GET and POST basically allow information to be sent back to the webserver from a browser (or other HTTP client for that matter).

Imagine that you have a form on a HTML page and clicking the "submit" button sends the data in the form back to the server, as "name=value" pairs.

Choosing GET as the "method" will append all of the data to the URL and it will show up in the URL bar of your browser. The amount of information you can send back using a GET is restricted as URLs can only be 1024 characters.

**http://www.jguru.com/images/pdfImage.jpg**

**Cost-Effective, Scale-Out Backup in One Solution**

[**Download Now**](javascript:%20return%20false;?BLUID=20140106012928990147991)

A POST on the other hand will (typically) send the information through a socket back to the webserver and it won't show up in the URL bar. You can send much more information to the server this way - and it's not restricted to textual data either. It is possible to send files and even binary data such as serialized Java objects!

A PUT allows you to "put" (upload) a resource (file) on to a webserver so that it be found under a specified URI. DELETE allows you to delete a resource (file). These are both additions to HTTP/1.1 and are not usually used. HEAD returns just the HTTP headers for a resource. TRACE and OPTIONS are also HTTP/1.1 additions and also rarely used.

GET method:

1. Data is appended to the URL. - correct
2. Data is publicly available. - correct
3. ~~It is a single call system.~~ - GET/POST/PUT methods both send and receive response/data
4. ~~Maximum data that can be sent is 256.~~ - browser dependant up to 4K
5. ~~Data transmission is faster.~~ - GET/POST/PUT methods all transmit at the same speed
6. This is the default method for many browsers. - For non-form based data only

POST method:

1. ~~Data is not appended to the URL.~~ - POST data can either be appended to the URL or in body
2. ~~Data is secret.~~ - only for https sessions otherwise is public
3. ~~It is a two call system.~~ - GET/POST/PUT methods both send and receive response/data
4. There is no limit on the amount of data. That is, any amount of data can be sent. - correct
5. ~~Data transmission is comparatively slow.~~ - GET/POST/PUT methods all transmit at the same speed
6. ~~No default and should be explicitly specified~~ - it is possible to transmit POST data on a GET method

**Difference between GET and POST methods**

* **Fundamental Difference is probably the Visibility** - GET request is sent via the URL string (appended to the URI with a question-mark as separator), which is visible whereas POST request is encapsulated in the body of the HTTP request and can't be seen.
* **Length** - Since, GET request goes via URL, so it has a limitation for its length. It can't be more than 255 characters long (though this is browser dependent, but usually the max is 255 characters only). Whereas no such maximum length limitation holds for the POST request for the obvious reason that it becomes a part of the body of the HTTP request and there is no size limitation for the body of an HTTP request/response.
* **Performance** - GET request is comparatively faster as it's relatively simpler to create a GET request and the time spent in the encapsulation of the POST request in the HTTP body is saved in this case. In addition, the maximum length restriction facilitates better optimization of GET implementation.
* **Type of Data** - GET request is sent via URL string and as we all know that URL can be text-only, so GET can carry only text data whereas POST has no such restriction and it can carry both text as well as binary data.
* **Caching/Bookmarking** - again for the obvious reason that a GET request is nothing but an URL hence it can be cached as well as Bookmarked. No such luxuries with a POST request.
* **FORM Default** - GET is the default method of the HTML FORM element. To submit a FORM using POST method, we need to specify the method attribute and give it the value "POST".
* **Data Set** - GET requests are restricted to use ASCII characters only whereas POST requests can use the 'enctype' attribute with a value "multipart/form-data" to use the [Universal Multiple-Octet Coded Character Set](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39921) (UCS).