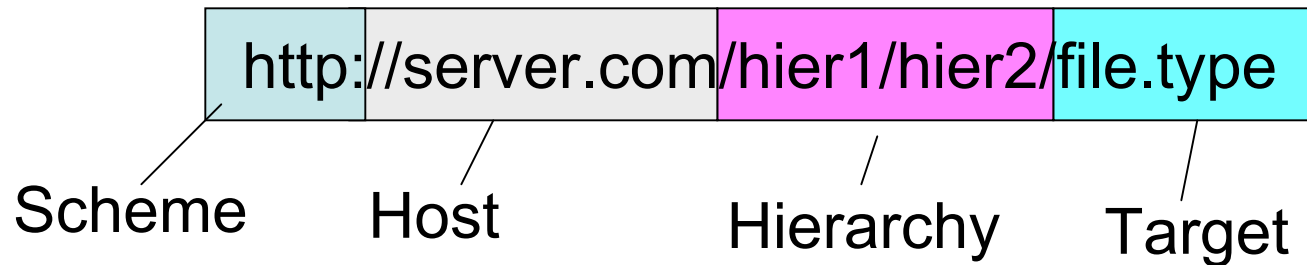


# URI



PATH = Hierarchy+Target

*There are different terms used in different books and documents -- these are the ones we will use in this class (and on exams)*

Note: Remember, URL is a URI from the HTTP scheme (our definition, used differently sometimes)

# URIs (more detail)

- <scheme>://<user>:<password>@<host>:port/...
  - These are less used (port is more common)
- ...<path>;<params>?<query>#<frag>
  - These are used often
    - ? => start query (& and = used for query string)
    - # => fragment ID (client, not server)
- Special characters
  - % (escape token for special chars)
  - #,? As above
  - ; parameter delimitator (not always used)
  - :,\$,+ reserved
  - @,&= used in some URI schemes, used in queries

# HTTP

- Request message
  - Method: GET, POST, PUT, HEAD, etc.
  - URL path
  - Version: HTTP/<major>.>minor>
  - URL host

```
GET /path1/path2/foo.nam?moose=q&fish=y HTTP/1.1  
Host: www.example.com
```

# HTTP cont

- Response Message
  - Status code (200, 302, 404, etc.) 3-digit
  - Reason phrase (human readable)
  - Headers [name: value <CRLF>]\*
  - Entity Body - CRLF or the message

```
HTTP/1.0 200 OK
Content-type: text/plain
Content-length: 19
<crLf>
<b>Hi, I'm the body</b>
```

# Methods

- GET no body
- HEAD no body
- POST body
- PUT body
- TRACE no body
- OPTIONS no body
- DELETE no body

# Status codes

- Informational            100-199    100-101
- Successful                200-299    200-206
- Redirection               300-399    300-305
- Client error              400-499    400-415
- Server error              500-599    500-505

# General headers

- Connection
- Date
- MIME-version
- Via (if message has gone through proxies of gateways)
- Others less used
  - Trailer, transfer-encoding, upgrade

# Request headers

- Client-IP
- From
- Host
- Referer (URI of doc with this request)
- UA (implemented in most clients, not spec)
  - UA-color, UA-CPU, UA-Disp, UA-OS
- User agent
  - The name of the application making the request



# Accept headers

- Accept: what MIME types are ok
- Accept- headers
  - Charset, Encoding, Language

MIME= Multipurpose Internet Mail Extensions

# Condition Req. headers

- Expect (what server behaviors are needed)
- If-Match (same "entity tag" -uri+version)
- If-None-Match (diff entity tags)
- If-Modified-Since
- If-Unmodified-Since

# Req. Sec. headers

- Authorization (password, key, etc.)
- Cookie (pass a token)
- Cookie2 (cookie version info)

# Response Headers

- Age
- Retry-after
- Server
- Title
- Warning (more detail than phrase)
- Public (*deprecated* - req. method list)

# Content Headers

- Content-Base (base URL)
- Content-Encoding
- Content-Language
- Content-Location
- Content-MD5
- Content-Range (byte range)
- Content-Type