**一、URL的概念**

统一资源定位符（或称统一资源定位器/定位地址、URL地址等，英语：Uniform Resource Locator，常缩写为URL），有时也被俗称为**网页地址（网址）。**如同在网络上的门牌，是因特网上标准的资源的地址（Address）。

**二、URL的格式**

**2.1 标准格式**

协议类型:[//服务器地址[:端口号]][/资源层级UNIX文件路径]文件名?查询

**2.2 完整格式**

协议类型:[//[访问资源需要的凭证信息@]服务器地址[:端口号]][/资源层级UNIX文件路径]文件名?查询

其中【访问凭证信息@；:端口号；?查询；#片段ID】都属于选填项。

**三、URL的语法规则**

比如网址 http://segmentfault.com/html/index.asp，必须遵守以下的语法规则：

scheme://host.domain:port/path/filename

**3.1 说明**

（1）**scheme** - 定义因特网服务的类型。最常见的类型是 http

（2）**host** - 定义域主机（http 的默认主机是 www）

（3）**domain** - 定义因特网域名，比如 w3school.com.cn

（4）**:port** - 定义主机上的端口号（http 的默认端口号是 80）

（5）**path** - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）。

（6）**filename** - 定义文档/资源的名称

**3.2 URL Schemes**

以下是其中一些最流行的 scheme：

| **Scheme** | **访问** | **用于...** |
| --- | --- | --- |
| http | 超文本传输协议 | 以 http:// 开头的普通网页。不加密。 |
| https | 安全超文本传输协议 | 安全网页。加密所有信息交换。 |
| ftp | 文件传输协议 | 用于将文件下载或上传至网站。 |
| file |  | 您计算机上的文件。 |

**四、URL的类型**

**4.1 绝对URL**

绝对URL（absolute URL）显示文件的完整路径，这意味着绝对URL本身所在的位置与被引用的实际文件的位置**无关**。

**4.2 相对URL**

相对URL（relative URL）以包含URL本身的文件夹的位置为**参考点**，描述目标文件夹的位置。

一般来说，对于**同一服务器**上的文件，应该总是使用**相对URL**，它们更容易输入，而且在将页面从本地系统转移到服务器上时更方便，只要每个文件的相对位置保持不变，链接就仍然是有效地。

以下为建立路径所使用的几个特殊符号，及其所代表的意义。

（1） .：代表**目前所在的目录**，相对路径。 如： <a>文本 </a> 或 <img src="./abc" />

（2） ..：代表**上一层**目录，相对路径。 如： <a>文本 </a> 或 <img src="../abc" />

（3） ../../：代表的是**上一层目录的上一层**目录，相对路径。 如： <img src="../../abc" />

（4） /：代表**根目录**，绝对路径。 如：[文本] (/abc) 或 <img src="/abc" />

**五、href的概念**

**5.1 规范解释**

href (Hypertext Reference)指定网络资源的位置，从而在当前元素或者当前文档和由当前属性定义的需要的锚点或资源之间定义一个链接或者关系。

**5.2 通俗理解**

href 目的不是为了引用资源，而是为了建立联系，让当前标签能够链接到目标地址。

**六、src的概念**

source（缩写），指向外部资源的位置，指向的内容将会应用到文档中当前标签所在位置。

**七、href和src的区别**

**7.1 请求资源类型不同**

（1）href 指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的联系。

（2）在请求 src 资源时会将其指向的资源下载并应用到文档中，比如 JavaScript 脚本，img 图片；

**7.2 作用结果不同**

（1）href 用于在当前文档和引用资源之间确立联系；

（2）src 用于替换当前内容；

**7.3 浏览器解析方式不同**

（1）若在文档中添加 ，浏览器会识别该文档为 CSS 文件，就会**并行下载资源并且不会停止对当前文档的处理。**这也是为什么建议使用 link 方式加载 CSS，而不是使用 @import 方式。

（2）当浏览器解析到 ，会**暂停其他资源的下载和处理，**直到将该资源加载、编译、执行完毕，图片和框架等也如此，类似于将所指向资源应用到当前内容。这也是为什么建议把 js 脚本放在底部而不是头部的原因。

**八、link和@import的区别**

两者都是外部引用 CSS 的方式，但是存在一定的区别：

（1）link是XHTML标签，除了能够加载CSS，还可以定义RSS等其他事务；而@import属于CSS范畴，只可以加载CSS。

（2）link引用CSS时，在页面载入时同时加载；@import需要页面完全载入以后再加载。

（3）link是XHTML标签，无兼容问题；@import则是在CSS2.1提出的，低版本的浏览器不支持。

（4）link支持使用Javascript控制DOM改变样式；而@import不支持。