

1、常用那几种浏览器测试？有哪些内核(Layout Engine)?

答：

(Q1) 浏览器：IE，Chrome，FireFox，Safari，Opera。

(Q2) 内核：Trident，Gecko，Presto，Webkit。

2、 说下行内元素和块级元素的区别？行内块元素的兼容性使用？（ IE8 以下）

答：

(Q1) 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 lineheight)，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。

(Q2) 兼容性：display:inlineblock;*display:inline;*zoom:1;

3、 清除浮动有哪些方式？比较好的方式是哪一种？

答：

(Q1)

(1) 父级 div 定义 height。

(2) 结尾处加空 div 标签 clear:both。

(3) 父级 div 定义伪类:after 和 zoom。

(4) 父级 div 定义 overflow:hidden。

(5) 父级 div 定义 overflow:auto。

(6) 父级 div 也浮动，需要定义宽度。

(7) 父级 div 定义 display:table。

(8) 结尾处加 br 标签 clear:both。

(Q2) 比较好的是第 3 种方式，好多网站都这么用。

4、 box-sizing 常用的属性有哪些？分别有什么作用？

答：

(Q1) box-sizing: content-box|border-box|inherit;

(Q2) 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。

5、 Doctype 作用？标准模式与兼容模式各有什么区别？

答：

(Q1) 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。

(Q2) 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。

6、 HTML5 为什么只需要写 ？

答：

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。

7、 页面导入样式时，使用 link 和@import 有什么区别？

答：

（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。

8、介绍一下你对浏览器内核的理解？

答：

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。

9、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？

答：

(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。

(1)绘画 canvas;

(2)用于媒介回放的 video 和 audio 元素;

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

(4)sessionStorage 的数据在浏览器关闭后自动删除;

(5)语意化更好的内容元素，比如 article、footer、header、nav、section;

(6)表单控件，calendar、date、time、email、url、search;

(7)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，

10、简述一下你对 HTML 语义化的理解？

答：

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。

JavaScript 部分

1、介绍 js 的基本数据类型

答： Undefined、Null、Boolean、Number、String

2、js 有哪些内置对象？

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

3、this 对象的理解

答：this 总是指向函数的直接调用者（而非间接调用者）；

如果有 new 关键字，this 指向 new 出来的那个对象；

在事件中，this 指向触发这个事件的对象，特殊的是，IE 中的 attachEvent 中的 this 总是指向全局对象 Window。

4、eval 是做什么的？

答：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 + ')')。

5、DOM 怎样添加、移除、移动、复制、创建和查找节点

答：

```
// 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
```

6、null 和 undefined 的区别？

答：

 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。

7、new 操作符具体干了什么呢？

答：

- （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 。

8、JSON 的了解？

答：

 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'}

9、call() 和 apply() 的区别和作用？

答：

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);

10、如何获取 UA?

答：

```
function whatBrowser() {  
    document.Browser.Name.value=navigator.appName;  
    document.Browser.Version.value=navigator.appVersion;  
    document.Browser.Code.value=navigator.appCodeName;  
    document.Browser.Agent.value=navigator.userAgent;  
}
```

其他

1、HTTP 状态码知道哪些?

答：

100 Continue 继续，一般在发送 post 请求时，已发送了 http header 之后服务端将返回此信息，表示确认，之后发送具体参数信息

200 OK 正常返回信息

201 Created 请求成功并且服务器创建了新的资源

202 Accepted 服务器已接受请求，但尚未处理

301 Moved Permanently 请求的网页已永久移动到新位置。

302 Found 临时性重定向。

303 See Other 临时性重定向，且总是使用 GET 请求新的 URI。

304 Not Modified 自从上次请求后，请求的网页未修改过。

400 Bad Request 服务器无法理解请求的格式，客户端不应当尝试再次使用相同的内容发起请求。

401 Unauthorized 请求未授权。

403 Forbidden 禁止访问。

404 Not Found 找不到如何与 URI 相匹配的资源。

500 Internal Server Error 最常见的服务器端错误。

503 Service Unavailable 服务器端暂时无法处理请求（可能是过载或维护）。

2、你有哪些性能优化的方法?

答：

（1） 减少 http 请求次数：CSS Sprites, JS、CSS 源码压缩、图片大小控制合适；网页 Gzip，CDN 托管，data 缓存，图片服务器。

（2） 前端模板 JS+数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数

（3） 用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能。

（4） 当需要设置的样式很多时设置 className 而不是直接操作 style。

- (5) 少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。
- (6) 避免使用 CSS Expression (css 表达式) 又称 Dynamic properties(动态属性)。
- (7) 图片预加载, 将样式表放在顶部, 将脚本放在底部 加上时间戳。

3、 什么叫优雅降级和渐进增强?

答:

优雅降级: **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。

渐进增强: 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, 向页面增加无害于基础浏览器的额外样式和功能的。当浏览器支持时, 它们会自动地呈现出来并发挥作用。

4、 哪些常见操作会造成内存泄漏?

答:

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象, 并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为 **0** (没有其他对象引用过该对象), 或对该对象的惟一引用是循环的, 那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话, 会引发内存泄漏。

闭包、控制台日志、循环 (在两个对象彼此引用且彼此保留时, 就会产生一个循环)。

5、 线程与进程的区别

答:

一个程序至少有一个进程, 一个进程至少有一个线程。

线程的划分尺度小于进程, 使得多线程程序的并发性高。

另外, 进程在执行过程中拥有独立的内存单元, 而多个线程共享内存, 从而极大地提高了程序的运行效率。

线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行, 必须依存在应用程序中, 由应用程序提供多个线程执行控制。

从逻辑角度来看, 多线程的意义在于一个应用程序中, 有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用, 来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别。