设置高度
overflow:hidden;
或者设置line-height 小于你设置的高度

4. 行内属性标签，设置display:block后采用float布局，又有横行的margin的情况，IE6间距超过设置的间距

```
display:inline;  
display:table;  
display:block;
```

5. img标签默认有间距

```
img{  
float:left;  
}
```

6. IE下,event对象有x,y属性,但是没有pageX,pageY属性;
Firefox下,event对象有pageX,pageY属性,但是没有x,y属性。

7. Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示,
可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决。

8. 超链接访问过后hover样式就不出现了 被点击访问过的超链接样式不在具有hover和active。解决方法是改变CSS属性的排列顺序:

```
L-V-H-A : a:link {} a:visited {} a:hover {} a:active {}
```

Css hack 利用浏览器一些不标准的，或者可以称之为 bug 的特性，达到特定的目的。

? li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

浏览器的默认行为是把inline元素间的空白字符（空格换行tab）渲染成一个空格

1. 写在一行
2. Float
3. flexBox布局

4. 父级元素设置font-size: 0px;

```
5. ul{  
  letter-spacing:-9px;  
}  
li{  
  letter-spacing:normal;  
}
```

6.

```
<ul>  
  <li>1</li><li>  
    2</li><li>  
      3</li>  
</ul>
```

? css定位

absolute

生成绝对定位的元素，第一个定位为absolute/relative的父元素进行定位。

不占用文档流位置，叠层通过z-index定义

fixed（老IE不支持）

生成绝对定位的元素，它的containing block一律为根元素即html/body。

在页面位置固定，不随滚动条移动。

position: fixed; 在android下无效怎么处理？

fixed是相对整个页面固定位置，在屏幕上滑动只是在移动viewport

relative

生成相对定位的元素，相对他的父元素的内容框（即去掉padding的部分）进行定位。

占用文档流位置，叠层失效

static

默认值。元素出现在正常的流中（忽略 top, bottom, left, right z-index 声明）。

inherit

规定从父元素继承 position 属性的值。

? 隐藏元素

1. display: none 不占用位置

2. height: 0; width: 0; overflow: hidden;

3. z-index: -1000

4. position: absolute; left: -999999px;

5. opacity: 0;

6. visibility: hidden 占用位置

7. visibility: collapse 作用相当于display:none，当应用于table标签时相当于visibility: hidden（firefox,

chrome支持，IE会显示元素)

? position跟display、margin collapse、overflow、float这些特性相互叠加后会怎么样？

display为none：

- 元素不被渲染
- position, float无效

display非none & position=absolute/fixed：

- float无效

display = inline-block的元素它的margin不会和垂直方向上的其他元素margin折叠

? 请解释一下为什么需要清除浮动？清除浮动的方式？

浮动会使当前标签产生向上浮的效果，同时会影响到前后标签、父级标签的位置及 width height 属性。

```
<div class="div1">
<div class="div2"></div> //float:left
<div class="div3"></div> //float:left
</div>
<div class="div4 clearfloat"></div>
```

(1) 父级元素设置 height

缺点：只适合高度固定的布局

(2) 父级元素设置 display: table

(3) 父级元素设置 overflow: hidden/auto

```
.div1{
  width:300px; /*or zoom:1, 不能定义height*/
  background-color:gray;
  overflow:auto/hidden;
}
```

(4) .clearfloat {

clear:both;

}

(5) .div1::after{

display:block;

clear:both;

content:"";

visibility:hidden;

height:0;

}

```
.div1{
    zoom:1; //设置或检索对象的缩放比例，可解决ie6,ie7浮动问题
}
```

zoom属性是ie浏览器的专有属性，触发hasLayout属性，设置或检索对象的缩放比例（此时高度宽度就会重新计算了）

？ 外边距合并问题

原因：普通文档流中block元素的垂直margin发生合并，合并后的值取最大margin值（若有负值，取绝对值后用正直减去最大值）。、

行内元素、浮动元素、绝对定位元素、设置了overflow属性的元素不会出现此现象。

任意元素的父元素不参与margin的折叠

根元素的垂直margin不会被折叠

消除方法是形成BFC

• 相邻兄弟元素之间

①.

```
<divclass="div1">
<divclass="div2"></div>
<divclass="div3"></div>
</div>
```

最后一个元素

②. 不在一个bfc内

```
<divclass="div1">
<divclass="div2"></div>
<divclass="dib3box"><divclass="div3"></div></div>
</div>
```

```
.dib3box{
    overflow:hidden;
}
.div2{
    width:50px;
    height:50px;
    margin:10px0;
    background-color:pink;
}
.div3{
    width:50px;
    height:50px;
    margin:10px0;
    background-color:pink;
}
```

- 父子元素之间

```
<div class="div1">  
<div class="div2"></div>  
</div>
```

- ①. 父元素加边框
- ②. 父元素加padding

? 怎么让Chrome支持小于12px 的文字？

- ①. 用图片：如果是内容固定不变情况下，使用将小于12px文字内容切出做图片，这样不影响兼容也不影响美观。

- ②.

```
p{  
    font-size: 12px;  
    -webkit-transform: scale(0.5); /*font-size is 6px in the fact*/  
}
```

? 让页面里的字体变清晰，变细用CSS怎么做？

```
-webkit-font-smoothing: antialiased;
```

? 手写动画，最小时间间隔多少合适？

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为 $1/60 * 1000\text{ms} = 16.7\text{ms}$

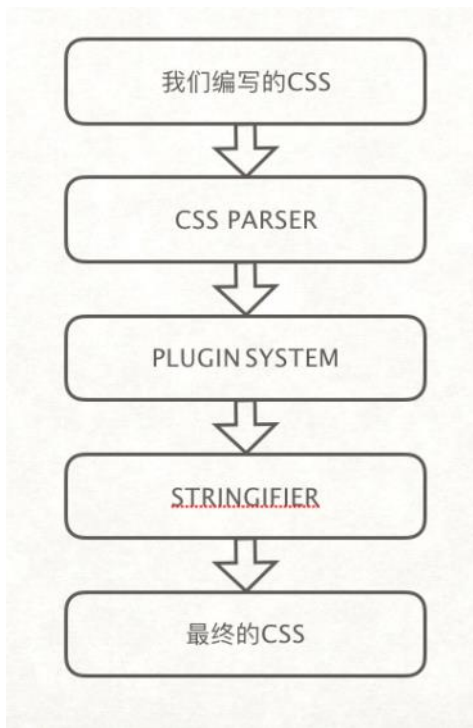
? CSS预处理器、postCSS后处理器问题

CSS 预处理器用一种专门的编程语言，进行web页面样式设计，再编译成正常的css文件，以供项目使用。

常用的有SASS, LESS, Stylus

	SASS/SCSS	LESS	Stylus
背景	最早诞生	基于sass改进	来自与node.js社区
安装环境	依赖于Ruby环境	依赖于Node.js环境	依赖于Node.js环境
语法规则	以缩进作为分隔符区分代码（和python类似） 或用花括号	用花括号	

PostCSS 可以理解为平台，提供了一个解析器，它能够将 CSS 解析成抽象语法树（AST）。



PostCSS 它需要一个插件系统才能够发挥作用

它能够为 CSS 提供额外的功能；

通过在 PostCSS 这个平台上，我们能够开发一些插件，来处理我们的CSS，比如热门的：autoprefixer

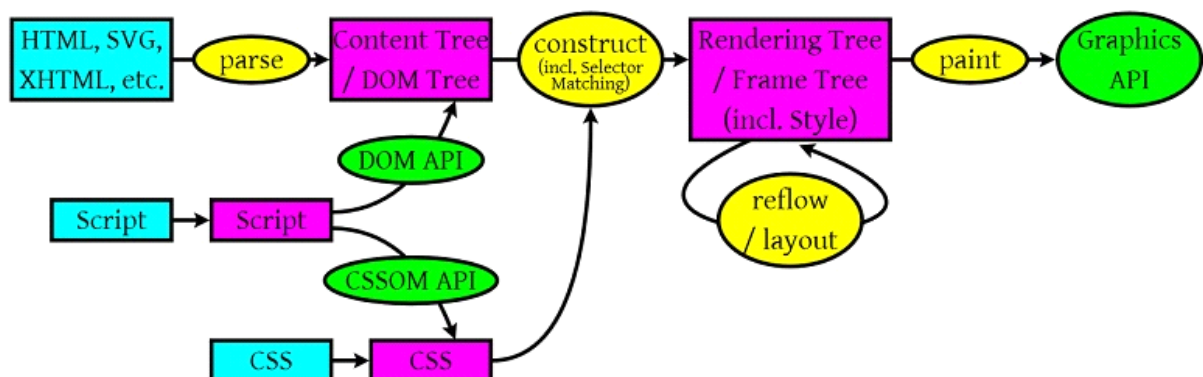
CSS Parser 可以理解为一个内部过程，而插件程式主要体现在：

postcss([autoprefixer])

- SASS等工具：源代码 -> 生产环境 CSS
- PostCSS：源代码 -> 标准 CSS -> 生产环境 CSS

PostCSS & Less结合使用方法：<http://www.w3cplus.com/PostCSS/using-postcss-together-with-sass-stylus-or-less.html>

? 页面渲染原理与性能优化



1. 浏览器解析这三个东西：

- 解析HTML/XHTML/SVG，生成DOM树
- 解析css文件产生CSS Rule树（css规则树）。
- 解析javascript，通过DOM API和CSSOM API来操作DOM树和CSS Rule树

2. 解析完成后，浏览器会根据DOM树和CSS Rule树来构造渲染树（ Rendering Tree ）。
 - 渲染树并不完全等同于DOM树，因为一些display:none的东西就没必要放在渲染树中了。
 - CSS Rule树主要是为了完成匹配并把CSS Rule附加到渲染树上的每个DOM结点。
 - 然后，计算每个DOM节点的位置，这又叫layout和reflow过程。
3. 最后通过调用操作系统Native GUI的API绘制（ painting ）

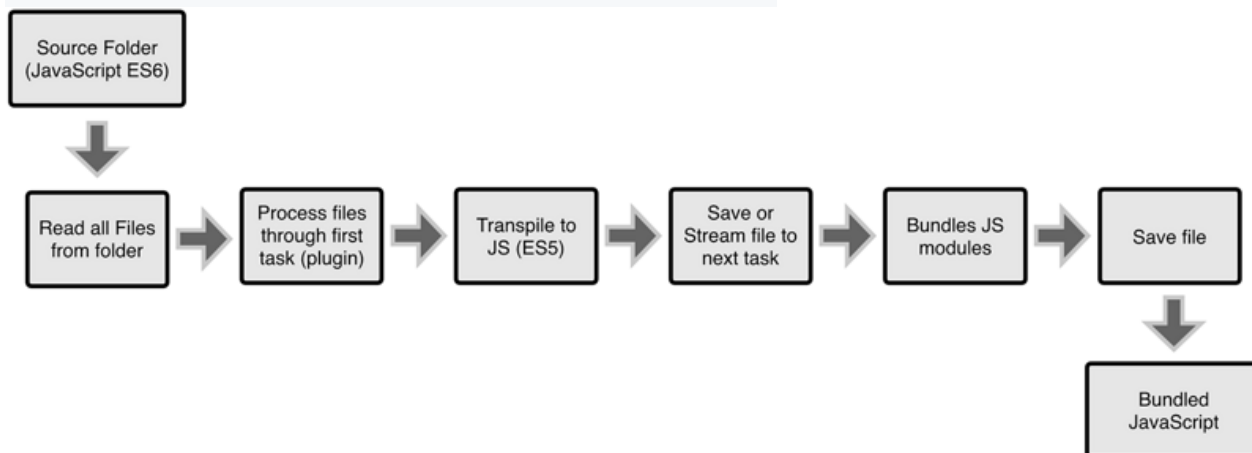
? 抽离样式模块

webpack <https://segmentfault.com/a/1190000006178770#articleHeader1> WebPack可以看做是模块打包机：分析你的项目结构，找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言（ Scss， TypeScript等 ），并将其打包为合适的格式以供浏览器使用。它把所有的文件都可以当做模块处理，包括JavaScript代码，CSS和fonts以及图片等，只有通过合适的loaders，它们都可以被当做模块被处理。

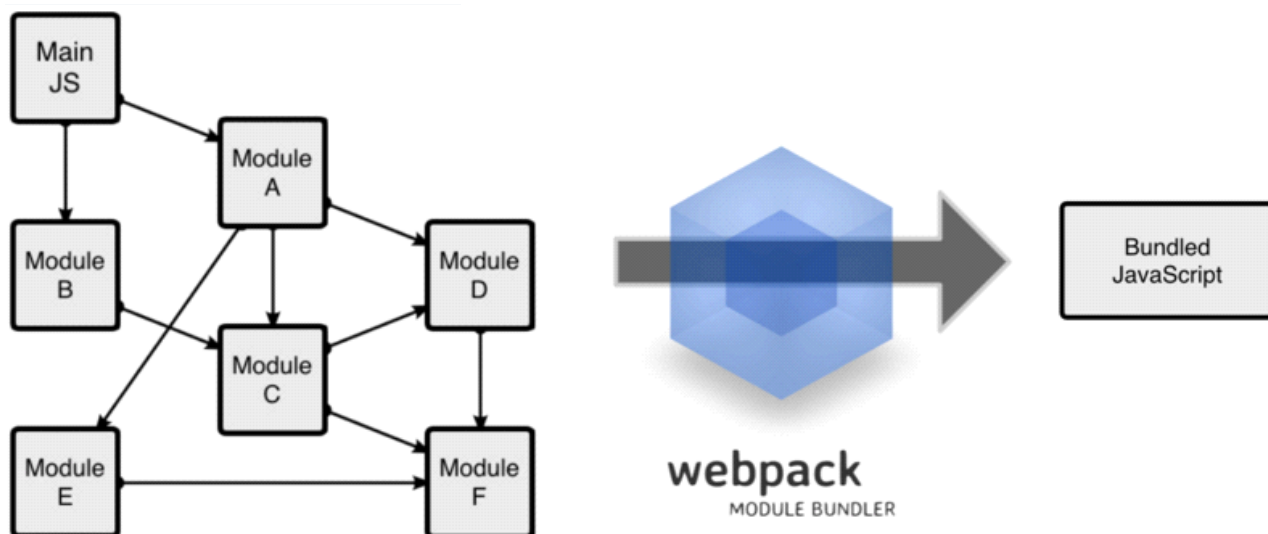
Webpack和Grunt以及Gulp相比有什么特性？

Gulp/Grunt是一种能够优化前端的开发流程的工具，而WebPack是一种模块化的解决方案。Webpack的处理速度更快更直接，能打包更多不同类型的文件。

Grunt和Gulp的工作方式是：在一个配置文件中，指明对某些文件进行类似编译，组合，压缩等任务的具体步骤，这个工具之后可以自动替你完成这些任务。



Webpack的工作方式是：把你的项目当做一个整体，通过一个给定的主文件（如：index.js），Webpack将从这个文件开始找到你的项目的所有依赖文件，使用loaders处理它们，最后打包为一个浏览器可识别的JavaScript文件。



通过配置文件使用webpack

```
//webpack.config.js
module.exports = {
  entry: __dirname + "/app/main.js", //已多次提及的唯一入口文件
  output: {
    path: __dirname + "/public", //打包后的文件存放的地方
    filename: "bundle.js" //打包后输出文件的文件名
  }
}
```

// “__dirname”是node.js中的一个全局变量，它指向当前执行脚本所在的目录。

webpack提供两个组件处理CSS：

- ❖ **css-loader**：使你能够使用类似@import 和 url(...)的方法实现 require()的功能
- ❖ **style-loader**：将所有的计算后的样式加入页面中

webpack应用

- ❖ 生成source Maps（对应编译文件和源文件的方法）——配置devtool
- ❖ 构建本地服务器（基于node.js）——webpack-dev-server组件
- ❖ Loaders——json-loader组件，配置webpack.config.js下的modules
webpack通过调用外部的脚本或工具可以对各种各样的格式的文件进行处理

? 元素竖向的百分比设定是相对于容器的宽度

? 全屏滚动的原理

方法1：JS获取鼠标滚动事件

```
/** This is high-level function.
```

```

    * It must react to delta being more/less than zero.
    */
function handle(delta) {
    if (delta < 0)
        ...;
    else
        ...;
}

/** Event handler for mouse wheel event.
 */
function wheel(event){
    var delta = 0;

    if (!event) { /* For IE. */

        event = window.event;

    }

    if (event.wheelDelta) { /* IE/Opera/Chrome/Safari. */

        /* .wheelDelta indicating the distance that the mouse wheel rolled*/
        delta = event.wheelDelta/120; //wheelDelta is always a multiple of 120.

    } else if (event.detail) { /** Mozilla case. */

        /** In Mozilla, sign of delta is different than in IE.
         * Also, delta is multiple of 3.
         */
        delta = -event.detail/3;

    }

    /** If delta is nonzero, handle it.
     * Basically, delta is now positive if wheel was scrolled up,
     * and negative, if wheel was scrolled down.
     */

    if (delta) {
        handle(delta);
    }

    /** Prevent default actions caused by mouse wheel.
     * That might be ugly, but we handle scrolls somehow
     * anyway, so don't bother here..
     */

    if (event.preventDefault) {
        event.preventDefault();
    }

    event.returnValue = false;
}

/** Initialization code.
 * If you use your own event management code, change it as required.
 */
if (window.addEventListener) {

    /** DOMMouseScroll is for mozilla. */
    window.addEventListener('DOMMouseScroll', wheel, false);

}

/** IE/Opera. */
window.onmousewheel = document.onmousewheel = wheel;

```

方法2：采用translateX属性控制容器位置

```
section.slideLeftPre{
```

```

        -webkit-animation: slideLeftPre 1s 1;
    }
    @-webkit-keyframes slideLeftPre{
        0%{
            -webkit-transform:translateX(0);
            opacity:1;
            z-index:2;
        }
        99%{
            -webkit-transform:translateX(-99%);
            opacity:1;
            z-index:2;
        }
        100%{
            -webkit-transform:translateX(-100%);
            opacity:0;
            z-index:2;
        }
    }
}

```

方法3：采用left属性控制容器位置

```

section.slideLeftPre{
    -webkit-animation: slideLeftPre 1s 1;
}
@-webkit-keyframes slideLeftPre{
    0%{
        left: 0;
        opacity:1;
    }
    99%{
        left: -99%;
        opacity:1;
    }
    100%{
        left: -100%;
        opacity:0;
    }
}
}
section.slideLeftNext{
    -webkit-animation: slideLeftNext 1s 1;
}
@-webkit-keyframes slideLeftNext{
    0%{
        left:100%;
        opacity:1;
        z-index:2;
    }
    100%{
        left:0;
        opacity:1;
        z-index:2;
    }
}
}

```

? 响应式设计 <https://isux.tencent.com/responsive-web-design.html>

响应式网络设计概念最初由 Ethan Marcotte 提出，目的是为移动设备提供更好的体验，并且整合从桌面到手机的各种屏幕尺寸和分辨率，用技术来使网页适应从小到大（现在到超大）的不同分辨率的屏幕。

- 响应式布局

1. 百分比布局

2. bootstrap栅格布局

3. Flex盒布局

- 引入viewport meta标签

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no">
```

- 通过media query设置样式

设备类型(media type):

all 所有设备
screen 电脑显示器
print 打印用纸或打印预览视图
handheld 便携设备
tv 电视机类型的设备
speech 语音和音频合成器
braille 盲人用点字法触觉反馈设备
embossed 盲文打印机
projection 各种投影设备
tty 使用固定密度字母栅格的媒介，比如电传打字机和终端

设备特性(media feature):

width 浏览器宽度
height 浏览器高度
device-width 设备屏幕分辨率的宽度值
device-height 设备屏幕分辨率的高度值
orientation 浏览器窗口的方向纵向还是横向，当窗口的高度值大于等于宽度时该特性值为portrait，否则为landscape
aspect-ratio 比例值，浏览器的纵横比
device-aspect-ratio 比例值，屏幕的纵横比

```
/* for 480 px width screen */  
  
@media only screen (min-device-width:321px) and (max-device-width:480px) {  
    selector { ... }  
}
```

IE6~8响应式布局的兼容问题——respond.js文件

<http://www.cnblogs.com/xcsn/p/5586859.html>

<https://github.com/scottjehl/Respond>

respond.js原理：

第一步，将head中所有外部引入的CSS文件路径取出来存储到一个数组当中；

第二步，遍历数组，并一个个发送AJAX请求；

第三步，AJAX回调后，分析response中的media query的min-width和max-width语法（注意，仅仅支持min-width和max-width），分析出viewport变化区间对应相应的css块；

第四步，页面初始化时和window.resize时，根据当前viewport使用相应的css块。

? ::before 和 :after中双冒号和单冒号的区别？作用？

单冒号：用于CSS3伪类（向选择器添加特殊的效果） : Pseudo-classes

双冒号：用于CSS3伪元素（将特殊的效果添加到某些选择器） : Pseudo-elements

before— 新内容+当前内容

after—当前内容+新内容

在代码顺序上，::after生成的内容也比::before生成的内容靠后。

如果按堆栈视角，::after生成的内容会在::before生成的内容之上

? 深度理解行高

行高：在应用到一个块级元素时，它定义了该元素中基线之间的最小距离而不是最大距离。

normal

number: 此 $x * \text{font-size}$ = 行间距

length: 固定行间距

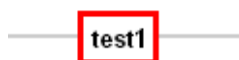
%: 基于 $x\% * \text{font-size}$ = 行间距

inherit: 继承父元素line-height属性值

行间距：line-height与font-size之差

line-height撑开了div的高度（一个没有设置height属性的div由line boxes的高度堆积而成）

行高的垂直居中特性：无论line boxes所占据的高度是多少（无论比文字大还是比文字小），其占据的空间都是与文字内容公用水平中垂线的。



行高的应用

- ❖ 单行文字的垂直居中对齐：line-height设置为box需要的大小

❖ 多行文字的垂直居中对齐

1. 高度不固定：使用padding

2. 高度固定：

```
div{
width:170px;
border:dashed1pxred;
line-height:300px;
}
div span{
display:inline-block;
vertical-align:middle;
line-height:1.4em;
}
<div>
<span>ilikeyou</span><br/>
<span>idon'tlikeyou</span>
</div>
```

❖ 大小不固定的图片的水平垂直居中

1. 透明GIF图片+背景定位

```


img{
background-position:center;
}
```

2. display: table-cell + 文字大小控制居中

```

```

3. display: inline-block + 文字大小控制居中

```
<div>
<a href="#zhangxinxu">  </a>
<a href="#zhangxinxu">  </a> </div>
a {
display: inline-block;
width:1.2em;
font-size:128px;
text-align:center;
vertical-align:middle;
}
a img{
vertical-align:middle;
```

}

❖ 在日志文章显示中的应用

px 的表示方法首先是要被淘汰的。因为文章里面的文字是有大有小的，使用px定值，由于继承性，无法实现根据文字大小自动调整间距，会出现大号文字重叠的现象。

normal默认在1~1.2行距，阅读吃力。

%有继承性，也不行。

解决方法：line-height: xx

? 图片格式问题 ? webp?

JPEG图片以24位颜色存储单个位图。JPEG是与平台无关的格式，支持最高级别的压缩，不过这种压缩是有损的。

PNG便携式网络图形（Portable Network Graphics）是一种无损压缩的位图图形格式^{1]}。其设计目的是试图替代GIF和TIFF文件格式，同时增加一些GIF文件格式所不具备的特性。

格式	压缩模式	交错支持	透明支持	动画支持
JPG	有损压缩	支持	不支持	不支持
PNG	无损压缩	支持	支持	不支持

JPG的特性

- 1、支持摄影图像或写实图像的高级压缩，并且可利用压缩比例控制图像文件大小。
- 2、有损压缩会使图像数据质量下降，并且在编辑和重新保存JPG格式图像时，这种下降损失会累积。

PNG的特性

- 1、能在保证最不失真的情况下尽可能压缩图像文件的大小。
- 2、PNG用来存储灰度图像时，灰度图像的深度可多到16位，存储彩色图像时，彩色图像的深度可多到48位，并且还可存储多到16位的α通道数据。
- 3、对于需要高保真的较复杂的图像，PNG虽然能无损压缩，但图片文件较大，不适合应用在Web页面上

可见在存储图像时采用JPG还是PNG主要依据图像上的色彩层次和颜色数量进行选择。一般层次丰富颜色较多的图像采用JPG存储，而颜色简单对比强烈的则需要采用PNG。但也会有一些特殊情况，例如有些图像尽管色彩层次丰富，但由于图片尺寸较小，上面包含的颜色数量有限时，也可以尝试用PNG进行存储。而有些矢量工具绘制的图像由于采用较多的滤镜特效也会形成丰富的色彩层次，这个时候就需要采用JPG进行存储了。

WebP

webP是一种支持有损压缩和无损压缩的图片文件格式，派生自图像编码格式 VP8，Fancy 采样算法。

webP & PNG对比：

WebP 在解码时间与 PNG 有较明显差异（毫秒级别）之外，总体使用体验和 PNG 基本无差异。但webP压缩率比PNG高出许多。

webP特征：

- 图片规格：16n * 16n
- WebP 格式的图片不会存在毛边问题
- 色彩数
 - 小于 256 色：以图标，图形，剪贴画为代表，最适合采用 WebP 无损压缩，精细度完美，体积大幅减少；
 - 大于 256 色：以多数表情图，广告图为代表，最适合采用 WebP 有损压缩，选择较高压缩比（建议压缩质量为 100% ~ 75%）
 - 远大于 256 色：以风景照，视频截图为代表，最适合采用 WebP 有损压缩，选择适中压缩比（建议压缩质量为 75% 以下）

来自 <<http://isux.tencent.com/introduction-of-webp.html>>

? 移动端布局相关问题

- overflow: scroll在IOS上滑动卡顿问题怎么处理？

`-webkit-overflow-scrolling: touch`

💡 视差滚动效果

视差滚动则是在滚动的时候，内容和多层次的背景以不同速度或不同方向的移动，形成立体的运动效果。

<http://justcoding.iteye.com/blog/2010032>

<http://jonathannicol.com/blog/2011/08/06/build-a-parallax-scrolling-website-interface-with-jquery-and-css/>

背景层、贴图层、内容层

1. 原理

为每个展示层设置不同的z-index（值越大，浮于越上层）；

在js脚本中为每个展示层设置css的top属性值（与起始位置相比鼠标滚动的相对距离*比率）以对应不同的

滚动速度。

```
function parallaxScroll(){  
    // 利用css 中的top 属性移动每一个展示层  
    let scrolled=$(window).scrollTop();  
    $('#parall-bg1').css('top',(0-scrolled*0.75)+'px');// 相减，视觉上屏幕向下滚动  
    $('#parall-bg2').css('top',(0-scrolled*0.5)+'px');  
    $('#parall-bg3').css('top',(0-scrolled*0.25)+'px');  
    $('#parall-bg4').css('top',(0-scrolled*0.05)+'px');  
}
```

2. 插件

[Scrollorama](#)

[curtain.js](#) 类似于幕布升起的效果

[jQuery-Parallax](#)

[stellar.js](#)

[iparallax](#)

[Skrollr](#)

[Parallax.js](#)

[A Simple Parallax Scrolling Technique via Nettuts+](#)

[Parallax Slider](#)

3. 教程

[Javascript视差效果](#)

[Nike Better World场景的背后](#) 一份为那些急切想重现NIKE网站滚动效果的设计师准备的教程。

[用jQuery和CSS构建一个具有视差滚动效果的网站界面](#) 一份为那些想在此领域了解的更深的设计师准备的例子和教程。

[用jQuery创建一个时尚的具有视差背景的效果](#) 一份教程，在背景上帮你添加一些滚动的云彩。

[动画背景式的网页头部](#) 用一个动画式的网页头部来让你的访问者为你欢呼吧！

[视差滚动教程](#) 一份视差效果和内容同时出现的有趣的教程。



如何修改chrome记住密码后自动填充表单的黄色背景？

```
input:-webkit-autofill, textarea:-webkit-autofill, select:-webkit-autofill{  
    background-color:darkorange;  
    color:rgb(0,0,0);  
}
```



居中div

- 水平居中

```
div{  
    position:static;  
    width:200px;  
    height:200px;  
    margin:0 auto;
```

```

}
让绝对定位的div水平居中
div{
    position:absolute;
    width:200px;
    height:200px;
    background-color:pink;
    top:0;
    left:0;
    bottom:0;
    right:0;
    margin:auto;
}

```

- 水平垂直居中

```

div{
    position:absolute;
    width:400px;
    height:200px;
    background-color:pink;
    top:50%; /*top_percent=子盒子顶部距父盒子顶部的距离/父盒子高度*/
    left:50%;
    margin:-100px 0 0 -200px; /*外面距为自身高度一半*/
}

```

方法2：

```

body{
    height:600px;
    display:flex;
    justify-content:center; /*水平居中*/
    align-items:center; /*垂直居中*/
}
div{
    width:100px;
    height:100px;
    background-color:pink;
}

```



css绘制三角形

```

width:0px;
height:0px;
border-width:20px;
border-style:solid;
border-color:transparent transparent red transparent; //向上的三角形

```



css绘制圆形

```

width:10px;

```

```
height:10px;
border-radius:10px;
border-color:red;
border-style:solid;
```



div剩余高度自动填充

有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度。

```
<div id="div1">
  <div id="div2"></div>
  <div id="div3"></div>
</div>
```

方法1：absolute position

```
#div1{
width:300px;
height:300px;
border:3px solid black;
position:relative;
}
#div2{
width:100%;
height:100px;
background-color:darkorange;
}
#div3{
width:100%;
position:absolute;
top:100px;
left:0;
bottom:0;
background-color:pink;
}
```

方法2：padding&margin相抵

```
#div1{
width:300px;
height:300px;
padding-top:100px;
border:3px solid black;
}
#div2{
width:100%;
height:100px;
margin-top:-100px;
background-color:darkorange;
}
#div3{
width:100%;
height:100%;
background-color:pink;
}
```


? JavaScript各种规范 <http://blog.csdn.net/luoxinwhu/article/details/77619318>

? 原型链

► prototype和__proto__的区别



* prototype是函数才有的属性 * __proto__是每个对象都有的属性

* 但__proto__不是一个规范属性，
只是部分浏览器实现了此属性，对
应的标准属性是[[Prototype]]

注：大多数情况下，__proto__可以理解为“构造器的原型”，即：
__proto__ === constructor.prototype
(通过Object.create()创建的对象不适用此等式，图2有说明)

@水乙

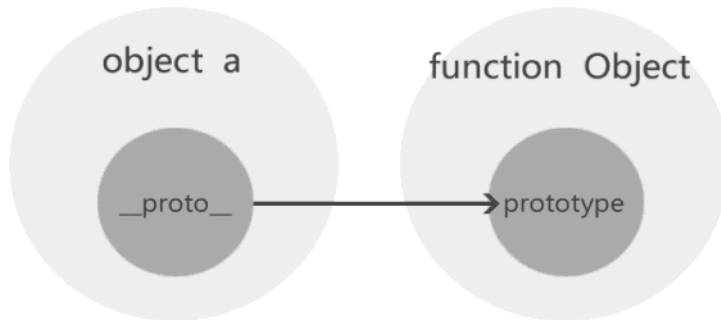
```
var a = {};  
console.log(a.prototype); //undefined  
console.log(a.__proto__); //{  
  
var b = function(){};  
console.log(b.prototype); //b {  
console.log(b.__proto__); //undefined
```

► __proto__属性指向谁？

__proto__的指向取决于对象创建时的实现方式。以下图表列出了三种常见方式创建对象后，__proto__分别指向谁。

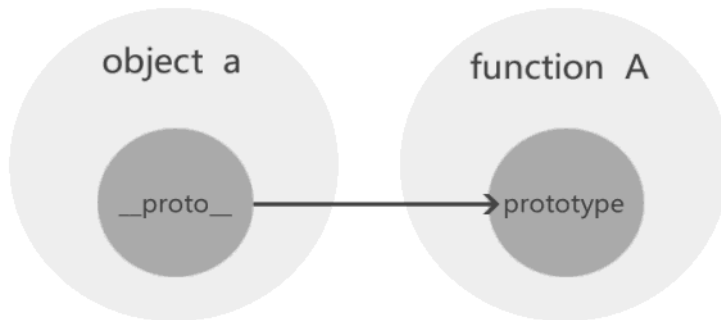
1、字面量方式

```
var a = {};
```



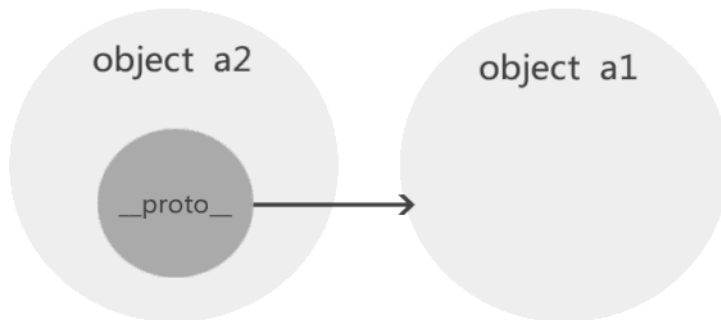
2、构造器方式

```
var A = function(){};  
var a = new A();
```



3、Object.create方式

```
var a1 = {}  
var a2 = Object.create(a1);
```



@水乙

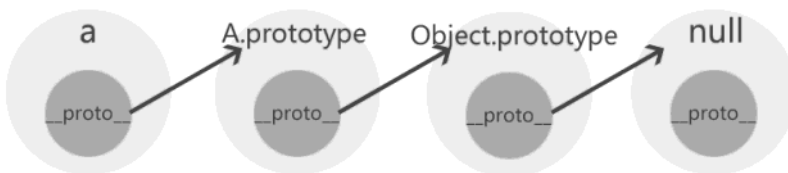
```
var a2 = {};  
console.log(a2.__proto__ === Object.prototype); //true  
console.log(a2.__proto__ === a2.constructor.prototype); //true  
  
var A = function() {};  
var a1 = new A();  
console.log(a1.__proto__ === A.prototype); //true  
  
var a3 = Object.create(a2);  
console.log(a3.__proto__ === a2); //true
```

► 什么是原型链？

由于__proto__是任何对象都有的属性，而js里万物皆对象，所以会形成一条__proto__连起来的链条，递归访问__proto__必须最终到头，并且值是 null。

当js引擎查找对象的属性时，先查找对象本身是否存在该属性，如果不存在，会在原型链上查找，但不会查找自身的prototype

```
var A = function(){};
var a = new A();
```



@水乙

instance.constructor.prototype = instance.__proto__

```
var A = function() {};
var a1 = new A();
console.log(a1.__proto__ === A.prototype); //true
console.log(a1.__proto__.__proto__ === Object.prototype); //ture
console.log(a1.__proto__.__proto__.__proto__ === null); //true
```

? JavaScript如何实现继承 <http://blog.csdn.net/luoxinwhu/article/details/77267275>

- 构造继承
- 原型继承
- 实例继承
- 拷贝继承

? javascript作用链域

全局作用域

1. 最外层函数和再最外层函数外面定义的变量
2. 所有未定义直接赋值的变量
3. window对象的属性（例如：window.name window.top）

```
let a1 = 23; //全局作用域
console.log(a1); //23

let a3; //全局作用域
console.log(a3); //undefined
```

局部作用域：局部作用域一般只在固定的代码片段内可访问到。最常见的在函数内部。

函数作用域是在函数定义的时候确定的，而不是在函数执行的时候确定的，所以这种作用域往深了说叫词法作用域。


```
function A() {
  // 函数A的作用域
  let a2 = 200;
  function B() {
    //函数B的作用域
    console.log('in the func A()' + a2);
  }
  B();
}
console.log(a2); //ReferenceError: a2 is not defined
```

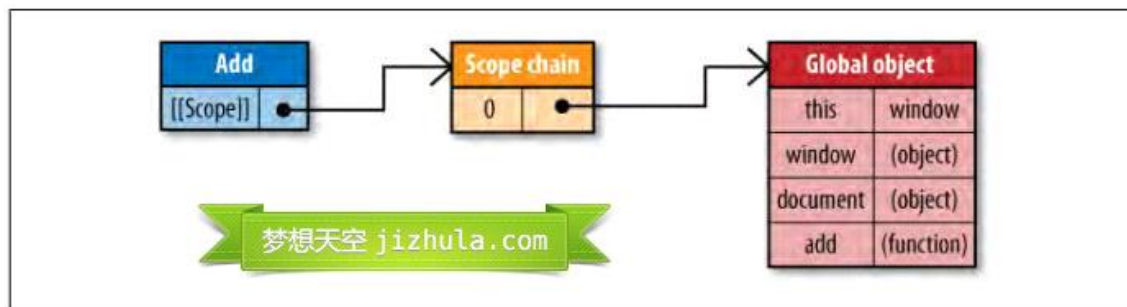
作用域链

全局函数无法查看局部函数的内部细节，但局部函数可以查看其上层的函数细节，直至全局细节。
当需要从局部函数查找某一属性或方法时，如果当前作用域没有找到，就会上溯到上层作用域查找，直至全局函数，这种组织形式就是作用域链。

当一个函数创建后，它的作用域链会被创建此函数的作用域中可访问的数据对象填充。例如定义下面这样一个函数：

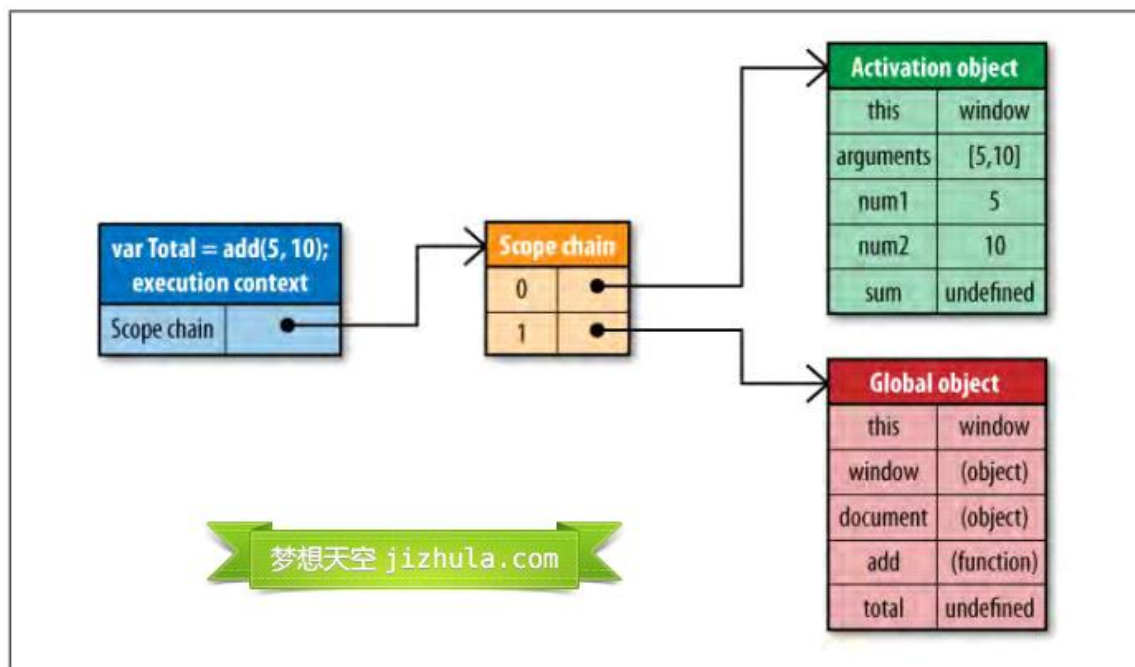
```
1 function add(num1,num2) {
2     var sum = num1 + num2;
3     return sum;
4 }
```

Function对象有一个内部属性Scope，该内部属性包含了函数被创建的作用域中对象的集合，这个集合被称为函数的作用域链。



```
1 var total = add(5,10);
```

执行此函数时会创建一个称为“运行期上下文(execution context)”的内部对象



从作用域链的结构可以看出，在运行期上下文的作用域链中，标识符所在的位置越深，读写速度就会越慢。如上图所示，因为全局变量总是存在于运行期上下文作用域链的最末端，因此在标识符解析的时候，查找全局变量是最慢的。所以，在编写代码的时候应尽量少使用全局变量，尽可能使用局部变量。

作用域优化：如果一个跨作用域的对象被引用了一次以上，则先把它存储到局部变量里再使用。

```
function changeColor(){
    document.getElementById("btnChange").onclick=function(){
        document.getElementById("targetCanvas").style.backgroundColor="red";
    };
}
```

```
1 function changeColor(){
2     var doc=document;
3     doc.getElementById("btnChange").onclick=function(){
4         doc.getElementById("targetCanvas").style.backgroundColor="red";
5     };
6 }
```

! 作用域被篡改

1. with()语句
2. try-catch语句，但可以通过优化减小catch语句对性能的影响

```
try{
    doSomething();
}catch(ex){
    handleError(ex); //委托给处理器方法
}
```

? **this对象的理解** <http://blog.csdn.net/luoxinwhu/article/details/77334235>

? **闭包问题** <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Closures>

? 脚本注入问题

❖ js脚本注入

转义

过滤

❖ HTML入侵

html转义：将特殊字符或HTML标签转换为与之意思相对应的字符

< 转义为 < > 转义为 >

<script>alert('attack');</script> 转义为 <script>alert(' attack');

</script>

❖ SQL注入

SpringMVC利用拦截器 <http://www.imooc.com/article/6137>、<http://blog.csdn.net/bcqtt/article/details/51670457>

? eval

本质：是一个函数

功能：动态判断源代码中的字符串，并产生一个值

危险：如果一个函数调用了eval，那么解释器将无法对这个函数做进一步优化

使用方式：eval(str) -

如果传入的参数不是字符串，它直接返回这个函数。如果参数是字符串，它会把字符串当成JavaScript代码进行编译，如果编译失败者抛出一个语法错误异常。如果编译成功，则开始执行这一段代码，并返回字符串中的最后一个表达式或语句的值，如果最后一个表达式或语句没有值，则最终返回undefined。如果字符串抛出一个异常，这个异常将把该调用传递给eval()。

? JavaScript内存泄漏

不再用到的内存，没有及时释放，就叫做内存泄漏。大多数语言提供自动内存管理，减轻程序员的负担，这被称为"垃圾回收机制" (garbage collector)。

垃圾回收机制怎么知道，哪些内存不再需要呢？

最常使用的方法叫做“引用计数” (reference counting)：语言引擎有一张"引用表"，保存了内存里面所有的资源（通常是各种值）的引用次数。如果一个值的引用次数是0，就表示这个值不再用到了，因此可以将这块内存释放。

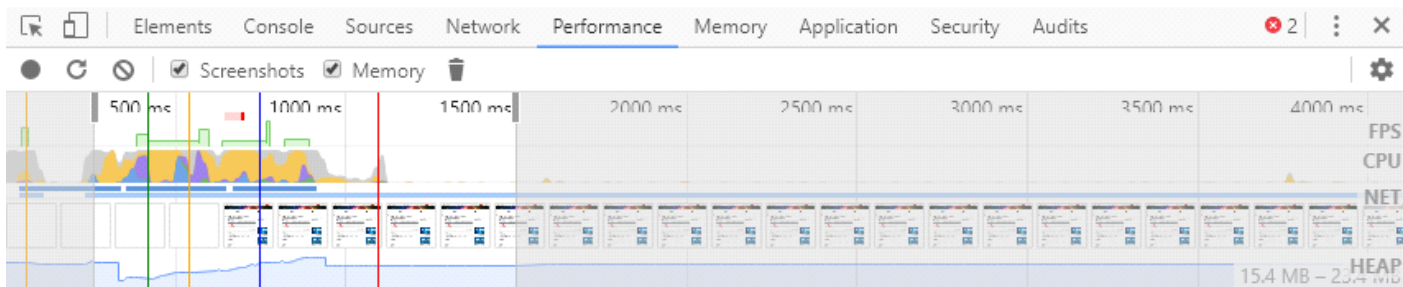
怎样可以观察到内存泄漏呢？

如果连续五次垃圾回收之后，内存占用一次比一次大，就有内存泄漏。这就要求实时查看内存占用。

- 使用Chrome浏览器可以查看内存使用情况：

1. 打开开发者工具，选择Performance 面板
2. 在顶部的勾选 Memory
3. 点击左上角圆形的录制按钮。
4. 在页面上进行各种操作，模拟用户的使用情况。
5. 一段时间后，点击对话框的 stop 按钮，面板上就会显示这段时间的内存占用情况。

如果内存占用基本平稳，接近水平，就说明不存在内存泄漏。



- 使用cmd命令行

```
C:\Users\LX>node
> console.log(process.memoryUsage());
{ rss: 22278144,
  heapTotal: 8425472,
  heapUsed: 4024168,
  external: 9039 }
undefined
```

rss (resident set size) : 所有内存占用, 包括指令区和堆栈。

heapTotal : "堆"占用的内存, 包括用到的和没用到的。

heapUsed : 用到的堆的部分。

external : V8 引擎内部的 C++ 对象占用的内存。

清除内存的方法?

ES6推出两种数据结构 **WeakSet**, **WeakMap**

```
const wm = new WeakMap();

const element = document.getElementById('example');

wm.set(element, 'some information');
wm.get(element) // "some information"
```

? JS延迟加载

- setTimeout(funcName, delayTime)
- 在body的最后引入JS文件
- async / defer

<script>, <script async>, <script defer>的区别?

通过script标签插入javascript代码。第一种直接在script标签之间插入js代码, 第二种即是通过src属性引入外部js文件

<script> 立即加载, 立即指向

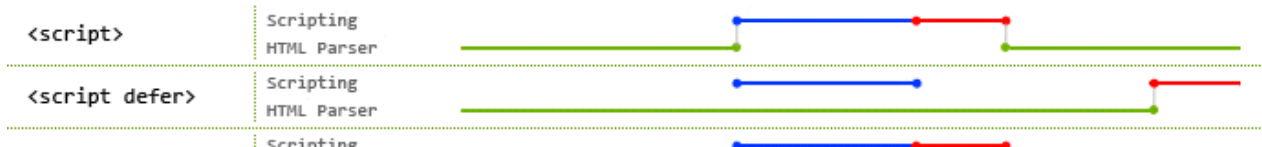
浏览器会立即加载并执行相应的脚本。也就是说在渲染script标签之后的文档之前, 不等待后续加载的文档元素, 读到就开始加载和执行, 此举会阻塞后续文档的加载;

<script async> 同步加载, 立即执行

有了async属性, 表示后续文档的加载和渲染 & js脚本的加载和执行是并行进行的, 即异步执行;

<script defer> 同步加载, 顺序执行

有了defer属性, 加载后续文档的过程 & js脚本的加载(此时仅加载不执行)是并行进行的(异步), js脚本的执行需要等到文档所有元素解析完成之后, DOMContentLoaded事件触发执行之前。





？ 如何避免用户多次点击造成的多次请求

仅仅需要对某些可能具有复杂后台业务逻辑、或者文件上传、或者调用其他非本工程接口导致网络延迟等等情况需要去做“防止重复点击提交”。

1. 定位标志位：点击触发请求后，标志位为false量；请求（或者包括请求后具体的业务流程处理）后，标志位为true量。通过标志位来判断用户点击是否具备应有的响应
2. 卸载及重载绑定事件：点击触发请求后，卸载点击事件；请求（或者包括请求后具体的业务流程处理）后，重新载入绑定事件
3. 替换（移除）按钮DOM：点击触发请求后，将按钮DOM对象替换掉（或者将之移除），自然而然此时不在具备点击事件；请求（或者包括请求后具体的业务流程处理）后，给新的按钮DOM定义点击事件。

```
$('.J-login').click(function () {
    var loginBtn = this;

    //1.先进行表单验证
    //.....

    //2.异步提交
    $.ajax({
        url: 'login.do',
        data: $(this).closest('form[name="loginForm"]').serialize(),
        type: 'post',
        beforeSend: function () {
            //3.让提交按钮失效，以实现防止按钮重复点击
            $(loginBtn).attr('disabled', 'disabled');

            //4.给用户友好状态提示
            $(loginBtn).text('登录中...');
        },
        complete: function () {
            //5.让登陆按钮重新有效
            $(loginBtn).removeAttr('disabled');
        },
        success: function(msg){
            if (msg === 'ok') {
                alert('登录成功! ');

                //TODO 其他操作...
            } else {
                alert('登录失败，请重新登录! ');
            }
        }
    });
});
```

4. 设定请求额度：允许它重复点击（或者其他用户事件），但是不允许在一定的时间内超过次数XX次

```

var timer = null;

$(input).keyup(function(){
    var value = $(this).val();

    clearTimeout(timer);

    //如果键盘敲击速度太快，小于100毫秒的话就不会向后台发请求，但是最后总会进行一次请求的。
    timer = setTimeout(function() {
        //触发请求
        $.ajax({
            url: 'typeahead.do',
            type: 'get',
            data: value,
            success: function () {
                //显示匹配结果
                //.....
            }
        });
    }, 100);
});

```

? js编码和页面编码不一致

比如：http://www.yyy.com/a.html 中嵌入了一个http://www.xxx.com/test.js。a.html 的编码是gbk或gb2312的。而引入的js编码为utf-8的，那就需要在引入的时候<script src="http://www.xxx.com/test.js" charset="utf-8"></script> 同理，如果你的页面是utf-8的，引入的js是gbk的，那么就需要加上charset="gbk".

? JS模块化开发

1. 用对象封装模块：缺点是暴露了所有模块成员，内部状态易被外部改写

```

var module1 = new Object({

    _count : 0,

    m1 : function (){
        //...
    },

    m2 : function (){
        //...
    }

});

```

2. 立即执行函数

```

var module1 = (function(){

    var _count = 0;

    var m1 = function(){
        //...
    };

    var m2 = function(){
        //...
    };

    return {
        m1 : m1,
        m2 : m2
    };

})();

```

3. 放大模式（模块很大的情况，或模块之间有相互调用关系）

```

var module1 = (function (mod){

    mod.m3 = function () {
        //...
    };

    return mod;

})(module1);

```

4. 宽放大模式

```

var module1 = ( function (mod){

    //...

    return mod;

})(window.module1 || {});

```

5. 输入全局变量

```

var module1 = (function ($, YAHOO) {

    //...

})(jQuery, YAHOO);

```

上面的module1模块需要使用jQuery库和YUI库，就把这两个库（其实是两个模块）当作参数输入module1。这样做除了保证模块的独立性，还使得模块之间的依赖关系变得明显。

? ajax解决浏览器缓存问题

在载入数据的同时将数据缓存到内存中，一旦数据被加载其中，只要我们没有刷新页面，这些数据就会一直被缓存在内存中，当我们提交的URL与历史的URL一致时，就不需要提交给服务器，也就是不需要从服务器上面去获取数据，虽然这样降低了服务器的负载提高了用户的体验，但是我们不能获取最新的数据。为了保证我们读取的信息都是最新的，我们就需要禁止他的缓存功能。

```
1. xmlhttp.setRequestHeader("if-Modified-Since","0");//浏览器页面最后修改时间
2. xmlhttp.setRequestHeader("Cache-Control","no-cache");
3. url 后面加上一个时间戳 "nowtime="+new Date().getTime()
4. url 后面加上一个随机数 'fresh=" + Math.random()
5. $.ajaxSetup({cache:false});
```

? null 和 undefined的区别

```
let a = undefined;
console.log(Number(a)); //NaN

let b = null;
console.log(Number(b)); //0
```

```
x; // 未定义的变量 ReferenceError: x is not defined*
console.log(typeof null); // object
console.log(typeof undefined); // undefined
console.log(typeof x); // 声明却未赋值的变量，undefined
```

null：表示“没有对象”，即该处不应该有值，转为数值时为0

典型用法：

a. 作为函数的参数，表示该函数的参数不是对象

b. 作为对象原型链的终点 `console.log(Object.getPrototypeOf(Object.prototype)); //null`

undefined：表示“缺少值”，即此处本应该有值，但还没有赋值，转为数值时为NaN

典型用法：

a. 变量声明，还未赋值（即未定义变量）

b. 调用函数时，应该提供的参数没有提供，该参数等于undefined

c. 对象没有赋值的属性，该属性值为undefined

d. 函数没有返回值，默认返回undefined


```

var i;
i // undefined

function f(x){console.log(x)}
f() // undefined

var o = new Object();
o.p // undefined

var x = f();
x // undefined

```

```

1 console.log(NaN===NaN); // false
2 console.log(NaN==NaN); // false
3 console.log(null===null); // true
4 console.log(null == undefined); // true
5 console.log(null === undefined); // false
6 console.log(null == 0); // false
7 console.log(undefined == 0); // false
8 console.log(false === 0); // false
9 console.log([1,2]==[1,2]); // false
10 console.log(1=='1'); // true
11 console.log(1==='1'); // false

```

```

console.log(typeof (1+'2')); // string
console.log(typeof null); // object
console.log(typeof undefined); // undefined
console.log(typeof x); // undefined, 声明但未赋值的变量
var y;
console.log(typeof y); // undefined
x; // ReferenceError: x is not defined

```

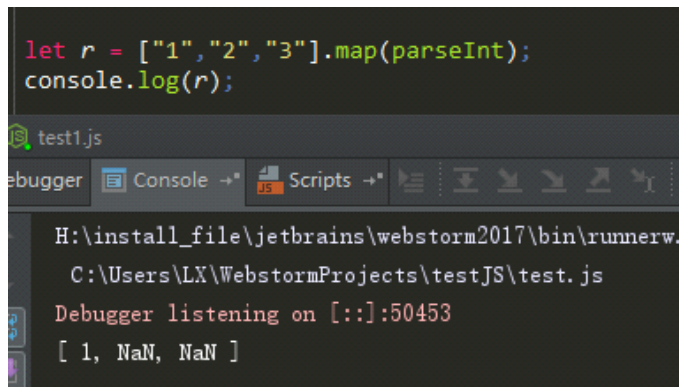
? .call() & .apply()

call(thisObj, arg1, ...) 参数逐个传入

apply(thisObj, argArr) 参数以数组的方式传入

? 为什么 ["1","2","3"].map(parseInt) 返回结果不是 [1,2,3]

```
let r = ["1","2","3"].map(parseInt);
console.log(r);
```



H:\install_file\jetbrains\webstorm2017\bin\runnerw.
C:\Users\LX\WebstormProjects\testJS\test.js
Debugger listening on [::]:50453
[1, NaN, NaN]

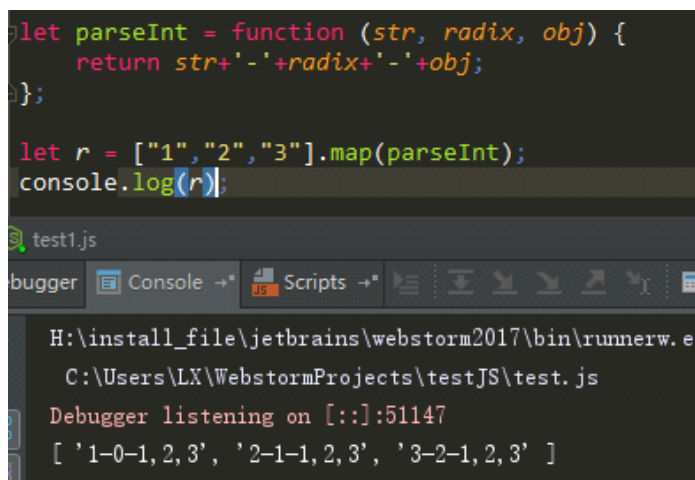
map函数使用规则 `array.map(callback)`
callback包含3个参数 (value, index, array1)
那么对于此题而言

value	1	2	3
index	0	1	2
array1	1,2,3	1,2,3	1,2,3

此题中`map` 函数将数组的值 `value` 传递给了 `parseInt` 的`value` , 将`index`传递给了`parseInt`的`radix` , 将`array1`也传递给了`parseInt` , 而`parseInt`函数只接收2个参数 (`val`, `radix`)

```
let parseInt = function (str, radix, obj) {
    return str+'-'+radix+'-'+obj;
};

let r = ["1","2","3"].map(parseInt);
console.log(r);
```



H:\install_file\jetbrains\webstorm2017\bin\runnerw.e
C:\Users\LX\WebstormProjects\testJS\test.js
Debugger listening on [::]:51147
['1-0-1, 2, 3', '2-1-1, 2, 3', '3-2-1, 2, 3']

? Polyfill

★ JavaScript基本规范

1. 命名规范
 - 变量：小驼峰式，前缀尽量是名词

- 常量：全大写
- 函数：小驼峰式，前缀尽量是动词
- 构造函数：大驼峰式
- 类
 - 公共属性和方法：和变量和函数命名规则一致
 - 私有属性和方法：前缀为_（下划线）

2. 注释规范

- 函数注释
 - @param
 - @return
 - @author
 - @version
 - @example

★ **浏览跨域访问** <http://blog.csdn.net/luoxinwhu/article/details/77618976>

★ **JSON (Javascript object notation)** <http://blog.csdn.net/luoxinwhu/article/details/77413462>

★ **Ajax (Asynchronous Javascript And XML)**

<http://www.runoob.com/ajax/ajax-tutorial.html>

核心是 [XMLHttpRequest](#) 对象

```
if(window.XMLHttpRequest) {
    xmlhttp = new XMLHttpRequest();
}
else {
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}

xmlhttp.onreadystatechange = function () {
    if(xmlhttp.readyState==4 && xmlhttp.status==200) {
        document.getElementById('AJAX_msg').innerHTML = xmlhttp.responseText;
    }
};

xmlhttp.open("GET", "getHint.php?q=" + str, true); //true表示使用ajax
xmlhttp.send();
```

💡 **数组的随机排序**

```
6   let arr=[1,2,3,4,5,6,7];
7
8   function randSort(arr) {
9     for(let i=0, len=arr.length; i<len; i++) {
10      /* parseInt()向下取整
11       * random()函数返回0~1之间的随机数*/
12      let rand = parseInt(Math.random()*len);
13      let temp = arr[rand];
14      arr[rand] = arr[i];
15      arr[i] = temp;
16    }
17    return arr;
18  }
19
20  console.log(randSort(arr));
21
22
```

Run test1.js

H:\install_file\jetbrains\webstorm2017\bin\runnerw.exe H:\in

[5, 2, 7, 3, 4, 6, 1]

Process finished with exit code 0

```
function randSort2(arr) {
  let mixedArr = [];
  while(arr.length>0) {
    let rand = parseInt(Math.random()*arr.length);
    mixedArr.push(arr[rand]);
    arr.splice(rand, 1); //从rand位置开始删除1个元素
  }
  return mixedArr;
}

console.log(arr);
console.log(randSort2(arr));
```

test1.js

H:\install_file\jetbrains\webstorm2017\bin\runnerw.exe H:\inst

[1, 2, 3, 4, 5, 6, 7]

[5, 3, 6, 4, 7, 1, 2]

```
console.log(arr);
arr.sort(function () {
  /**
   * sort()方法: 为数组排序
   * 接受一个方法作为参数, 当函数的返回值为1时交换传入的两个数组元素的顺序
   */
  return Math.random()-0.5;
});
console.log(arr);
```

test1.js

H:\install_file\jetbrains\webstorm2017\bin\runnerw.exe H:\install_file\nodej

[1, 2, 3, 4, 5, 6, 7]

[2, 6, 7, 3, 1, 4, 5]

布局专题

2017年9月17日 9:35

★ 单列布局

- 水平居中

STAR法则

2017年8月17日 20:18

Situation情境：HR希望你描述一个你遇到的挑战、项目。（曾经完成的某件重要的且可以当作考评标准的任务的背景状况。比较绕口好好理解下）

Task任务：你当时的任务是什么？HR想知道你是怎样在上述情境下明确自己的任务的。

Action行动：你做了什么？HR要知道你为上边的任务所付出的行动，为什么这么做？还有其他方案吗？

Result结果：结果怎样？从你的行动中，得到了什么？有没有完成目标呢？你获得了那些经验教训呢？之后有没有再用到那些经验呢？

来自 <<http://www.jianshu.com/p/d833c4cd4f46>>

八大问

2017年9月16日 星期六 09:10

1. Describe an instance where you set your sights on a high/demanding goal and saw it through completion. 描述一个例子，证明你给自己确立了一个很高的目标，然后完成了这个目标。

问题分析：考查意图：制订高目标的勇气 + 完成高目标的执行力。关键词：demanding goal + saw it through. 所以，在描述的时候要着重描述这个任务为什么这么demanding，有些什么具体的困难，你是怎么样一步一步去克服的。

What：以211院校背景争取到武汉大学的推免名额

Situation (demanding)：在我大四那年，是开放外校推免政策的第二年。部分知名高校通过举办夏令营的方式提前发放一部分推免的名额。与我而言，困难之处在于我需要以211的院校背景和众多名校本校的优秀学生同时竞争这些少量的推免名额。并且众多高校只给专业第一的985院校学生发放推免名额，而我当时的排名是专业第三。

Task：在我接到复试通知那天起，距离复试只有13天。在此期间，我需要准备2门期末考试，以及复试可能会考察到的专业课。

Action：

1. 首先，我上网收集了武汉大学在2014年夏令营复试中考察的内容，判断出这些知识属于哪些专业课程。整理出需要复习的专业课程书录。
2. 其次，结合武汉大学历年考研大纲，我将这些专业课程按照考试范围比重做了排序。并根据自身的掌握情况为每门课程以及各个科目的重点单元设置了具体的复习时间。
3. 为了兼顾期末考试，对于这13天我做了详细的时间和内容分配的规划。为了保持记忆的连续性和做题的熟练度，利用每天上午某段固定时间复习期末考试科目，余时间用于复习复试科目。按照复试科目的重要性，我采用先重要后次要的顺序。鉴于时间紧迫，我决定第一轮复习以各个科目的重点章节为主，最大程度保障考试分数。

result：以复试专业排名第2拿到了推免的名额。

2. Summarize a situation where you took the initiative to get others to complete an important task, and played a leading role to achieve the results you wanted. 描述一个例子，你团结了一群人共同努力，并为取得成功起到了带头作用。

what：项目管理大赛

situation：

task

action：

result：

3. Describe a situation where you sought out relevant information, defined key issues, and decided on which steps to take to get the desired results. 描述一个例子，你找到相关信息，确定重点事项，然后决定了采取哪些步骤来获得满意的结果。

问题分析：考查意图：分析—判断—决策能力。关键词：seek out relevant information, define key issues, decide on the steps。面试官希望了解你分析处理一件事情的全过程。和刚才一个题目考查如何争取别人的协助是完全不同的。所以，你在描述的时候要重点描述你做整个事情的经过，而不是和什么人一起做。

What：武大复试过程中受到厦大夏令营通知

situation：

task：

action：

result：

4. Describe an instance where you made effective use of facts to secure the agreement of others. 描述一个例子，证明你用事实去说服别人。

点评：考查意图：以事实说服他人的能力。关键词：facts, secure agreements。所以，你的描述应该非常直截了当，先把大家的不同意见列出来，以及他人为什么不同意；然后逐条以事实进行说服。

5. Give an example of how you worked effectively with people to accomplish an important goal. 描述一个例子，证明你有效地与人合作，共同完成了一件重要的事情。

点评：考查意图：团队合作能力。关键词：worked effectively with people。所以，你的描述应该很清晰：我做什么+别人做什么。

What：

6. Describe a creative/innovative idea that you produced which significantly contributed to the success of an activity or project. 描述一个例子，你提供了一个创新的想法，使得某一个活动或者项目取得了成功。

问题分析：考查意图：创新能力。关键词是creative idea，significantly contributed。所以，你的描述要包括你的创新意识、创新依据、有了创新的想法之后如何去实现它，以及这个创新想法起到了多么重要的作用。

7. Provide an example of how you assessed a situation and achieved good results by focusing on the most important priorities. 描述一个例子，说明你如何评估形势，并且把精力放在最重要的事情上，然后取得了比较好的结果。

问题分析：这个问题的关键词是assessed a situation, focused on the priorities，考查的是你分析事情和找

到问题关键点的能力。所以，你的描述应该是着重于自己的分析过程。

(见1. *action*描述)

8. Provide an example of how you acquired technical skills and converted them to practical application. 描述一个例子，说明你如何获得了技能，并且把这些技能转化成实际应用。

问题分析：这个问题的关键词是convert technical skills to practical application，真正的考查意图是你的“学习能力”。

【个人博客搭建过程】

9. 你的优点和缺点？

自学能力（魔方）、时间规划能力（保研）、有
在待人处世方面不够圆滑，没有意识到需要适当夸奖他人

自我介绍

2017年6月28日 16:23

● 1分钟

大家好，我叫罗鑫，硕士就读于武汉大学电子与通信工程专业，本科毕业于福州大学，目前研究方向是嵌入式系统。曾连续2年荣获校一等奖学金。担任过校级协会主席和院级学生会的部长。曾在中国金融POS机领域的龙头企业联迪实习。我敢于大胆行动，具备明确清晰的长、短期目标。由于追求高效的执行力，有时会忽略一些既定问题的审查。职业发展目标是希望在未来的3-5年成为独当一面的部门经理。谢谢！

● 3分钟

尊敬的面试官，您好！我叫罗鑫，目前就读于武汉大学电子与通信工程硕士专业，研究方向是嵌入式系统。目前重点在做的一个项目是数据中心可靠性之内存错误预警研究。这个是我们实验室和华为2012杭州研究所合作的一个为期约2年的项目。我是这个项目组的骨干成员。对于网络相关基础知识我掌握的不错，我的本科毕业设计内容是基于IEEE802.11P协议的车载无线通信系统的实现，还被评为优秀毕业设计。除此之外，我的自学能力比较好，曾花费3个月从零开始自学前端开发相关技能并独立完成个人博客网页的搭建。

我是一个对未来有明确规划的人，从大一起我就有了读研的坚定目标，在努力了4年之后以专业第2的成绩被保送到武汉大学。在担任福州大学校友协会分会的会长期间，我成功邀请2000余名杰出校友参与了福州大学55周年校庆，并且在担任团支书和班长期间，策划、组织过各种团立项和班级活动，并连续2年被评为“优秀团支书”。

在工作之余，我喜欢健身、瑜伽，每天必做的事就是夜跑3公里，曾学习过9年的拉丁舞，是一枚热衷运动的妹子。我是一个充满好奇心并喜欢思考探索的人，喜欢观看像地理发现、宇宙奥秘等节目。希望能加入华为团队和优秀的前辈们一起成长，谢谢您！

● English Personal introduction

Good afternoon, sir! It's very grateful to have this opportunity for an interview. I am LuoXin, born in Fujian province. I am a postgraduate student of Wuhan University. My major is electronics and communication engineering. Data center reliability analysis is my research direction. Recently, my job focus on **the warning system of memory errors in the data center**. I have a well understanding of the Internet knowledge. My graduation thesis which is awarded the excellent graduation design is mainly about the Internet. Besides, I have strong study capability, I ever won the first scholarship three times and postgraduate special scholarship. I mastered the skill of playing magic cube within 5 hours.

In terms of student work, I was the president of alumni association of Fuzhou University and took apart in the preparatory work for the 55th anniversary celebration.

In my spare time, I prefer the gym and yoga. I usually run over 3 kilometers every night. I'm good at Latin dance which I ever practiced for more than 9 years. I really hope I can be a member of *** and do a slightly positive contribution. Thanks!

二、如何准备一篇“万能”的自我介绍并保证不扣分

我的方法是：**3个关键词+支撑案例**（为什么是3个？这是管理咨询通用的习惯，3也是让人记住的最好数字）

（1）第一部分（一句话简单明了）：我是谁，什么学校什么专业，非常高兴参加今天XX公司XX岗位的面试。正如您在简历中所看到的，我希望用3个关键词来介绍自己。

（2）第二部分（3个关键词+支撑案例）：3个关键词不但要**因人而异**，更要**因公司而异**。结合你要面的行业、公司和岗位所需求的技能和特质来给出，最好是**4字词语**。3个词的顺序正是你简历的顺序，**第一个词针对“教育背景”**，这一块你可以说**“快速学习”、“复合专业”、“数理背景”**等等。**第二个词针对“工作/实习经历”**，这一块你可以说**“高效负责”**，也可以结合岗位，比如研究性质强的，你可以说**“潜心研究”**，然后重点列举你之前的工作中的研究成果和经验，会用哪些软件和数据库，以前的学术论文著作等。**第三个词结合课外活动和特长爱好**，你可以说**“爱好交际”**或者**“热爱生活”**，然后说你的学生会、乐队、社团、志愿经历等。切记注意，一定要契合公司的需求，不要捡了芝麻丢了西瓜，比如对方需要研究能力，那就别对志愿工作大谈特谈。（至少在自我介绍里少谈）

（3）第三部分（简明扼要表明谢意和兴趣）：一句话说明自己对XX行业XX类型的工作很感兴趣，同时XX公司是这一行业的杰出代表，非常感谢有机会参加这次面试。

通过上面3个部分，首尾两部分表达对该岗位的认知和兴趣（**主观**），中间部分介绍自己（**客观**），这样结合起来层次丰满，几乎适用于任何行业的面试（包括但不限于求职，比如考研之类的面试也可以套用）。

开放性问题准备

2017年6月28日 18:52

STAR流程

situation：当时背景情况如何？——线上营销获取客户的流程是什么

Target：你的目标是什么？——流程有什么需要完善的地方

action：你采取了什么样的行动？——针对这些问题你是怎么处理的

result：结果如何？——结果如何？是否还有改进空间

最让你失落/后悔/遗憾的事

1. 报考专业没有选择医生。若想在省级大医院从医一定要读博。那个时候害怕要读这么多年的书。家里有个医生对家人来说可以省去很多麻烦。

最让你感到自豪/成功/喜悦的事

1. 拉丁舞，获得华南地区单人舞少年组第4名。小学被评为“校园金花”
2. 义务卖报，筹集善款1600+元，去福利院看望老人20+名。
3. 参与筹备福州大学55周年校庆，2000+名校友。
4. 7个小时学会魔方
5. 拿到夏令营offer

最让你受挫的事

1. 问题：本科的时候一直是专业排名前3，有优越感、成就感，到了武大这么一个人才聚集之处，感觉到和周围同学的差距好大。那些本科是武大的同学通常只需要在考试前突击2、3天就能拿到不错的成绩，我通常需要提前1-2周进行复习。和实验室的师兄师姐们比，自己在就业技能方面也是不够扎实的。

解决方法：做了学习规划（阶段时间、内容）。

9-10月	C/C++，书籍、视频资源、GitHub项目， English技能提升，阅读300+文献，从一开始逐字翻译到现在的轻松阅读
11-1月	前端开发 数据中心可靠性研究 听音乐系的课，古典音乐鉴赏：“维也纳三杰”的三位古典乐派代表人:海顿、莫扎特和贝多芬。

性格里的缺点

答题角度只能从技能，知识两个方面

往高处说：能力层次有高有低，请你挑一个与你目前所在层次相隔较远的能力缺陷来说。

往远处说：术业有专攻，找一个与你本职工作间隔较远的专业能力缺陷来说。

1. 对于文字类的工作的排版、格式等有轻微的强迫症，遵循一定的规则，符合规范要求。（细节要求严苛）

导致的后果：对于别人交给我的材料如果不符合规范，我会很难受，一般会让对方修改，但有时这种建议对方不予采纳，或者会说我这人要求太严格、太死板。

2. 太容易相信别人（太单纯）

性格里的优点

1. 持之以恒的品质

对于属于我的工作、职责范围内的事，我都会尽100%的努力完成。例如：有一次因为项目到了中期检验阶段，我在实验室工作到1点完成了之后才回宿舍的。并且我每天坚持跑步至少3公里，这也能体现出我有坚强的意志力。

2. 对于人性我都是抱着积极乐观的态度，容易信任他人。

对你影响最大的三本书

1. 《从你的全世界路过》——张嘉佳。

这本书书名为从你的全世界路过，他在访谈节目中解释说“其实这是个病句。要么你是我的全世界，要么从你的世界路过。我们是无法从一个人的全世界路过的。大部分的希望都是病句。宁可拿病句做题目，也不要看不见希望。”每个人都应该充满希望，无论是对未来，还是对爱情。

这本书是由很多小故事组成的，故事里，形形色色的主人公到处串场，转身却又不见。那么多篇章，有温暖的，有明亮的，有落单的，有疯狂的，有无聊的，也有莫名其妙的。可你总能在每篇故事里找到那永恒的主题——爱。书中的故事总能戳中我的泪点，在别人的故事里，流自己的眼泪。

2. 《小王子》——安托万·德·圣·埃克苏佩里

“真正重要的东西，用眼睛是看不见的。”

3. 《围城》——钱钟书

我这只是看到了表面的荒谬，没有体会到内里的深刻。

想要问boss的问题

1. 华为有非常全面的培养体系，那么对于刚入职的我在面对这些海量的信息应该如何筛选才能有效快速的成长呢？
2. 您希望我在短期内解决哪些问题？

校园经历详述

2017年9月18日 10:42

【项目管理大赛】

(situation)

正值纪念雷锋精神50周年，学校深入开展雷锋活动，组织开展以志愿者服务为主题的项目管理大赛。

(task)

我们组织策划以春暖大地·爱满中国为主题的项目，希望为同学们提供参与慈善活动的平台和途径，另一方面提高团队成员们的实践和创新能力，树立市场经济的意识。

(action)

1. 项目可行性评估

项目实施的必要性、项目意义（社会意义、经济意义、校园意义）、实施条件（专业支持、人力资源、项目优势）

2. 活动内容划分：二手市场、代售残障人士手工艺品、爱心慰问。

3. 活动阶段性目标划分（WBS）

4. 活动区域划分：二手市场区、残疾人商品区、外界商家区、创业团队区、游戏区

5. 人员分组分工：宣传组、外联组、活动组、后勤组。

6. 风险评估（SWOT分析）：

内部优势S：

项目可操作性强

项目受益人群广，影响范围大

团队层举办过类似活动

媒体资源优势

内部劣势W：

已获取的赞助金额有限

外部机会O：

校方鼓励开展志愿服务类活动；

能够争取到商家的外联赞助

外部威胁T：

同一时间段，校园内举办的活动众多，同学们的注意力肯能被转移；

外出安全风险

风险识别及应对：

资金风险：双方协议，法律约束

管理风险：前期摸底、中期考核

环境风险：预案

技术风险：阶段性监测、口碑营销

存在的问题：

- a. 传单数量估计有误，再合理利用资源方面不足
- b. 收集的二手物品登记环节有些紊乱
- c. 每栋宿舍收集的物品数量差异较大，人员分配不太合理