HTTP part 2

- Http requests can come with a 3 digit code that dictates the status of the requst sent back to the client. These codes help signal to the recipient how your request was handled. Its a good error handler
- HTTP is a stateless protocol. This means by itself it cannot associate one request with another. There's no memory. Given the same parameters in an HTTP request we can expect the same resulting response.
- We will have tools later to emulate state, but these often require a client that is willing to cooperate. If you are looking to do something like block access to a specific client, it is very difficult to do using only HTTP because even if you give a client tracking headers (like cookies), the client can discard them.

```
const http = require("http");
const https = require("https");
const fs = require("fs");

const Myserver = (req, res) => {
  if (req.url == "/haha") {
    const readStream = fs.createReadStream("index.html");
    res.writeHead(200, { "Content-type": "text/html" });
    readStream.pipe(res);
    res.end();
  }
};

const options = {
  hostname: `${url}`,
  port: 8000,
  path: "/",
  method: "GET",
};

const req = https.request(options, (res) => {
  console.log("statusCode:", res.statusCode);
  console.log("headers:", res.headers);

  res.on("data", (d) => {
    process.stdout.write(d);
  });
});

req.on("error", (e) => {
  console.error(e);
});
req.end();

const server = http.createServer(Myserver);
server.listen(8000, () => {
```

```
    console.log("SERVER IS UP AND RUNNING");
});
```