|  |
| --- |
| [[https://myetudes.org/etudes-melete-tool/images/printer.png](https://myetudes.org/portal/tool/4c4d3792-8b10-40ce-8016-d7a5ac569a1c/print_module.jsf?printModuleId=1436385326) Send to Printer](https://myetudes.org/portal/tool/4c4d3792-8b10-40ce-8016-d7a5ac569a1c/print_module.jsf?printModuleId=1436385326) | [Close Window](https://myetudes.org/portal/tool/4c4d3792-8b10-40ce-8016-d7a5ac569a1c/print_module.jsf?printModuleId=1436385326) |
|  | |
| 15. Client Server Architecture  15.1. Overview  *Copyright (c) 2014, Rula Khayrallah*  So far we’ve been working with JavaScript in the browser, loading source documents from our file system.  This week, we’ll cross over to the server side and see how we can use JavaScript there.  We’ll also implement our own web server.  Then we’ll come back to the client side and cover communication with the server from that side.  Let’s first take a look at the distinction between a client and a server and how these two communicate.  **The browsers are all examples of clients:  they request some information from a server and then allow the user to interact with that information.**  **A web server is a program that waits for clients to make a request and then delivers a response. The response may include an HTML document, or any other content such as JavaScript modules, images, style sheets or more.**  15.2. HTTP  *Copyright (c) 2014, Rula Khayrallah*  Clients and servers communicate using the HTTP protocol.  **The HTTP protocol dictates what goes in a client request and in the server response.**  An HTTP request includes:   * The HTTP request method or "verb" such as POST or GET. * The URL being requested.  This is the resource that the client needs. * Optional  headers that may include authentication information. * An optional request body.   An HTTP response includes:   * A  status code such as 200 for 'OK' (success) or  404 for 'Not Found'. * Some headers that allow the server to pass additional information about the response such as  the content type (is it an HTML file?  Is it a CSS file, etc...) * The response body:  this may be a source HTML document or some other resource that was requested.   15.3. TCP/IP Addresses and Port Numbers  *Copyright (c) 2014, Rula Khayrallah*  The HTTP protocol relies on TCP/IP to send and receive sequences of bytes.  Computers connected to a TCP/IP network have an**IP address** such as 74.125.239.102 or 128.242.106.42.  We usually connect to servers using the**Domain Name System** (DNS).  **The DNS  maps IP addresses to domain names** such as google.com or foothill.edu.  **The special name localhost refers to the local computer and corresponds to the special IP address 127.0.0.1.**  **We could have more than one application running on the same computer. Each application must use a different port.** The ports are numbered from 1 to 65535. **In our examples, we’ll use port 8080 for our web server.**  To connect to a server running at a specific port, we append : followed by the port number to the IP address or host name.  For example, we’ll be using [http://localhost:8080](http://localhost:8080/) to connect to our web server.  Note that when we don’t specify a port, the default port 80 is used. |  |