

CSE 331: Web Development in 1995

The web uses a client-server architecture. Clients send requests and servers respond. Your browser is the client.

When you visit a page like `http://courses.cs.washington.edu/courses/cse331/26su/index.html`, it sends an HTTP GET request to the web server:

```
GET /courses/cse331/26su/index.html HTTP/1.1
Host: courses.cs.washington.edu
```

The server sends back an HTTP response:

```
HTTP/1.1 200 Found
Content-Length: 400
Content-Type: text/html; charset=iso-8859-1
```

```
<html>
...the page contents
</html>
```

Then, your browser renders the HTML for you.

So, a web server is just a program that listens for HTTP requests and sends back responses.

A *server framework* is a library that makes it easy to write your own web servers.

Tangent: servers are often expressed in an event-driven programming style.

```
while (true) {
    wait for next event;
    handle the event;
}
```

Besides serving static pages, servers can do stuff. For example, they can “calculate” the page response instead.

A GET request is supposed to be read-only: the client is asking the server for some existing data.

Clients can also request that the server *do* something, such as *changing* the data, using an HTTP POST request.

```
POST /submit HTTP/1.1
Host: localhost:9331
Content-Length: 16
Content-Type: application/x-www-form-urlencoded
```

```
name=e&message=f
```

Notice that the request itself has a body this time.