



24.3. Anatomy of URLs

A URL is used by a browser or other program to specify what server to connect to and what page to ask for. Like other things that will be interpreted by computer programs, URLs have a very specific formal structure. If you put a colon in the wrong place, the URL won't work correctly. The overall structure of a URL is:

```
<scheme>://<host>[:<port>]/<path>
```

Usually, the *scheme* will be http or https. The s in https stands for “secure”. When you use https, all of the communication between the two devices is encrypted. Any devices that intercepts some of the packets along the way will be unable to decrypt the contents and figure out what the data was.

Other schemes that you will sometimes see include ftp (for file transfer) and mailto (for email addresses).

The *host* will usually be a domain name, like si.umich.edu or github.com or google.com. When the URL specifies a domain name, the first thing the computer program does is look up the domain name to find the 32-bit IP address. For example, right now the IP address for github.com is 192.30.252.130. This could change if, for example, github moved its servers to a different location or contracted with a different Internet provider. Lookups use something called the Domain Name System, or DNS for short. Changes to the mapping from domain names to IP addresses can take a little while to propagate: if github.com announces a new IP address associated with its domain, it might take up to 24 hours for some computers to start translating github.com to the new IP address.

Alternatively, the host can be an IP address directly. This is less common, because IP addresses are harder to remember and because a URL containing a domain name will continue to work even if the remote server keeps its domain name but moves to a different IP address.

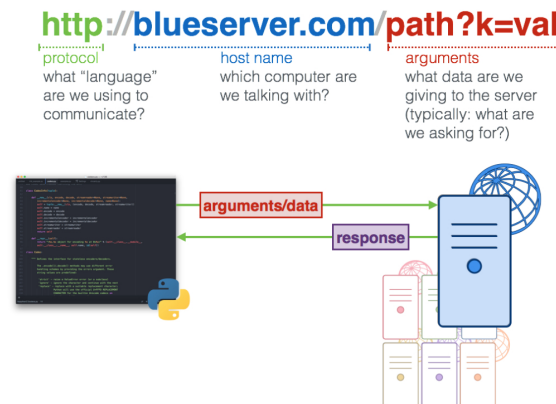
The *:port* is optional. If it is omitted, the default port number is 80. The port number is used on the receiving end to decide which computer program should get the data that has been received. We probably will not encounter any URLs that include the *:* and a port number in this course.

The */path* is also optional. It specifies something about which page, or more generally which contents, are being requested.

For example, consider the url <https://github.com/presnick/runestone>:

- [https://](https://github.com/presnick/runestone) says to use the secure http protocol
- [github.com](https://github.com/presnick/runestone) says to connect to the server at github.com, which currently maps to the IP address 192.30.252.130. The connection will be made on the default port, which is 443 for https.
- [/presnick/runestone](https://github.com/presnick/runestone) says to ask the remote server for the page presnick/runestone. It is up to the remote server to decide how to map that to the contents of a file it has access to, or to some content that it generates on the fly.

The url <http://blueserver.com/path?k=val> is another example that we can consider. The path here a bit different from <https://github.com/presnick/runestone> because it includes what are called “query parameters”, the information after the `?`.



You have attempted 1 of 1 activities on this page

24.2. The Internet: Behind the Scenes">

24.4. The HTTP protocol">

24.2. The Internet: Behind the Scenes">

✓ Completed. Well Done!

24.4. The HTTP protocol">Next Section - 24.4. The HTTP protocol