This is a basic introduction to HTTP(HyperText Transfer Protocol) and HTML. To understand the basic purpose of each HTTP/1.1 method and HTML is a good start to WebApp Development.
Go ahead!

HTTP && HTML

1.HTTP (HyperText Transfer Protocol)

Http functions as a request-response protocol in client-server computing model. HTTP protocol is a **application layer protocol** and presumes the common used **transport protocol TCP(Transmission Control Protocol)** .
See more

What makes HTTP important is those methods that HTTP/1.1 contains and for Web App development those methods really matter. So in the next content, brief introductions to those methods are my focus.

Since HTTP is a request-response protocol in Web App, first take a deep look into request methods, which obviously catch our attention.
HTTP/1.1 Request methods contain the following:

- GET
- POST
- HEAD
- TRACE
- PUT
- DELETE
- OPTIONS

1.1: Request Methods

1. GET
   GET method asks to get things from requested URL and GET method only retrieve data and should not have other effect.
2. POST
   POST method is like a fat-GET method, it asks the Server to accept the body info attached to the request and give it to the thing at the requested URL.
3. HEAD
   HEAD method is like a thin-GET method, unlike GET method return all the content at the requested URL, it only returns the head/meta info at the requested URL.
4. TRACE
   TRACE method asks a loopback of the request message so that the response will contains all the head info of the request message.
5. PUT
   The PUT method requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI.
6. DELETE
   DELETE method deletes the thing at requested URL.
7. OPTIONS
   OPTIONS method asks a list of HTTP methods that the requested URL can respond.

Safe Methods *vs* Idempotent Methods

- Safe Methods
  Safe methods mean that client only retrieves data from Server and have on other effect. Based on this definition, methods include **GET, HEAD, TRACE, OPTIONS** above can be regarded as safe methods since they only retrieve data from Server.
- Idempotent methods
  Idempotent methods can be defined as multiple operations of the same methods looks like only one operation has been done. Based on this definition, methods in safe methods should be considered as Idempotent methods. And besides that, methods include **PUT and DELETE** can also be regarded as idempotent methods.
  So the only method can be regarded as un-idempotent method is **POST**.

Request *vs* Response

Now take a deep look into the specific details of Client request and Server Response.

- **Client Request**

```
POST/sample.jspHTTP/1.1

Accept:image/gif.image/jpeg,*/*
Accept-Language:zh-cn
Connection:Keep-Alive
Host:localhost
User-Agent:Mozila/4.0(compatible;MSIE5.01;Window NT5.0)
Accept-Encoding:gzip,deflate
 //This blank line is important and indicates the end of head info
username=jinqiao&password=1234
```

  Explanantion:

  - First line(Status Line):
    - Method: GET
    - URI: sample.jsp
    - Protocol and Protocol Version: HTTP/1.1
  - Request Header
    This part contains the request client information and some other useful information, such as used language and body length
  - Body
    Request doesn't have a body

- **GET *VS* POST**
  The difference between these two methods contain three parts:
  - POST has a body while GET don't;
  - GET info can be added after Request line while POST don't;
  - GET method can send limited info while POST can send larger size data;

- **Server Response**
  Just like Client Request, three parts are included:
  - Status Line
    Format: **HTTP-Version Status-Code Reason-Phrase CRLF**
    like: **HTTP/1.1 200 OK \r\n**
    Status Code: 3 digits indicates the Request whether be accepted or processed
    - 1xx: Request has been accepted, to be processed;
    - 2xx: Request has been accepted and processed properly;
    - 3xx: Redirect – Request needs further operation;
    - 4xx: Client Error. Most seen 404(Not Found);
    - 5xx: Server Error
  - Response Header
    Response Header also contains information like:
    - Location
    - Server
    - Content Length
    - Content Language
    - Content-Encoding
    - Content-Type
    - Date
  - Body
    This part includes those information that the Request needs.