WEB前端：

1. 常用哪几种浏览器测试？有哪些内核（layout Engine）？

浏览器：IE、Chrome、Safari、Opera

内核：Trident、Gecko、Presto、Webkit

1. 行内元素和会计元素的区别？行内块元素的兼容性使用

行内元素：会在水平方向排列，不能包含块级元素，设置width无效，height无效（可以设置line-height），margin上下无效，padding上下无效。

块级元素：各占据一行，垂直方向排列。从新行开始结束接着一个断行。

兼容性：display:inline-block；\*display:inline；\*zoom:1；

1. 清除浮动有哪些方式？比较好的方式是哪一种？

* 父级div定义height
* 结尾处加空div标签clear：both
* 父级div定义overflow：hidden；**//经常使用**
* 父级div定义overflow：auto
* 父级div定义伪类：after和zoom
* 父级div也浮动，需要定义宽度
* 父级div定义display：table；
* 结尾处加br标签clear：both；

1. Box-sizing常用的属性有哪些？分别有什么作用？

Box-sizing:content-box|border-box|inherit;

* Content\_box；宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框（元素默认效果）。
* border-box:元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

1. Doctype作用？标准模式与兼容模式各有什么区别?

* 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。

1. HTML5 为什么只需要写 ？

* HTML5不基于 SGML，因此不需要对DTD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）。
* 而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

1. 页面导入样式时，使用link和@import有什么区别？

* 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;
* 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;
* 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。

1. 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)和JS引擎。

* 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。
* 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。

1. html5有哪些新特性？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

Q1

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消失
* 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

Q2

* IE8/IE7/IE6支持通过document.createElement方法产生的标签，
* 可以利用这一特性让这些浏览器支持HTML5新标签，
* 浏览器支持新标签后，还需要添加标签默认的样式。
* 当然也可以直接使用成熟的框架、比如html5shim，

1. 简述下你对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。

1. 介绍js的基本数据类型

Undefined、Null、Boolean、Number、String

1. Js有哪些内置对象：

* 数据封装类对象：Object、Array、Boolean、Number 和 String
* 其他对象：Function、Arguments、Math、Date、RegExp、Error

1. This对象的理解

* 它的功能是把对应的字符串解析成JS代码并运行；
* 应该避免使用eval，不安全，非常耗性能（2次，一次解析成js语句，一次执行）。
* 由JSON字符串转换为JSON对象的时候可以用eval，var obj =eval('('+ str +')')。

1. DOM怎样添加、移除、移动、复制、创建和查找节点

// 创建新节点

　　createDocumentFragment() //创建一个DOM片段

　　createElement() //创建一个具体的元素

　　createTextNode() //创建一个文本节点

　　// 添加、移除、替换、插入

　　appendChild()

　　removeChild()

　　replaceChild()

　　insertBefore() //在已有的子节点前插入一个新的子节点

　　// 查找

　　getElementsByTagName() //通过标签名称

　　getElementsByName() //通过元素的Name属性的值(IE容错能力较强，会得到一个数组，其中包括id等于name值的)

　　getElementById() //通过元素Id，唯一性

1. null和undefined的区别？

null是一个表示"无"的对象，转为数值时为0；undefined是一个表示"无"的原始值，转为数值时为NaN。

　　undefined：

　　（1）变量被声明了，但没有赋值时，就等于undefined。

　　（2) 调用函数时，应该提供的参数没有提供，该参数等于undefined。

　　（3）对象没有赋值的属性，该属性的值为undefined。

　　（4）函数没有返回值时，默认返回undefined。

　　null：

　　（1） 作为函数的参数，表示该函数的参数不是对象。

　　（2） 作为对象原型链的终点。

1. new操作符具体干了什么呢?

（1）创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。

　　（2）属性和方法被加入到 this 引用的对象中。

　　（3）新创建的对象由 this 所引用，并且最后隐式的返回 this 。

1. JSON 的了解？

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于JavaScript的一个子集。数据格式简单, 易于读写, 占用带宽小。

　　格式：采用键值对，例如：{'age':'12', 'name':'back'}

1. call() 和 apply() 的区别和作用？

apply()函数有两个参数：第一个参数是上下文，第二个参数是参数组成的数组。如果上下文是null，则使用全局对象代替。

　　如：function.apply(this,[1,2,3]);

　　call()的第一个参数是上下文，后续是实例传入的参数序列。

　　如：function.call(this,1,2,3);