

# HTTP & HTML: An Overview

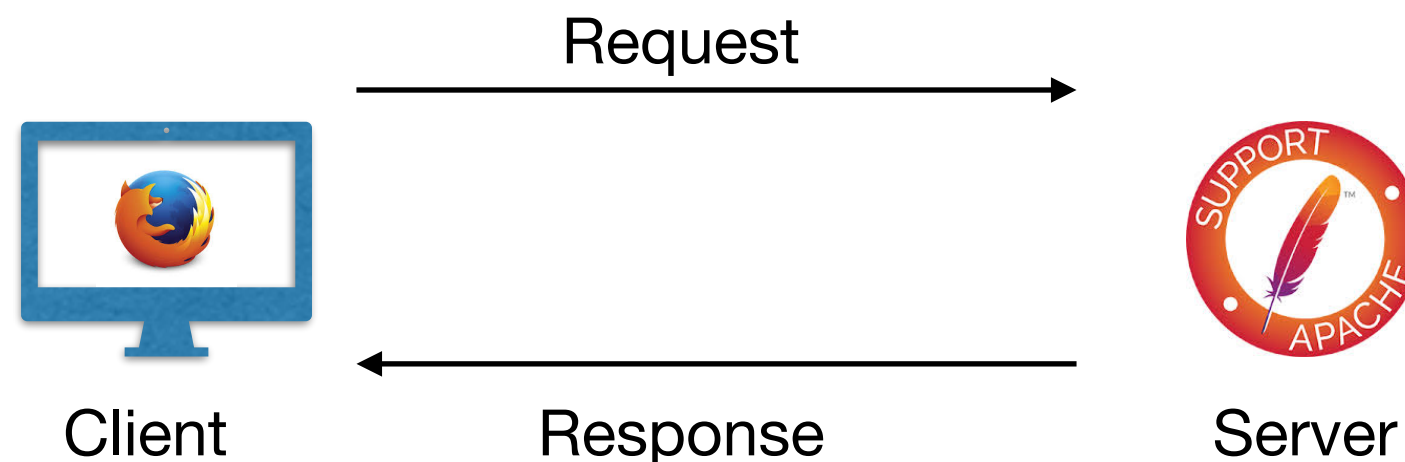
44-440/640-IoT

# Objectives

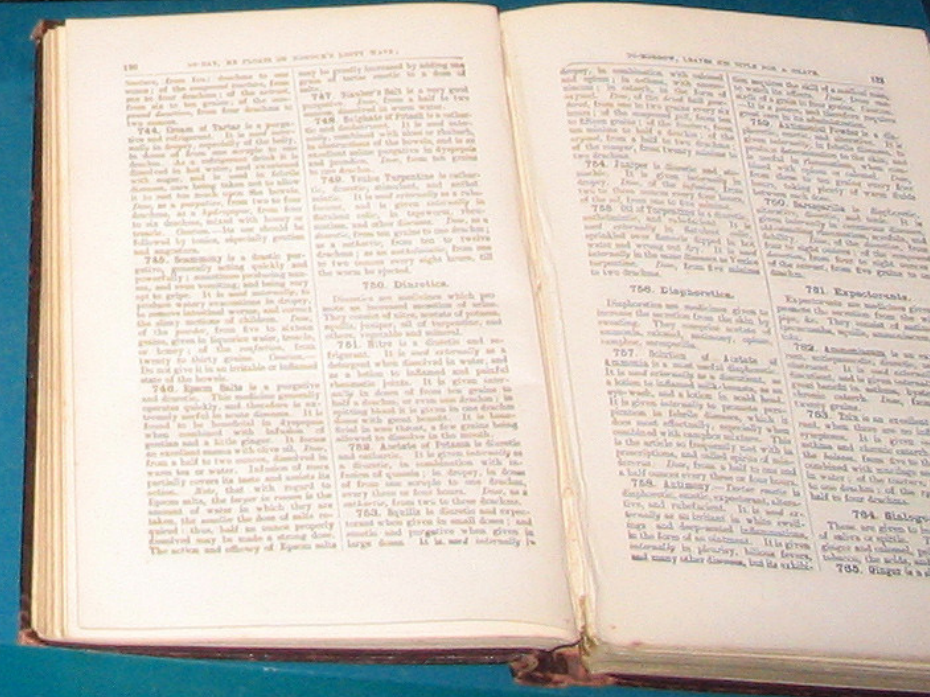
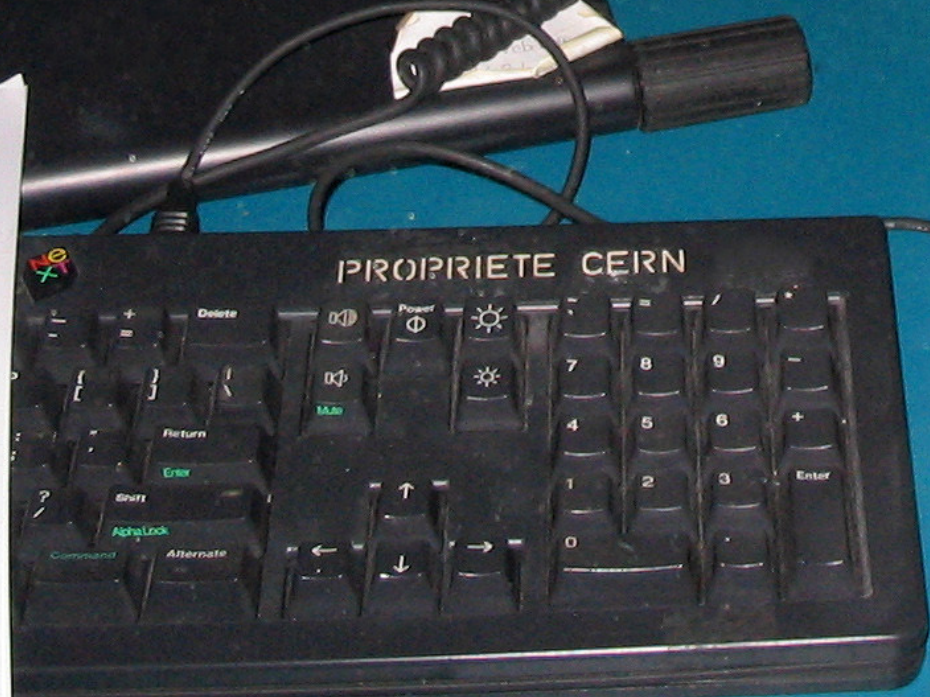
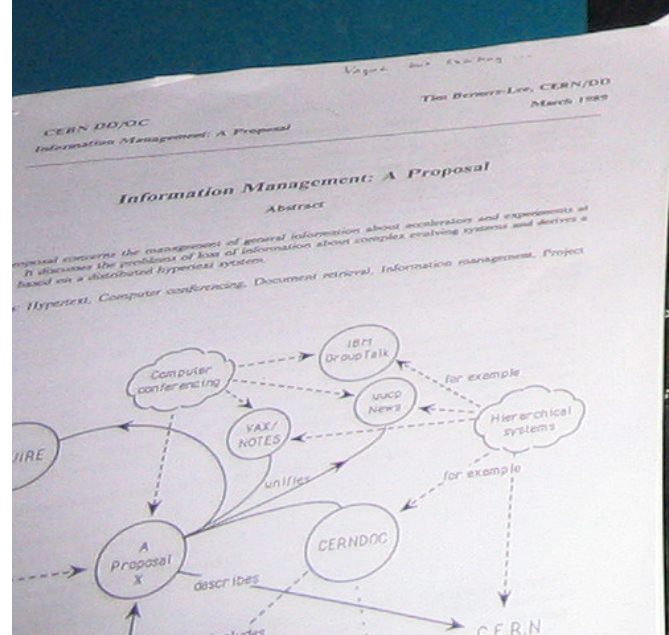
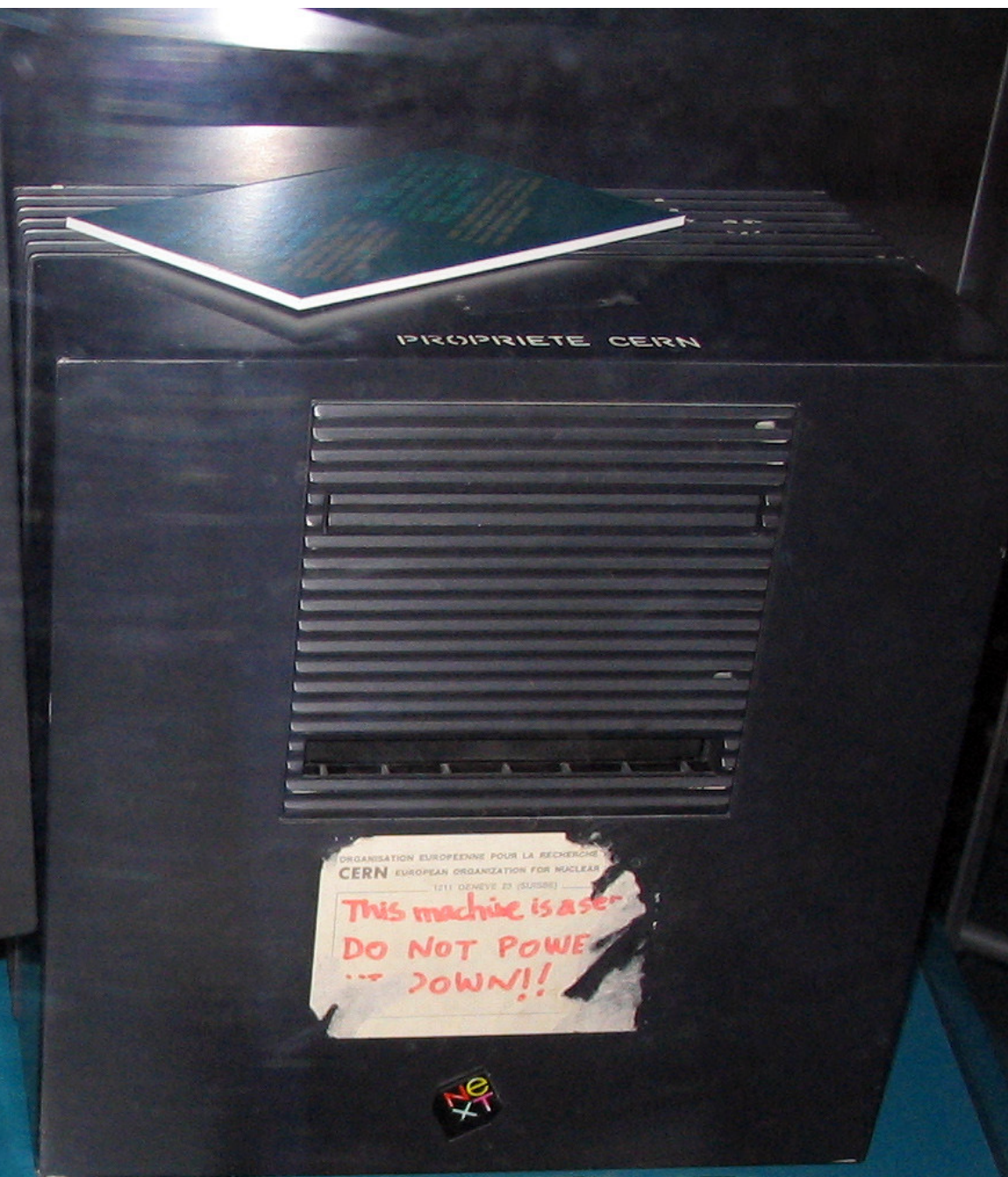
- Students will be able to explain
  - how HTTP works
  - who invented it
  - the structure of an HTTP request and response
  - the structure of GET and POST requests, and the structure of responses to them
  - the purpose of HTML
  - who invented it
  - basic HTML structure and tags

# An HTTP Overview

- The world wide web is powered by **HTTP**, short for Hypertext Transport Protocol, a stateless request/response protocol that exchanges messages between a **client** and a **server**.
- A client sends a **request**; a server sends a **response**
- Traditionally, a client has been a web browser ("user agent"), and a server a full-blown computer running a web server ("host"), but IoT devices can play both roles. HTTP was not built for IoT, predating it by 20 years+. But HTTP (and it's secure relative, **HTTPS**) work well as a vehicle for IoT communication.
- Fun facts: HTTP and HTML were invented by British computer scientist Tim Berners-Lee in 1989, while working at CERN, and his first web page is still up.



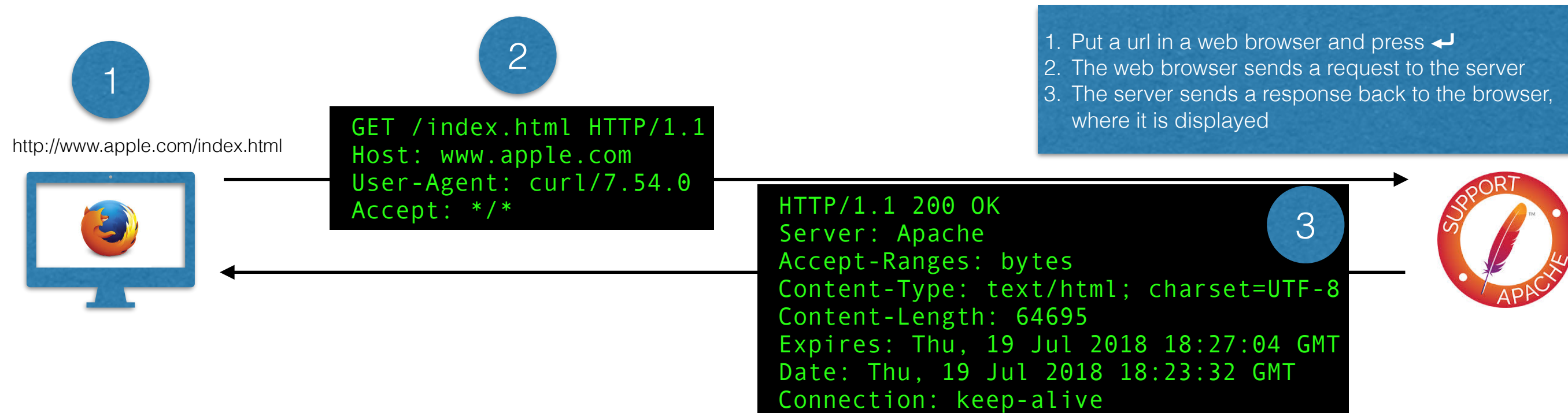






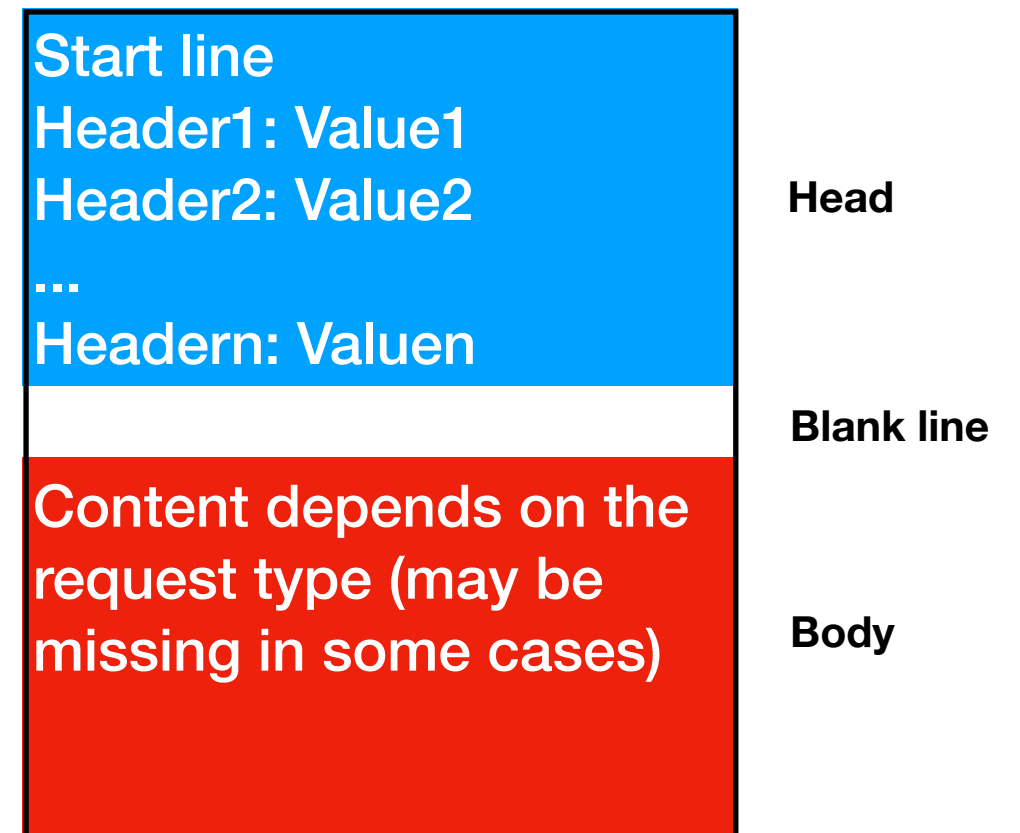
# URLs

- A request involves a **URL**, consisting of a **scheme**, **host name**, **path**, and optionally a **query string**
- e.g., <https://www.apple.com/index.html>, or <https://www.cbc.ca/news/local?type=headlines&lang=en>



# Requests

- The start line begins with a verb (GET, POST, PUT) or a noun (OPTIONS, HEADER) that describes the request type
- Headers provide specific information needed to perform the request. Each occupies 1 line and has this form:
  - header-name: value
- The request head ends with a blank line
- The body content depends on the request type (and may be missing entirely)

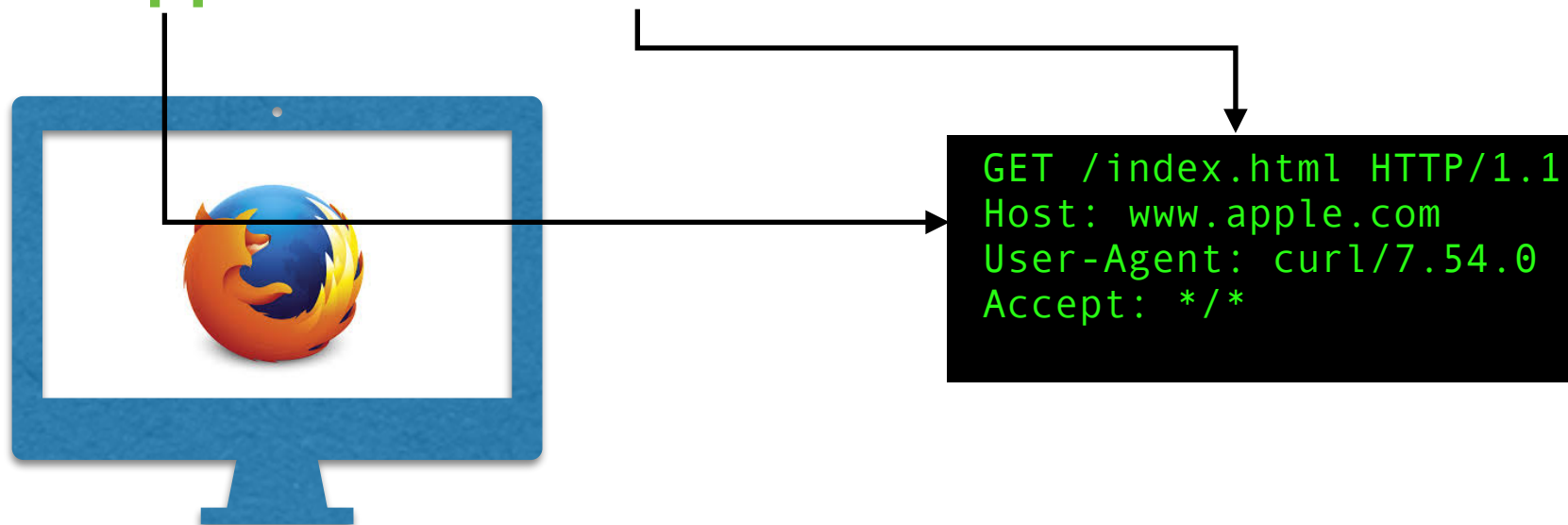


Any relation between this diagram and an inverted flag of the Netherlands is strictly coincidental

# A GET Request

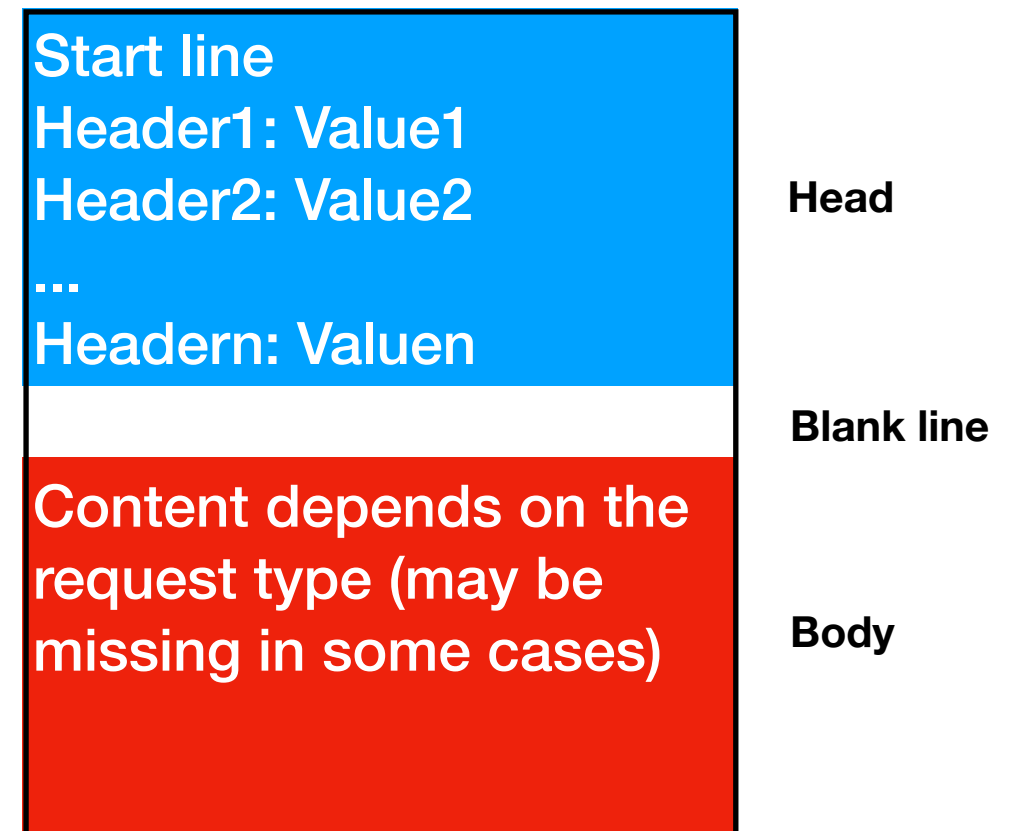
- **GET** requests are used to retrieve web pages. Your browser sends a GET request when you enter a URL in the address field and press ↵
- The start line starts with **GET**, the URL's **path**, and the HTTP version
- The **host** (www.apple.com) is sent in a Host header line
- The browser type is sent in a User-Agent header line
- What the browser is capable of displaying is sent in the Accept header line (in this case, anything)

<http://www.apple.com/index.html>



# Responses

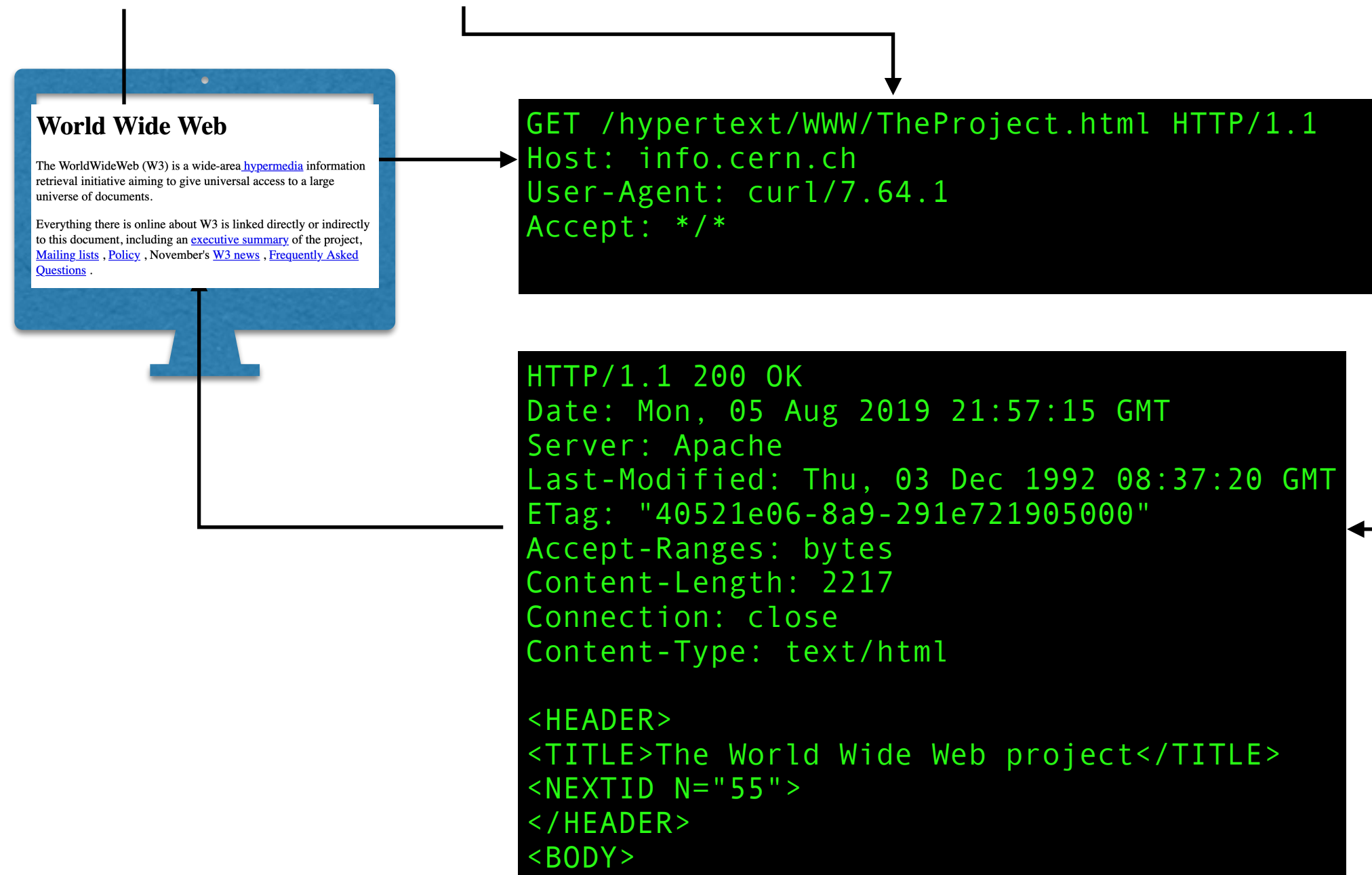
- At a high level, an HTTP response has the same structure as an HTTP request
- The Start line consists of the HTTP version, followed by status code (200 means OK)





# A GET Response

<http://info.cern.ch/hypertext/WWW/TheProject.html>



# A POST Request

- A **POST** request posts information from a web browser to a server, embedding information in the request body
- To send a POST message, a form would include a method parameter with the value "POST", and an action parameter identifying the URI to be invoked

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>

<h2>A Formidable Form</h2>
<form action="https://requestbin-mpr.herokuapp.com/19643o51" method="POST">
First Name: <input type="text" name="firstName" id="firstName"><br/>
Last Name: <input type="text" name="lastName" id="lastName"><br/>
<input type="submit">

</form>
</body>
</html>
```

## A Formidable Form

First Name:

Last Name:

# A Post Request

POST /19643o51 HTTP/1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Content-Type: application/x-www-form-urlencoded

Origin: https://mprogers.github.io

Cookie: session=eyJyZWNLbnQiOlSiMTk2NDNvNTEiXX0.ECs3hA.wfBrH7UCCuHEsP8gbpRh1T8143A

Content-Length: 36

Accept-Language: en-us

Host: requestbin-mpr.herokuapp.com

User-Agent: Mozilla/5.0

Referer: https://mprogers.github.io/PostDemo.html

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

firstName=John+A.&lastName=MacDonald

## A Formidable Form

First Name:

Last Name:

```
<form action="https://..." method="POST">
First Name: <input type="text" name="firstName" id="firstName"><br/>
Last Name: <input type="text" name="lastName" id="lastName"><br/>
<input type="submit">
</form>
```

HTTP/1.1 200 OK

Content-Type: text/html; charset=utf-8

Set-Cookie: session=eyJyZWNLbnQiOlSiMTk2NDNvNTEiXX0.ECs3ig.2l6neS\_z2hbzVn84UGNDqErKDHg; HttpOnly; Path=/

Via: 1.1 vegur

Date: Tue, 06 Aug 2019 16:08:42 GMT

Content-Length: 3

Connection: keep-alive

Server: unicorn/19.9.0

Sponsored-By: https://www.runscope.com

ok



# HTML

- HTML, aka Hypertext Markup Language, is the language that underlies all web pages
- It is a markup language, and uses tags to convey meaning. These tags are understood by web browsers, that then render the content accordingly.
- It was invented by Tim Berners-Lee (yup, the same guy who invented HTTP - he was very prolific!)
- HTML is really all about defining the *structure* of a document - where are the paragraphs, tables, major headings, lists, etc. A document's *appearance* depends on Cascading Style Sheets (CSS), not covered in this course.



run

share

+ new repl

my repls

talk



mprogers (0)

index.html

saved

```
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <title>Behold, a Web Page of Power and Awe</title>
7 <link href="style.css" rel="stylesheet" type="text/css" />
8 </head>
9 <body>
10 <h1>Internet of Things</h1>
11 <p>HTML uses tags to mark up content, so that a web browser can
12 interpret/render them in an appropriate fashion</p>
13 <p>For example, I am writing <em>ordinary</em> ASCII text, but I
have surrounded these words with paragraph tags <p> and </p>. These tags are not visible, but rather, serve as
instructions to the browser to render the enclosed text as a
paragraph. </p>
14 <p>Most tags come in pairs - a starting tag, and an ending tag
with the same name but preceded by a forward slash. Exceptions
to this rule, will not be covered here.</p>
15 <h2>A Few Select Tags</h2>
16 <ol>
17 <li><head></li>
18 <li><title></li>
19 <li><body></li>
20 <li><h1></li>
21 <li>etc ... </li>
22 </ol>
23 <h2>A Fabulous Form</h2>
24 <form action="somewhere.html" method="POST">
25   Name: <input type="text" name="customerName"><br/>
26   Comment: <input type="text" name="comment"><br/>
27   <input type="submit" value="Add Comment"/>
28 </form>
29 <script src="script.js"></script>
30 </body>
31 </html>
```

You write in HTML,  
marking up content  
by wrapping it in tags

result

console



https://EmotionalAuthenticRectangle--mprogers.repl.co



## Internet of Things

HTML uses tags to mark up content, so that a web browser can interpret/render them in an appropriate fashion

For example, I am writing *ordinary* ASCII text, but I have surrounded these words with paragraph tags `<p>` and `</p>`. These tags are not visible, but rather, serve as instructions to the browser to render the enclosed text as a paragraph.

Most tags come in pairs - a starting tag, and an ending tag with the same name but preceded by a forward slash. Exceptions to this rule, will not be covered here.

### A Few Select Tags

1. `<head>`
2. `<title>`
3. `<body>`
4. `<h1>`
5. etc ...

### A Fabulous Form

Name:

Comment:

Add Comment

The browser  
renders it  
appropriately

# Key Tags

- There are a whole slew of tags, check them out [here](#).