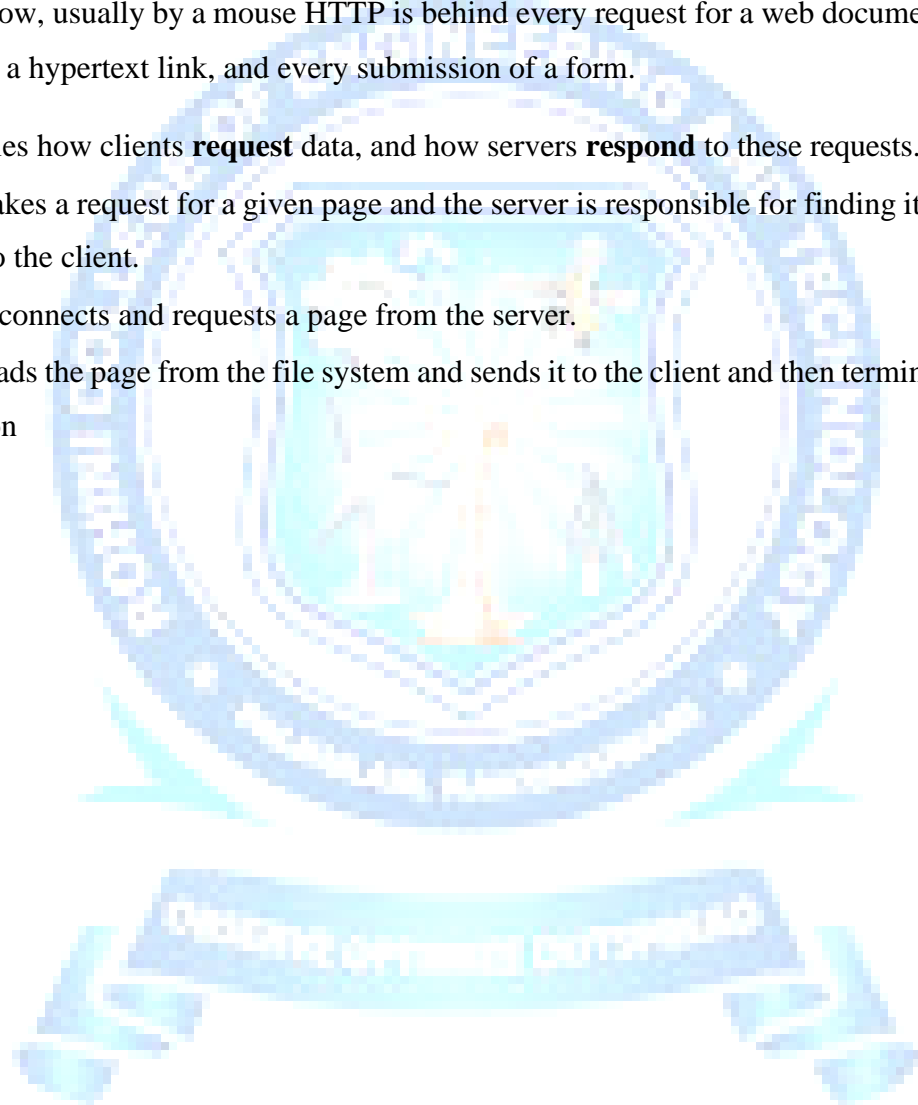**HTTP**

Hypertext Transfer Protocol (HTTP) is the communication protocol used by the Internet to transfer hypertext documents.

A protocol to transfer hypertext requests and information between servers and browsers
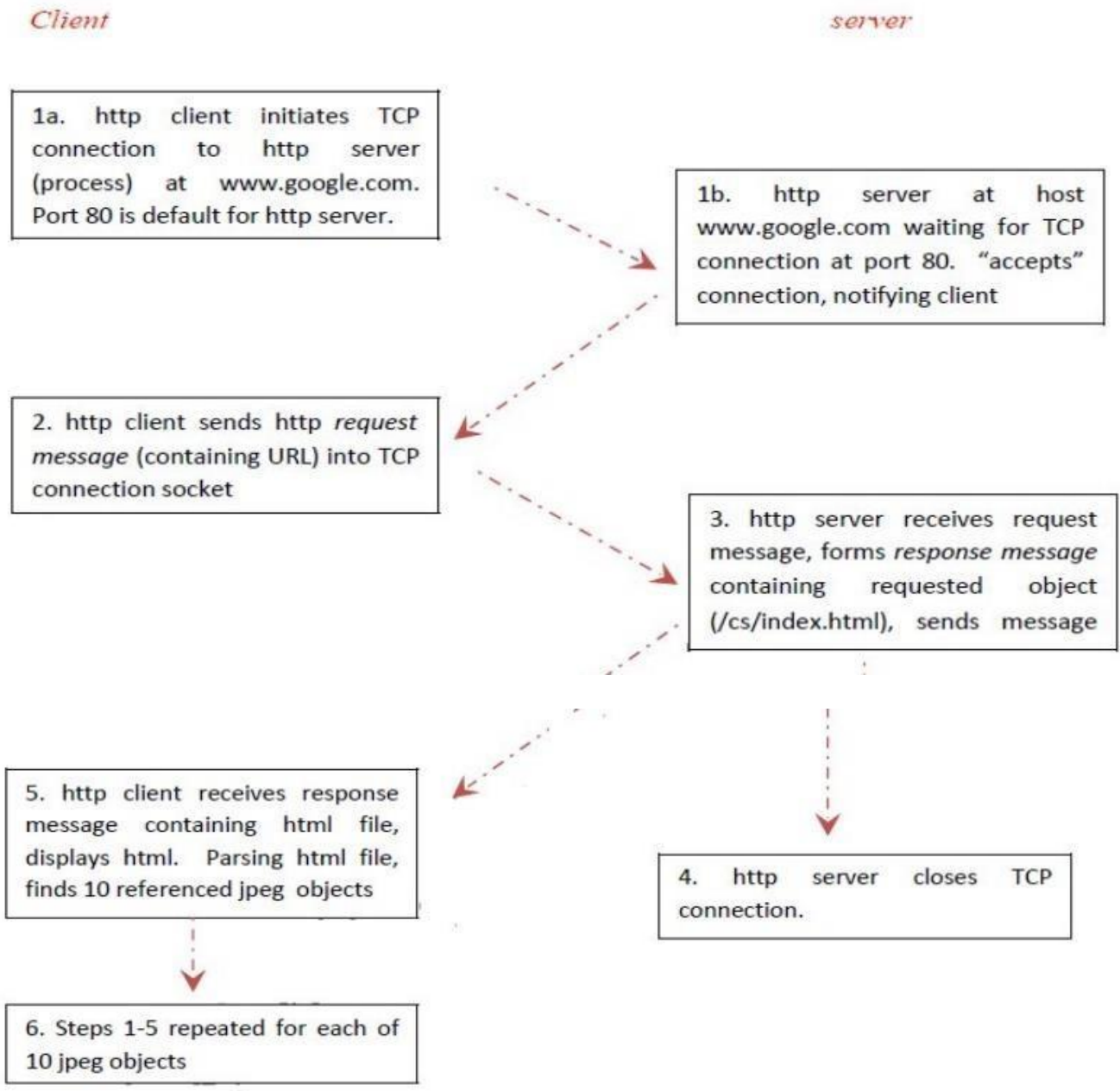
Hypertext is text, displayed on a computer, with references (hyperlinks) to other text that the reader can immediately follow, usually by a mouse HTTP is behind every request for a web document or graph, every click of a hypertext link, and every submission of a form.

- HTTP specifies how clients **request** data, and how servers **respond** to these requests.
- The client makes a request for a given page and the server is responsible for finding it and returning it to the client.
- The browser connects and requests a page from the server.
- The server reads the page from the file system and sends it to the client and then terminates the connection

HTTP Transactions

HTTP Transactions

Client   server

1a. http client initiates TCP connection to http server (process) at www.google.com. Port 80 is default for http server.

1b. http server at host www.google.com waiting for TCP connection at port 80. "accepts" connection, notifying client

2. http client sends http *request message* (containing URL) into TCP connection socket

3. http server receives request message, forms *response message* containing requested object (/cs/index.html), sends message

5. http client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

4. http server closes TCP connection.

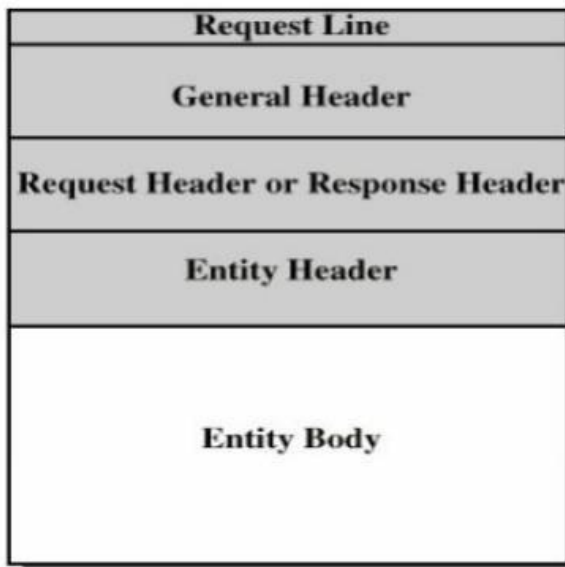6. Steps 1-5 repeated for each of 10 jpeg objects

HTTP Message:

HTTP message is the information transaction between the client and server.

Two types of HTTP Message:

 1. Requests

      a. Client to server

2. Responses

      a. Server to client

Request Line
General Header
Request Header or Response Header
Entity Header
Entity Body

Fields

      · Request line or Response line

      · General header

      · Request header or Response header

      · Entity header

      · Entity body

Request Message:

Request Line:

• A request line has three parts, separated by Spaces

o a *method* name

o the local path of the requested resource o the version of HTTP being used

• A typical request line is:

o GET /path/to/file/index.html HTTP/1.1

o **GET** is the most common HTTP method; it says "give me this resource". Other methods include **POST** and **HEAD.** Method names are always uppercase
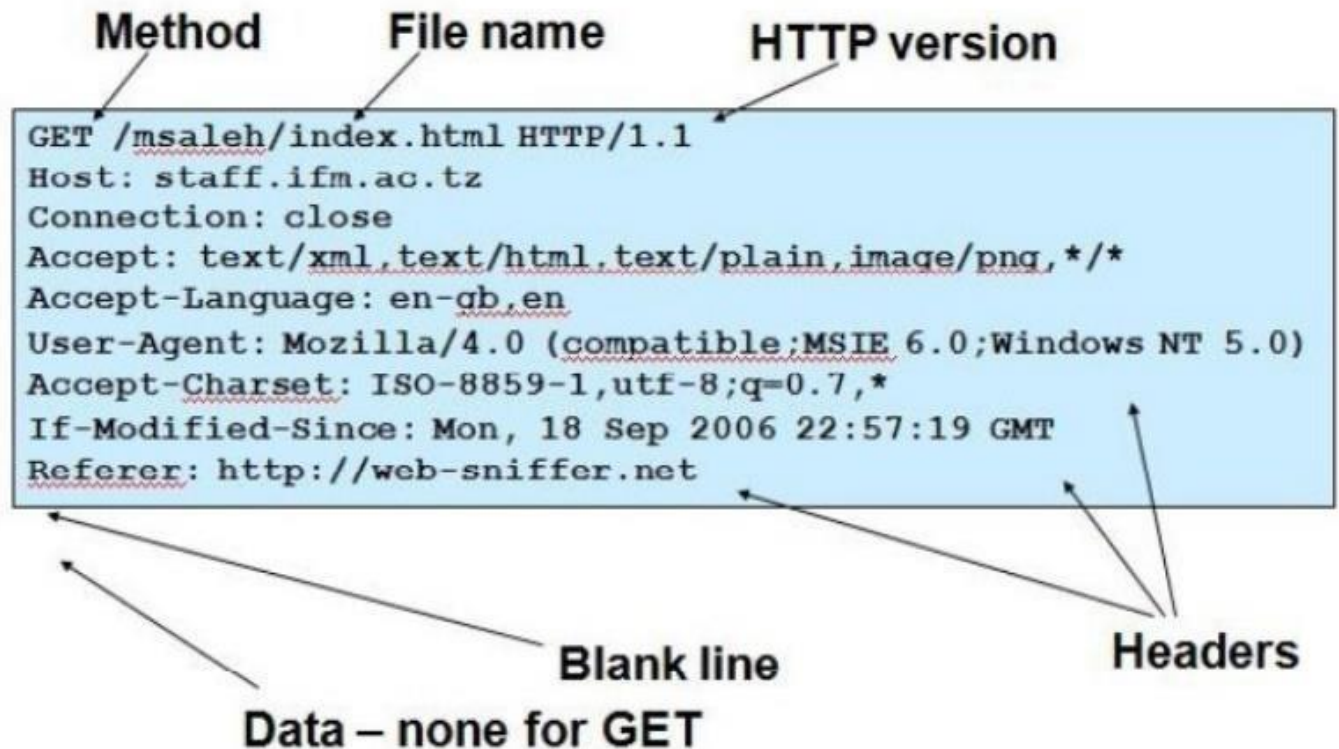
o The path is the part of the URL after the host name, also called the *request URI* o The HTTP version always takes the form "**HTTP/x.x**", uppercase.

Request Header:

HTTP Request Headers

Header	Description
From	Email address of user
User-Agent	Client s/w
Accept File	File types that client will accept
Accept-encoding	Compression methods
Accept-Language	Languages
Referrer	URL of the last document the client displayed
If-Modified-Since	Return document only if modified since specified
Content-length	Length (in bytes) of data to follow

HTTP Request

Method   **File name**   **HTTP version**

```
GET /msaleh/index.html HTTP/1.1
Host: staff.ifm.ac.tz
Connection: close
Accept: text/xml,text/html.text/plain,image/png,*/*
Accept-Language: en-gb,en
User-Agent: Mozilla/4.0 (compatible;MSIE 6.0;Windows NT 5.0)
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*
If-Modified-Since: Mon, 18 Sep 2006 22:57:19 GMT
Referer: http://web-sniffer.net
```

Blank line   **Headers**

Data – none for GET

Response Message:

Response Line:

- A request line has three parts, separated by spaces

    o the HTTP version,

    o a *response status code* that gives the result of the request, and

    o an English *reason phrase* describing the status code

- Typical status lines are:

    o HTTP/1.0 200 OK or

    o HTTP/1.0 404 Not Found

o The HTTP version is in the same format as in the request line, "**HTTP/x.x**".

o The status code is meant to be computer-readable; the reason phrase is meant to be human-readable, and may vary.

HTTP Request Header:

HTTP Response Headers

Header	Description
Server	Server software
Date	Current Date
Last-Modified	Modification date of document
Expires	Date at which document expires
Location	The location of the document in redirection responses
Pragma	A hint, e.g., no cache
MIME-version	
Link	URL of document's parent
Content-Length	Length in bytes
Allowed	Requests that user can issue, e.g., GET
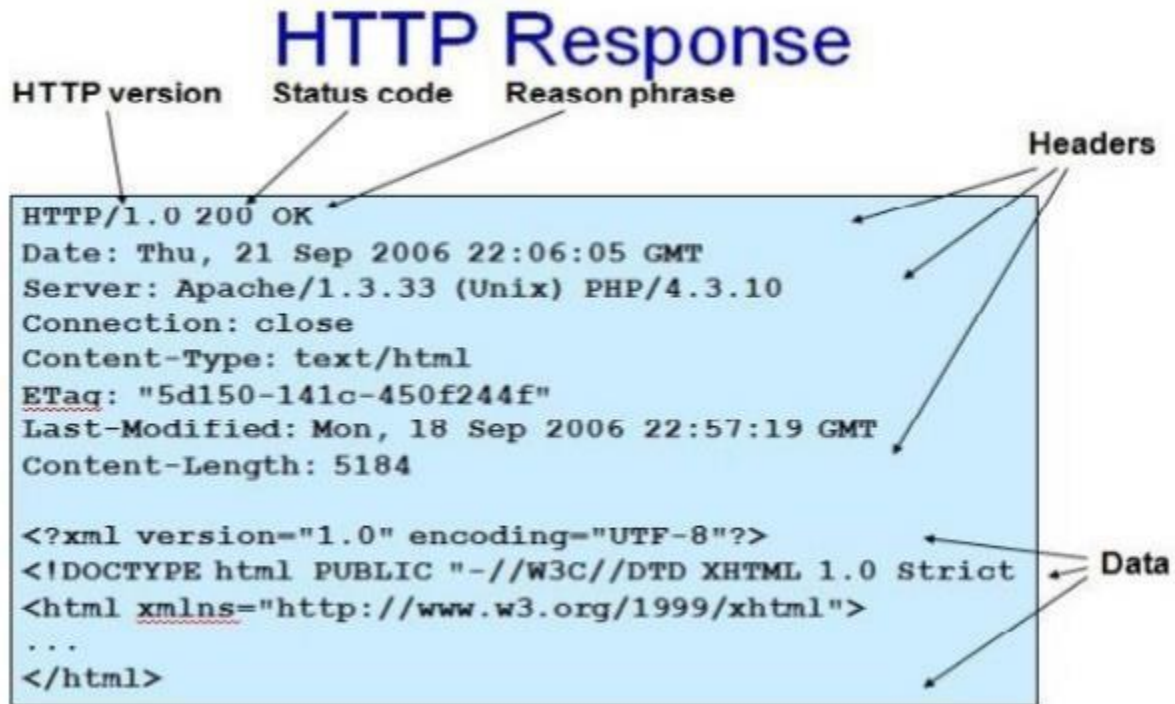
HTTP Method:

     • HTTP method is supplied in the request line and specifies the operation that the client has requested.

Some common methods:

     • Options

     • Get

     • Head

     • Post

     • Put

     • Move

     • Delete

Two methods that are mostly used are the GET and POST:

o **GET** for queries that can be safely repeated

o **POST** for operations that may have side effects (e.g. ordering a book from an on-line store).

The GET Method

• It is used to retrieve information from a specified URI and is assumed to be a safe, repeatable operation by browsers, caches and other HTTP aware components

• Operations have no side effects and GET requests can be re-issued.

• For example, displaying the balance of a bank account has no effect on the account and can be safely repeated.

• Most browsers will allow a user to refresh a page that resulted from a **GET**, without displaying any kind of warning

• Proxies may automatically retry **GET** requests if they encounter a temporary network connection problem.

• GET requests is that they can only supply data in the form of parameters encoded in the URI (known as a **Query String**) – [downside]

Cannot be unused for uploading files or other operations that require large amounts of data to be sent to the server.

The POST Method

• Used for operations that have side effects and cannot be safely repeated.

• For example, transferring money from one bank account to another has side effects and should not be repeated without explicit approval by the user.

If you try to refresh a page in Internet Explorer that resulted from a **POST**, it displays the following message to warn you that there may be side effects:

The **POST** request message has a content body that is normally used to send parameters and data

• The IIS server returns two status codes in its response for a **POST** request

> o The first is **100 Continue** to indicate that it has successfully received the **POST** request

> o The second is **200 OK** after the request has been processed.

HTTP response status codes

> • Informational (1xx)

> • Successful (2xx)

> • Redirection (3xx)

> > o 301: moved permanently

> • Client error (4xx)

o 403 : forbidden o 404: Not found

> • Server error (5xx)

> > o 503: Service unavailable

> > o 505: HTTP version not supported