1、浏览器页面有哪三层构成，分别是什么，作用是什么?

**参考答案**

构成：结构层、表示层、行为层

分别是：HTML、CSS、JavaScript

作用：HTML实现页面结构，CSS完成页面的表现与风格，JavaScript实现一些客户端的功能与业务。

2、HTML5的优点与缺点？

**参考答案**

优点： a、网络标准统一、HTML5本身是由W3C推荐出来的。

          b、多设备、跨平台

          c、即时更新。

          d、提高可用性和改进用户的友好体验；

          e、有几个新的标签，这将有助于开发人员定义重要的内容；

          f、可以给站点带来更多的多媒体元素(视频和音频)；

          g、可以很好的替代Flash和Silverlight；

          h、涉及到网站的抓取和索引的时候，对于SEO很友好；

          i、被大量应用于移动应用程序和游戏。

缺点： a、安全：像之前Firefox4的web socket和透明代理的实现存在严重的安全问题，同时web storage、web socket 这样的功能很容易被黑客利用，来盗取用户的信息和资料。

          b、完善性：许多特性各浏览器的支持程度也不一样。

          c、技术门槛：HTML5简化开发者工作的同时代表了有许多新的属性和API需要开发者学习，像web worker、web socket、web storage 等新特性，后台甚至浏览器原理的知识，机遇的同时也是巨大的挑战

          d、性能：某些平台上的引擎问题导致HTML5性能低下。

          e、浏览器兼容性：最大缺点，IE9以下浏览器几乎全军覆没。

3、Doctype作用? 严格模式与混杂模式如何区分？它们有何意义?

**参考答案**

回答1： （1）、 声明位于文档中的最前面，处于标签之前。告知浏览器的解析器，用什么文档类型 规范来解析这个文档。

            （2）、严格模式的排版和JS 运作模式是以该浏览器支持的最高标准运行。

            （3）、在混杂模式中，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。               （4）、DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

回答2： doctype声明指出阅读程序应该用什么规则集来解释文档中的标记。在Web文档的情况下，“阅读程序”通常是浏览器或者校验器这样的一个程序，“规则”则是W3C所发布的一个文档类型定义（DTD）中包含的规则。

 (1) 声明位于文档中的最前面的位置，处于标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的HTML 文档。

(2)所谓的标准模式是指，浏览器按 W3C 标准解析执行代码；怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式。 严格模式是浏览器根据web标准去解析页面，是一种要求严格的DTD，不允许使用任何表现层的语法，如

。严格模式的排版和JS 运作模式是以该浏览器支持的最高标准运行混杂模式则是一种向后兼容的解析方法，说的透明点就是可以实现IE5.5以下版本浏览器的渲染模式。

(3)浏览器解析时到底使用标准模式还是怪异模式，与你网页中的 DTD 声明直接相关， DTD 声明定义了标准文档的类型（标准模式解析）文档类型，会使浏览器使用相应的方式加载网页并显示，忽略 DTD 声明 ,将使网页进入怪异模式。

4、HTML5有哪些新特性、移除了哪些元素？

**参考答案**

Html5新增了 27 个元素，废弃了 16 个元素，根据现有的标准规范，把 HTML5 的元素按优先级定义为结构性属性、级块性元素、行内语义性元素和交互性元素 4 大类。

结构性元素主要负责web上下文结构的定义

section：在 web 页面应用中，该元素也可以用于区域的章节描述。

header：页面主体上的头部， header 元素往往在一对 body 元素中。

footer：页面的底部（页脚），通常会标出网站的相关信息。

nav：专门用于菜单导航、链接导航的元素，是 navigator 的缩写。

article：用于表现一篇文章的主体内容，一般为文字集中显示的区域。

级块性元素主要完成web页面区域的划分，确保内容的有效分割。

aside：用于表达注记、贴士、侧栏、摘要、插入的引用等作为补充主体的内容。

figure：是对多个元素进行组合并展示的元素，通常与 ficaption 联合使用。

code：表示一段代码块。

dialog：用于表达人与人之间的对话，该元素包含 dt 和 dd 这两个组合元素， dt 用于表示说话者，而 dd 用来表示说话内容。

行内语义性元素主要完成web页面具体内容的引用和描述，是丰富内容展示的基础。

meter：表示特定范围内的数值，可用于工资、数量、百分比等。

time：表示时间值。

progress：用来表示进度条，可通过对其 max 、 min 、 step 等属性进行控制，完成对进度的表示和监事。

video：视频元素，用于支持和实现视频文件的直接播放，支持缓冲预载和多种视频媒体格式。

audio：音频元素，用于支持和实现音频文件的直接播放，支持缓冲预载和多种音频媒体格式。

交互性元素主要用于功能性的内容表达，会有一定的内容和数据的关联，是各种事件的基础。

details：用来表示一段具体的内容，但是内容默认可能不显示，通过某种手段（如单击）与 legend 交互才会显示出来。

datagrid：用来控制客户端数据与显示，可以由动态脚本及时更新。

menu：主要用于交互菜单（曾被废弃又被重新启用的元素）。

command：用来处理命令按钮。

5、你做的网页在哪些浏览器测试过,这些浏览器的内核分别是什么?

**参考答案**

a、 IE: trident 内核

b、 Firefox ： gecko 内核

c、 Safari:webkit 内核

d、 Opera: 以前是 presto 内核， Opera 现已改用 Google Chrome 的 Blink 内核

e、 Chrome:Blink( 基于 webkit ， Google 与 Opera Software 共同开发 )

6、每个HTML文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？

**参考答案**

声明位于文档中的最前面的位置，处于标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。（重点：告诉浏览器按照何种规范解析页面）

7、说说你对HTML5认识?（是什么,为什么）

**参考答案**

**是什么：**

**HTML5指的是包括 HTML 、 CSS 和 JavaScript 在内的一套技术组合。它希望能够减少网页浏览器对于需要插件的丰富性网络应用服务（ Plug-in-Based Rich Internet Application ， RIA ），例如： AdobeFlash 、 Microsoft Silverlight 与 Oracle JavaFX 的需求，并且提供更多能有效加强网络应用的标准集。 HTML5 是 HTML 最新版本， 2014 年 10 月由万维网联盟（ W3C ）完成标准制定。目标是替换 1999 年所制定的 HTML 4.01和 XHTML 1.0 标准，以期能在互联网应用迅速发展的时候，使网络标准达到匹配当代的网络需求。**

**为什么：**

**HTML4陈旧不能满足日益发展的互联网需要，特别是移动互联网。为了增强浏览器功能 Flash 被广泛使用，但安全与稳定堪忧，不适合在移动端使用（耗电、触摸、不开放）。**

**HTML5增强了浏览器的原生功能，符合 HTML5 规范的浏览器功能将更加强大，减少了 Web 应用对插件的依赖，让用户体验更好，让开发更加方便，另外 W3C 从推出 HTML4.0 到 5.0 之间共经历了 17 年， HTML 的变化很小，这并不符合一个好产品的演进规则。**

8、对WEB标准以及W3C的理解与认识?

**参考答案**

**标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外 链css和 js 脚本、结构行为表现的分离、**

**文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，**

**容易维 护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性。**

9、HTML5行内元素有哪些,块级元素有哪些, 空元素有哪些?

**参考答案**

**(1)行内元素**

**a - 锚点**

**\* abbr - 缩写**

**\* acronym - 首字**

**\* b - 粗体 ( 不推荐 )**

**\* bdo - bidi override**

**\* big - 大字体**

**\* br - 换行**

**\* cite - 引用**

**\* code - 计算机代码 ( 在引用源码的时候需要 )**

**\* dfn - 定义字段**

**\* em - 强调**

**\* font - 字体设定 ( 不推荐 )**

**\* i - 斜体**

**\* img - 图片**

**\* input - 输入框**

**\* kbd - 定义键盘文本**

**\* label - 表格标签**

**\* q - 短引用**

**\* s - 中划线 ( 不推荐 )**

**\* samp - 定义范例计算机代码**

**\* select - 项目选择**

**\* small - 小字体文本**

**\* span - 常用内联容器，定义文本内区块**

**\* strike - 中划线**

**\* strong - 粗体强调**

**\* sub - 下标**

**\* sup - 上标**

**\* textarea - 多行文本输入框**

**\* tt - 电传文本**

**\* u - 下划线**

**\* var - 定义变量**

**(2)块元素 (block element)**

**\* address - 地址**

**\* blockquote - 块引用**

**\* center - 居中对齐块**

**\* dir - 目录列表**

**\* div - 常用块级容易，也是 css layout 的主要标签**

**\* dl - 定义列表**

**\* fieldset - form控制组**

**\* form - 交互表单**

**\* h1 - 大标题**

**\* h2 - 副标题**

**\* h3 - 3级标题**

**\* h4 - 4级标题**

**\* h5 - 5级标题**

**\* h6 - 6级标题**

**\* hr - 水平分隔线**

**\* isindex - input prompt**

**\* menu - 菜单列表**

**\* noframes - frames可选内容，（对于不支持 frame 的浏览器显示此区块内容**

**\* noscript - ）可选脚本内容（对于不支持 script 的浏览器显示此内容）**

**\* ol - 排序表单**

**\* p - 段落**

**\* pre - 格式化文本**

**\* table - 表格**

**\* ul - 非排序列表**

**可变元素**

**可变元素为根据上下文语境决定该元素为块元素或者内联元素。**

**\* applet - java applet**

**\* button - 按钮**

**\* del - 删除文本**

**\* iframe - inline frame**

**\* ins - 插入的文本**

**\* map - 图片区块 (map)**

**\* object - object对象**

**\* script - 客户端脚本**

**(3)空元素 ( 在 HTML[1] 元素中，没有内容的 HTML 元素被称为空元素 )**

**<br/> //换行**

**<hr> //分隔线**

**<input> //文本框等**

**<img> //图片**

**<link> <meta>**

10、什么是WebGL,它有什么优点?

**参考答案**

**WebGL（全写 Web Graphics Library ）是一种 3D 绘图标准，这种绘图技术标准允许把 JavaScript 和 OpenGL ES 2.0 结合在一起，通过增加OpenGL ES 2.0 的一个 JavaScript 绑定， WebGL 可以为 HTML5 Canvas 提供硬件 3D 加速渲染，这样 Web 开发人员就可以借助系统显卡来在浏览器里更流畅地展示 3D 场景和模型了，还能创建复杂的导航和数据视觉化。显然， WebGL 技术标准免去了开发网页专用渲染插件的麻烦，可被用于创建具有复杂 3D 结构的网站页面，甚至可以用来设计 3D 网页游戏等等。**

**WebGL完美地解决了现有的 Web 交互式三维动画的两个问题：**

**第一，它通过HTML脚本本身实现 Web 交互式三维动画的制作，无需任何浏览器插件支持 ;**

**第二，它利用底层的图形硬件加速功能进行的图形渲染，是通过统一的、标准的、跨平台的OpenGL接口实现的。**

**通俗说WebGL中 canvas 绘图中的 3D 版本。因为原生的 WebGL 很复杂，我们经常会使用一些三方的库，如 three.js 等，这些库多数用于 HTML5游戏开发。**

11、请你描述一下 cookies，sessionStorage 和 localStorage 的区别?

**参考答案**

**sessionStorage 和 localStorage 是 HTML5 Web Storage API 提供的，可以方便的在 web 请求之间保存数据。有了本地数据，就可以避免数据在浏览器和服务器间不必要地来回传递。**

**sessionStorage、 localStorage 、 cookie 都是在浏览器端存储的数据，其中 sessionStorage 的概念很特别，引入了一个“浏览器窗口”的概念。sessionStorage 是在同源的同窗口（或 tab ）中，始终存在的数据。也就是说只要这个浏览器窗口没有关闭，即使刷新页面或进入同源另一页面，数据仍然存在。关闭窗口后， sessionStorage 即被销毁。同时“独立”打开的不同窗口，即使是同一页面， sessionStorage 对象也是不同的**

**cookies会发送到服务器端。其余两个不会。**

**Microsoft 指出 Internet Explorer 8 增加 cookie 限制为每个域名 50 个，但 IE7 似乎也允许每个域名 50 个 cookie 。 Firefox 每个域名 cookie 限制为 50 个。 Opera 每个域名 cookie 限制为 30 个。 Firefox 和 Safari 允许 cookie 多达 4097 个字节，包括名（ name ）、值（ value ）和等号。 Opera 许 cookie 多达 4096 个字节，包括：名（ name ）、值（ value ）和等号。 Internet Explorer 允许 cookie 多达 4095 个字节，包括：名（ name ）、值（ value ）和等号。**

**区别：**

**- Cookie**

**+ 每个域名存储量比较小（各浏览器不同，大致 4K ）**

**+ 所有域名的存储量有限制（各浏览器不同，大致 4K ）**

**+ 有个数限制（各浏览器不同）**

**+ 会随请求发送到服务器**

**- LocalStorage**

**+ 永久存储**

**+ 单个域名存储量比较大（推荐 5MB ，各浏览器不同）**

**+ 总体数量无限制**

**- SessionStorage**

**+ 只在 Session 内有效**

**+ 存储量更大（推荐没有限制，但是实际上各浏览器也不同）**