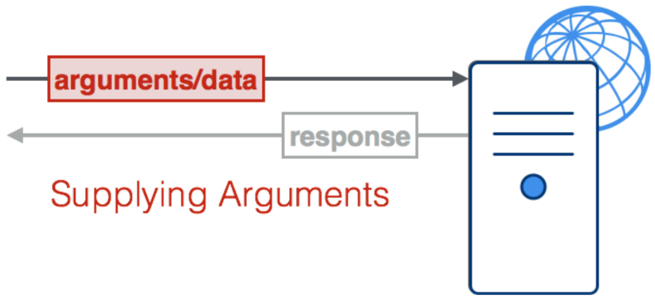
24.4. The HTTP protocol

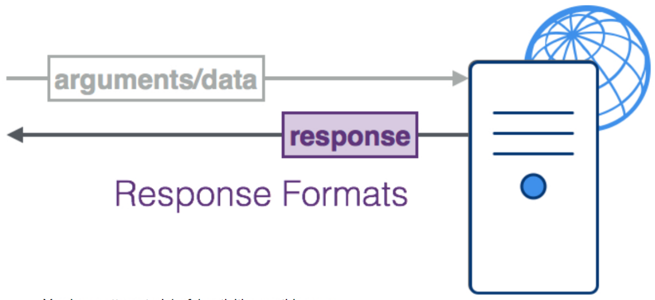
A protocol specifies the order in which parties will speak and the format of what they say and the content of appropriate responses.

HTTP is the protocol that specifies how web browsers or other programs communicate with web servers. One version of the formal specification, before it was later split into multiple documents, was IETF RFC 2616. It is 176 pages long! Fortunately, the basics are pretty easy to understand.

- **Step 1: the client makes a request to the server.**
    - If the request only involves fetching data, the client sends a message of the form `GET <path>`, where <path> is the path part of the URL
    - If the request involves sending some data (e.g., a file upload, or some authentication information), the message starts with `POST`
    - **In either case, the client sends some HTTP headers. These include:**
        - The type of client program. This allows the server to send back different things to small mobile devices than desktop browsers (a "responsive" website)
        - Any cookies that the server previously asked the client to hold onto. This allows the server to continue previous interactions, rather than treating every request as stand-alone. It also allows ad networks to place personalized ads.
    - After the HTTP headers, for a POST type communication, there is some data (the body of the request).



- **Step 2: the server responds to the client.**
    - **The server first sends back some HTTP headers. These include:**
        - a response code indicating whether the server thinks it has fulfilled the request or not.
        - a description of the type of content it is sending back (e.g., text/html when it is sending html-formatted text).
        - any cookies it would like the client to hold onto and send back the next time it communicates with the server.
    - After the headers come the contents. This is the stuff that you would see if you ask to "View Source" in a browser.



You have attempted 1 of 1 activities on this page

24.3. Anatomy of URLs">

24.5. Using REST APIs">

iatomy of URLs">

✔ Completed. Well Done!

24.5. Using REST APIs">Next Section - 24.5. Using REST APIs

| Back to top