Doctype 作用？标准模式与兼容模式各有什么区别?

<!DOCTYPE>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。 当 DOCTYPE 不存在或格式不正确会导致文档以兼容模式呈现。

标准模式与兼容模式的区别：

兼容模式（Quirks mode）：在 W3C 标准出台之前，不同的浏览器在页面的渲染上没有同一的规范，产生了差异。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。

标准模式（strict mode）：W3C 标准出台之后，不同浏览器对页面的渲染有了统一的标准。

```
在严格模式中 ：width 是内容宽度 ，元素真正的宽度 = margin-left + border-left-width + padding-left + width + padding-right + border-right- width +  margin-right;
在兼容模式中 ：width 则是元素的实际宽度 ，内容宽度 = width - ( padding-left + padding-right + border-left-width + border-right-width
```

HTML5 为什么只需要写 <!DOCTYPE HTML>?

HTML5 不基于 SGML(Standarded General Markup Language)，因此不需要对 DTD（Document Type Define）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。

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，每个元素都有默认的 display 值，如 div 的 display 默认值为“block”，则为“块级”元素；span 默认 display 属性值为“inline”，是“行内”元素。

（1）行内元素有：a b span img input select strong

（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> <img> <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？

使用链接 link 和导入 import 的好处就是易于维护，但当网速比较慢的时候，会出现加载中断的情况，导致页面排版错误。

（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;

（4）@import 可以在 css 中再次引入其他样式表，比如可以创建一个主样式表，在主样式表中再引入其他的样式表，如：

```
main.css
————————-
@import “sub1.css”;
@import “sub2.css”;

sub1.css
————————-
p {color:red};

sub2.css
————————-
.myclass {color:blue}
```

这样更利于修改和扩展，这样做有一个缺点，会对网站服务器产生过多的 HTTP 请求，以前是一个文件，而现在却是两个或更多文件了，服务器的压力增大，浏览量大的网站还是谨慎使用。

（5）当使用 JavaScript 控制 DOM 去改变样式的时候，只能使用 link 标签，因为@import 不是 DOM 可以控制的

```
<head>
    <style type="text/css">@import
url(http://www.dreamdu.com/style.css);</style>
</head>
```

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：解析和执行 javascript 来实现网页的动态效果。浏览器的快慢大部分说的是 JS 的渲染速度。

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。

4 款常用的浏览器内核（Rendering Engine）：

Trident→IE

Gecko→FireFox

Presto→Opera

Webkit→Chrome,Safari