

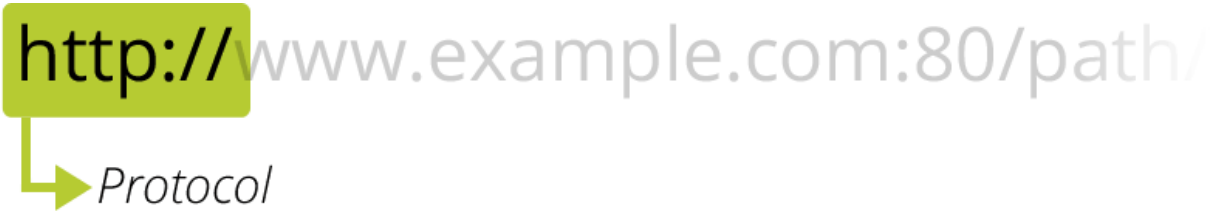
# 资源和 URI

## 1. 标识互联网上的内容

每个资源都由一个 (URI, 统一资源标识符) 来进行标识。资源的名称和位置由同一个 URL (统一资源定位符, 它是 URI 的一种) 来标识。

### 1.1 统一资源标识符的语法

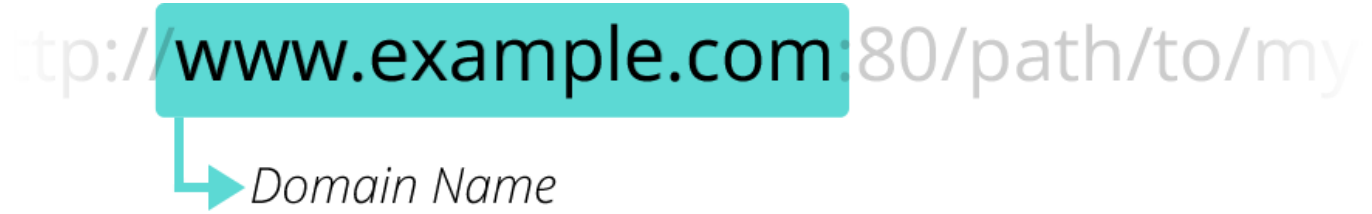
#### (1) 方案或协议



常见的方案或协议有：

方案	描述
data	前缀为 <code>data:</code> 协议的 URL, 唯一的不透明来源
file	本地文件
ftp	文件传输协议, 用于互联网上的文件传输
http/https	超文本传输协议/安全的超文本传输协议
mailto	电子邮件地址
ssh	安全 shell
tel	电话
urn	统一资源名称, 用于命名资源
view-source	资源的源代码
ws/wss	WebSocket 连接/加密的 WebSocket 连接

#### (2) 主机



`www.example.com` 既是一个域名, 也代表管理该域名的机构。它指示了需要向网络上的哪一台主机发起请求。

#### (3) 端口



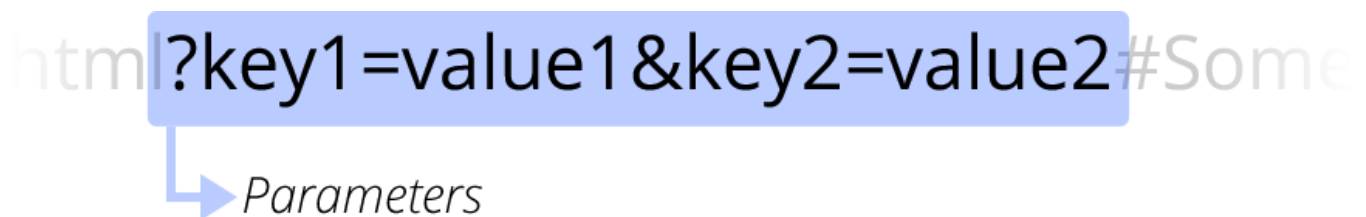
如果访问的该 Web 服务器使用 HTTP 协议的标准端口（HTTP 为 80，HTTPS 为 443）授予对其资源的访问权限，则通常省略此部分。否则端口就是 URI 必须的部分。

#### (4) 路径



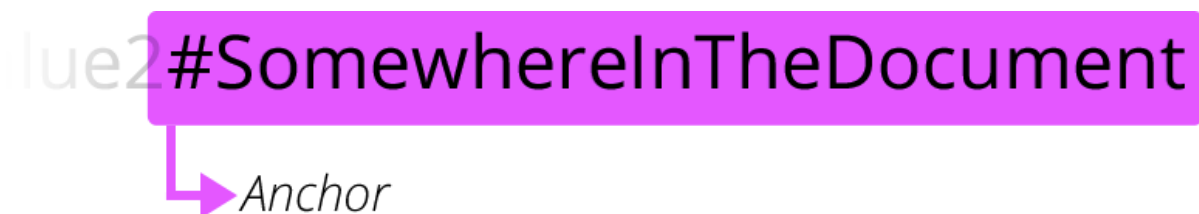
`/path/to/myfile.html` 是 Web 服务器上资源的路径。

#### (5) 查询



`?key1=value1&key2=value2` 是提供给 Web 服务器的额外参数。这些参数是用 `&` 符号分隔的键/值对列表。

#### (6) 片段



`#SomewhereInTheDocument` 是资源本身的某一部分的一个锚点。锚点代表资源内的一种“书签”，它给予浏览器显示位于该“加书签”点的内容的指示。例如，在 HTML 文档上，浏览器将滚动到定义锚点的那个点上；在视频或音频文档上，浏览器将转到锚点代表的那个时间。**# 号后面的部分，也称为片段标识符，永远不会与请求一起发送到服务器。**

## 2. Data URL

Data URL，即前缀为 `data:` 协议的 URL。

Data URL 由四个部分组成：前缀（`data:`）、指示数据类型的 MIME 类型、如果非文本则为可选的 base64 标记、数据本身：

```
data:[<mediatype>][;base64],<data>
```

`mediatype` 是个 MIME 类型的字符串，例如 `'image/jpeg'` 表示 JPEG 图像文件。如果被省略，则默认值为 `text/plain; charset=US-ASCII`。

一些示例：

```
data:,Hello%2C%20World!
```

简单的 `text/plain` 类型数据。逗号编码为：`%2C`，空格字符编码为：`%20`。

```
data:text/html,%3Ch1%3EHello%2C%20World!%3C%2Fh1%3E
```

一个 HTML 文档源代码 `<h1>Hello, World</h1>`

```
data:text/html,%3Cscript%3Ealert%28%27hi%27%29%3B%3C%2Fscript%3E
```

带有 `<script>alert('hi');</script>` 的 HTML 文档，用于执行 JavaScript 警告。

## 2.1 给数据作 base64 编码

Base64 是一组二进制到文本的编码方案。

### (1) 在 JavaScript 在编码

在 JavaScript 中，有两个函数被分别用来处理解码和编码 Base64 字符串：

- `btoa()`：从二进制数据“字符串”创建一个 Base-64 编码的 ASCII 字符串（“btoa”应读作“binary to ASCII”）
- `atob()`：解码通过 Base-64 编码的字符串数据（“atob”应读作“ASCII to binary”）

```
function utf8_to_b64(str) {  
    return window.btoa(unescape(encodeURIComponent(str)));  
}  
  
function b64_to_utf8(str) {  
    return decodeURIComponent(escape(window.atob(str)));  
}  
  
// Usage:  
utf8_to_b64("✓ à la mode"); // "4pyTIMOgIGxhIG1vZGU="   
b64_to_utf8("4pyTIMOgIGxhIG1vZGU="); // "✓ à la mode"
```

### 3. MIME 类型

**媒体类型**（也通常称为多用途互联网邮件扩展或 MIME 类型）是一种标准，用来表示文档、文件或一组数据的性质和格式。

#### 3.1. MIME 类型结构

MIME 类型由两部分组成：**主类型** 和 **子类型**：`type/subtype`。每种类型都有自己的一组可能的子类型。一个 **MIME 类型总是包含类型与子类型这两部分，且二者必需成对出现。**

#### 3.2. 常见 MIME 类型

##### (1) 文本类型

类型	描述
text/plain	普通文本
text/html	HTML 文档
text/css	CSS 样式表
text/javascript	JavaScript 脚本
text/xml	XML 文档
text/markdown	Markdown 文档
text/event-stream	服务器发送事件流

##### (2) 图片类型

类型	描述
image/avif	AV1 图像文件格式
image/gif	GIF 图像
image/jpeg	JPEG 图像
image/png	PNG 图像
image/svg+xml	SVG 图像
image/webp	WebP 图像

##### (3) 音频/视频类型

类型	描述
<code>audio/wave</code> 、 <code>audio/wav</code> 、 <code>audio/x-wav</code> 、 <code>audio/x-pn-wav</code>	采用 WAVE 容器的音频文件。
audio/ogg	Ogg Vorbis 音频
audio/mpeg	MP3 音频

类型	描述
audio/wav	WAV 音频
audio/webm	WebM 音频
video/mp4	MP4 视频
video/ogg	Ogg Theora 视频
video/webm	WebM 视频
video/3gpp	3GPP 视频

(4) multipart/form-data

multipart/form-data 可用于 HTML 表单从浏览器发送信息给服务器。

(5) multipart/byteranges

multipart/byteranges 用于把部分的响应报文发送回浏览器。

当状态码 206 Partial Content 被发出后，这个 MIME 类型用于指出这个文件由若干部分组成，每一个都有其请求范围。

(7) application/octet-stream

这是二进制文件的默认值。由于这意味着未知的二进制文件，浏览器一般不会自动执行或询问执行。浏览器将这些文件视为 Content-Disposition 标头被设置为 attachment 一样，并弹出“另存为”对话框。

4. 选择 www 或非 www 域名

4.1. 使用 HTTP 301 重定向

尝试使访问非规范网址的浏览器重定向到其规范的等效网址。

4.2. 使用 <link rel="canonical">

将一个特殊的 HTML <link> 元素添加到网页，就能指明网页的规范地址。它对页面的普通读者没有影响，但会告诉搜索引擎爬虫这个页面实际在哪。当加入这个标签，在两个域名中提供相同的内容，会告诉搜索引擎哪个域名是规范的。

```
<link href="http://example.org/whaddup" rel="canonical" />
```