COSC 360: Topic 1 Continued

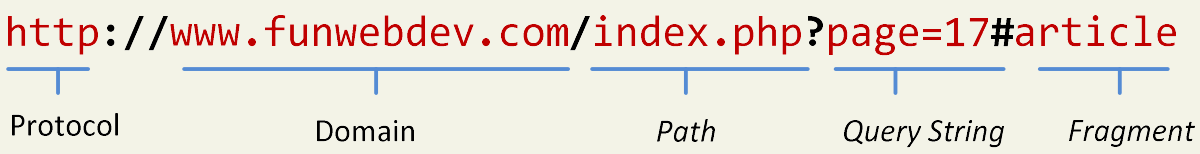
Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

URL Components

In order to allow clients to request particular resources from the server, a naming mechanism is required so that the client knows how to ask the server for the file.

For the web that naming mechanism is the **Uniform Resource Locator** (**URL**).



ftp://example.com/abc.txt

→ sends out an FTP request on port 21

http://example.com/abc.txt

→ transmits an HTTP request on port 80

Uniform Resource Locators

Domain

- The domain identifies the server from which we are requesting resources.
- Since the DNS system is case insensitive, this part of the URL is case insensitive.
- Alternatively, an IP address can be used for the domain:
  - https://1.1.1.1

Uniform Resource Locators

Port

- The optional port attribute allows us to specify connections to ports other than the defaults
- Add a colon after the domain, then specify an integer port number.

Uniform Resource Locators

Path

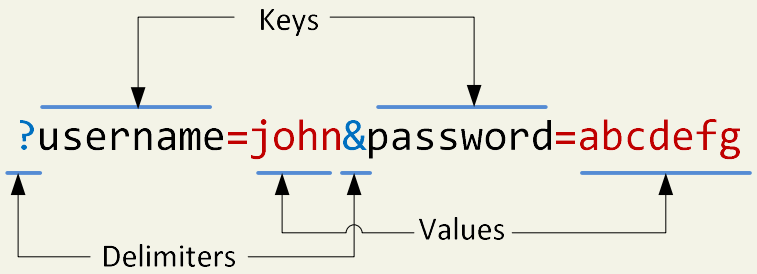
Familiar concept to anyone who has ever used a computer file system.

- The root of a web server corresponds to a folder somewhere on that server.
  - On many Linux servers that path is /var/www/html/
  - On Windows IIS machines it is often /inetpub/wwwroot/
- The path is optional
  - However, when requesting a folder or the top-level page, the web server will decide which file to send you.

Query String

Query strings will be covered in depth when we learn more about HTML forms and server-side programming.

They are the way of passing information such as user form input from the client to the server. In URL's they are encoded as key-value pairs delimited by “&” symbols and preceded by the “?” symbol.



Uniform Resource Locators

Fragment

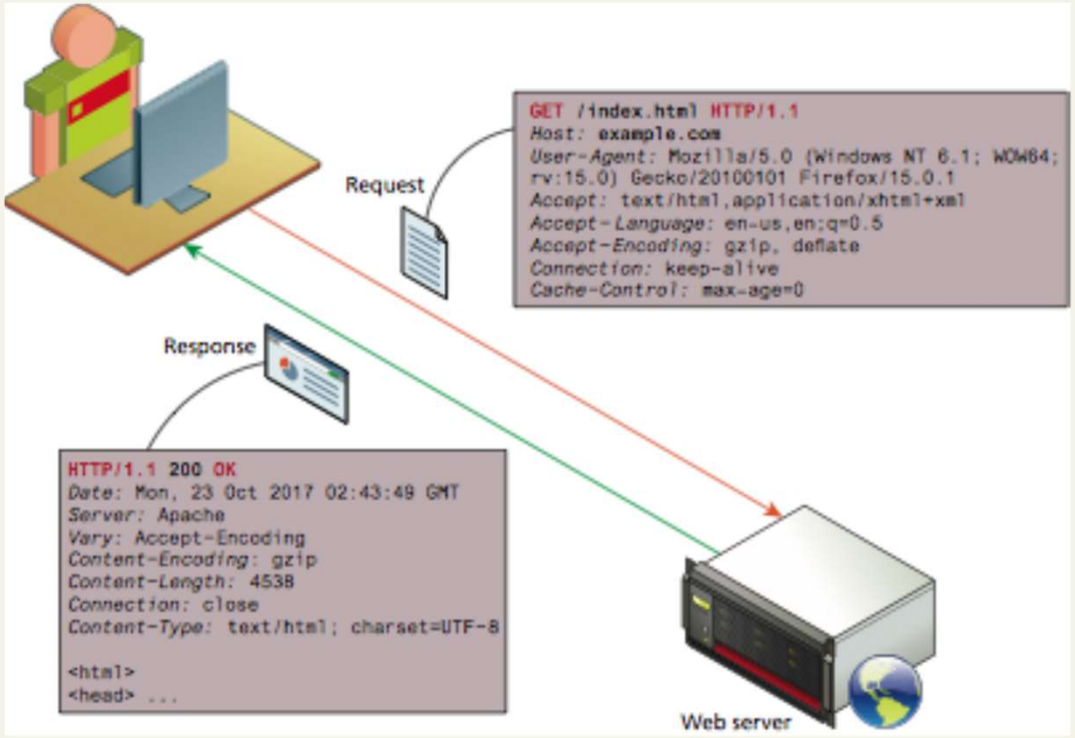
A way of requesting a portion of a page.

- Browsers will see the fragment in the URL, seek out the tag anchor in the HTML, and scroll the website to it.
- For example:
  - https://funwebdev.com/some-page#section1
- This would take us to the specified “section1” part of the “some-page” that we requested

Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

HTTP



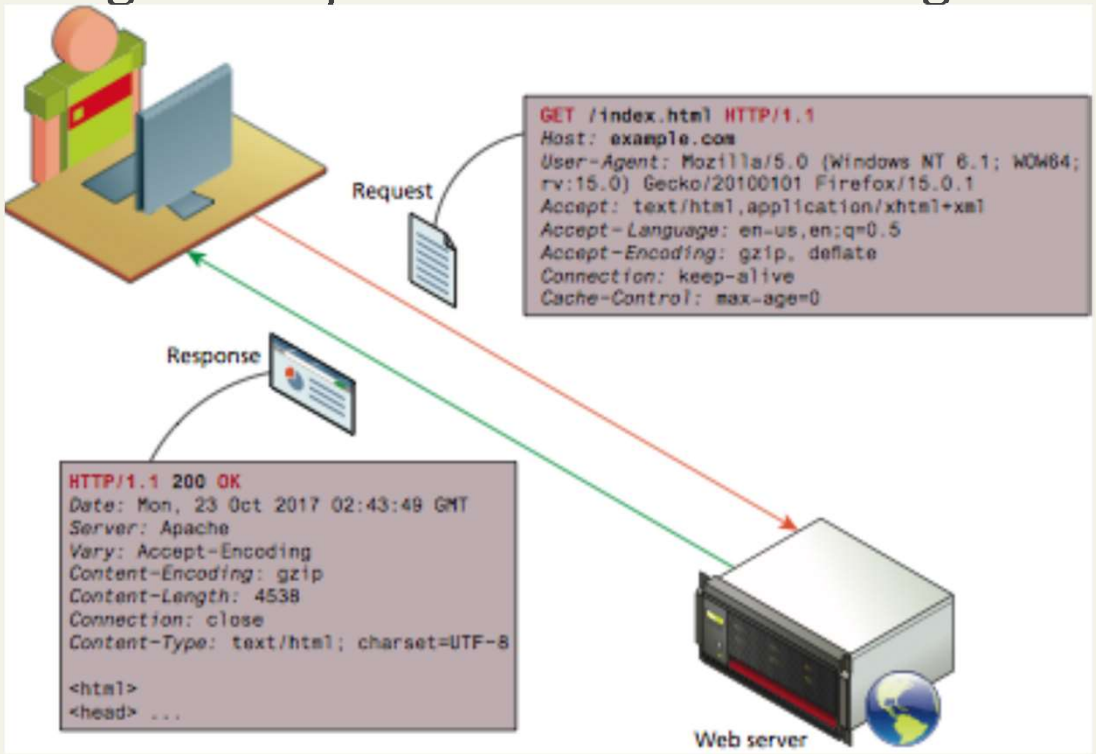
HTTP protocol: TCP connection on port 80 (by default).

Server waits for the request, and then responds with a response code, headers and an optional message (which can include files)

Hypertext Transfer Protocol

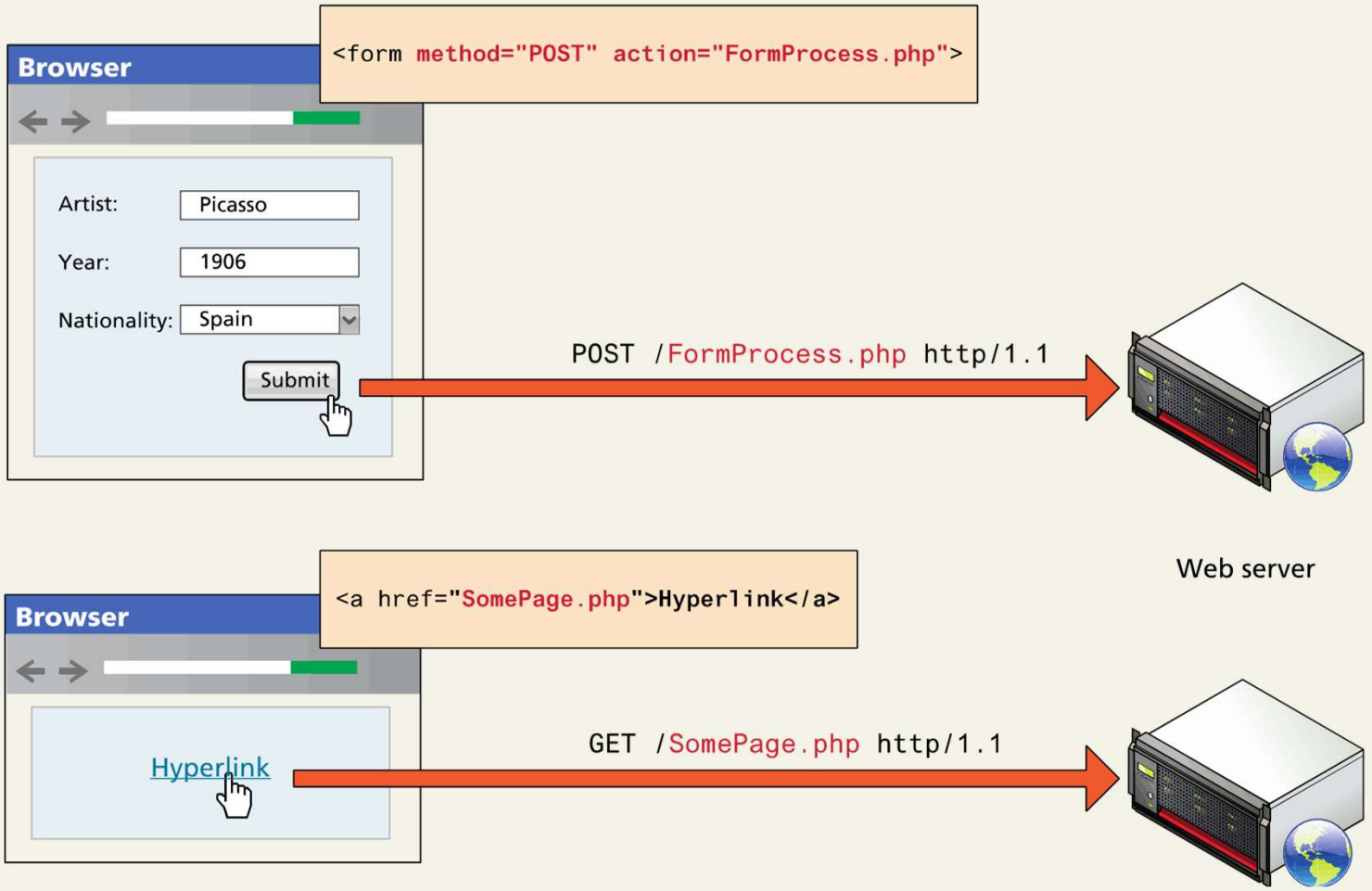
Headers

- **Request headers** include data about the client machine.
- **Response headers** have information about the server answering the request and the data being sent



Hypertext Transfer Protocol

Request Methods



Hypertext Transfer Protocol

Response Codes

- 2## codes are for successful responses,
- 3## are for redirection-related responses,
- 4## codes are **client** errors,
- 5## codes are **server** errors.

200: OK
301: Moved Permanently
304: Not Modified
307: Temporary redirect
400: Bad Request
401: Unauthorized
404: Not found
414: Request URI too long
500: Internal server error

Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

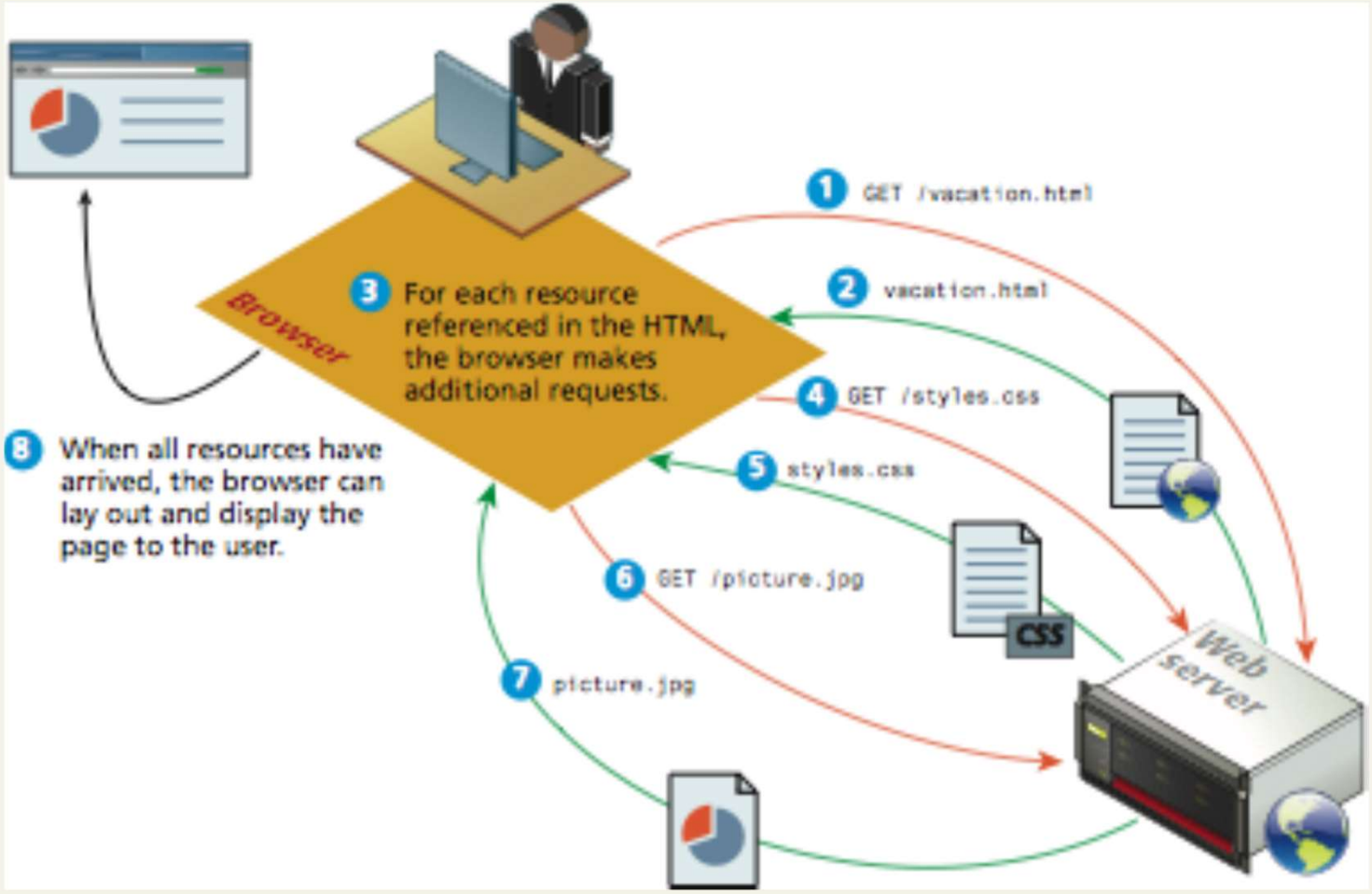
Web Requests

While we as web users might be tempted to think of an entire page being returned in a single HTTP response, this is not in fact what happens.

In reality the experience of seeing a single web page is facilitated by the client's browser which requests the initial HTML page, then parses the returned HTML to find all the resources referenced from within it, like images, style sheets and scripts.

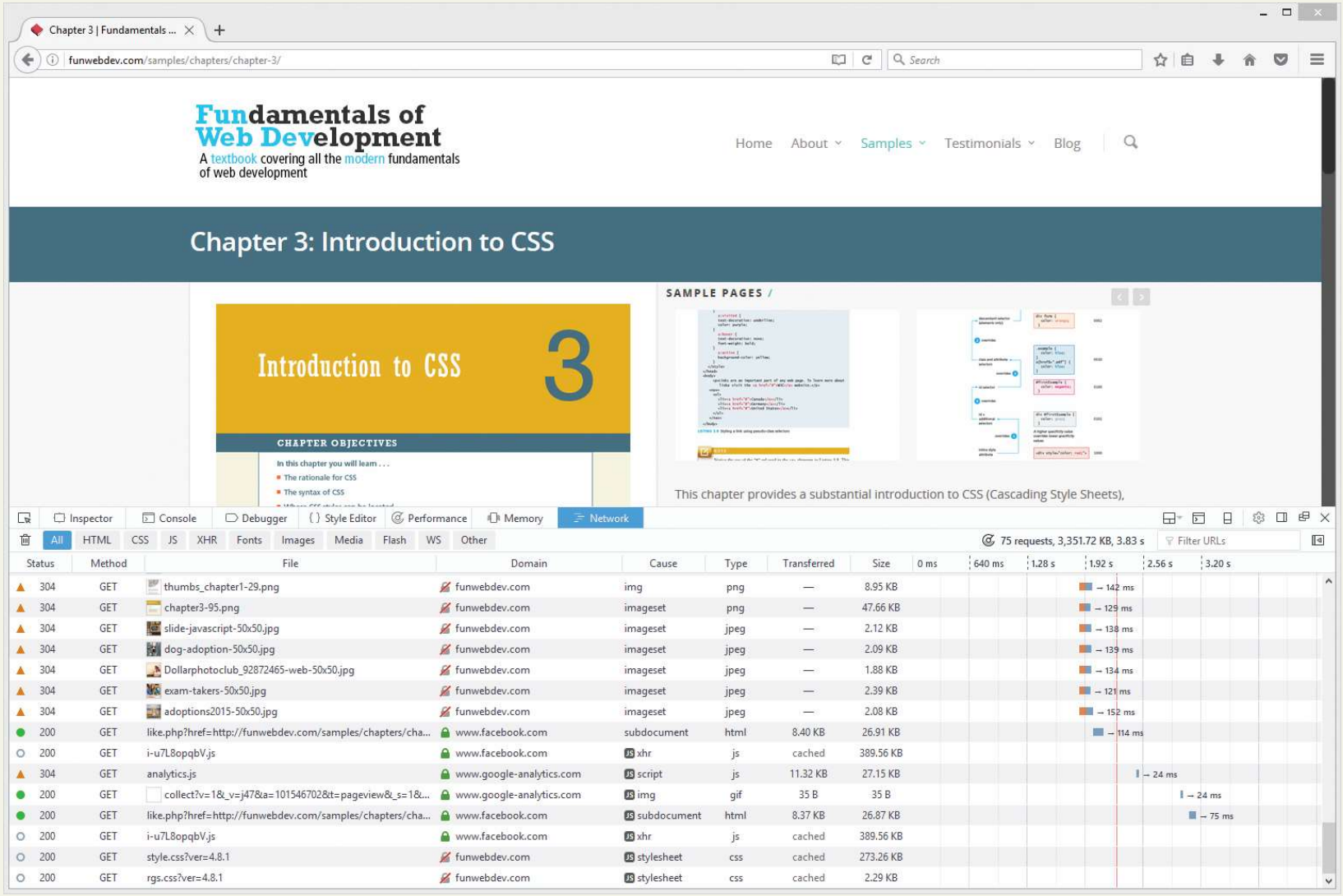
Only when all the files have been retrieved is the page fully loaded for the user

Browser parsing HTML and making subsequent requests



Web Browsers

Fetching a Web Page – Load Times and Cascades



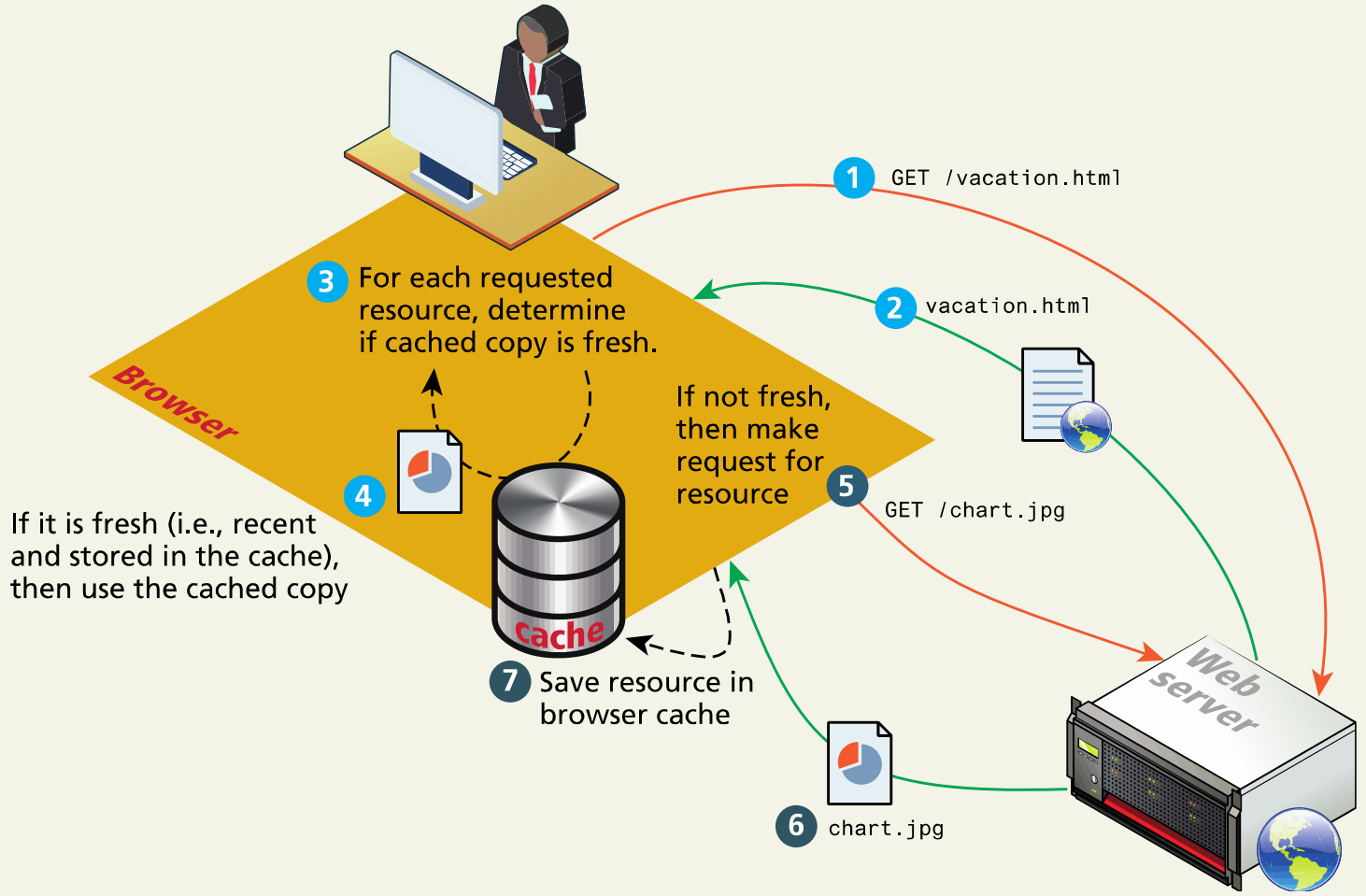
Web Browsers

Browser Rendering

- Interpreting the entire HTML markup together with the image and other assets into a grid of pixels for display within the browser window is called rendering the webpage.
- Implemented differently for each browser (Firefox, Chrome, Safari, Explorer, and Opera)

Web Browsers

Browser Caching



Question

We want to fetch a new page from the web server if it isn't cached using cache control. What kind of request should we use, and where does the cache-control parameter go?

A. HEAD, Request Body

B. GET, Request Body

C. GET, Request Headers

D. POST, Request Headers

Question

We want to fetch a new page from the web server if it isn't cached using cache control. What kind of request should we use, and where does the cache-control parameter go?

A. HEAD, Request Body

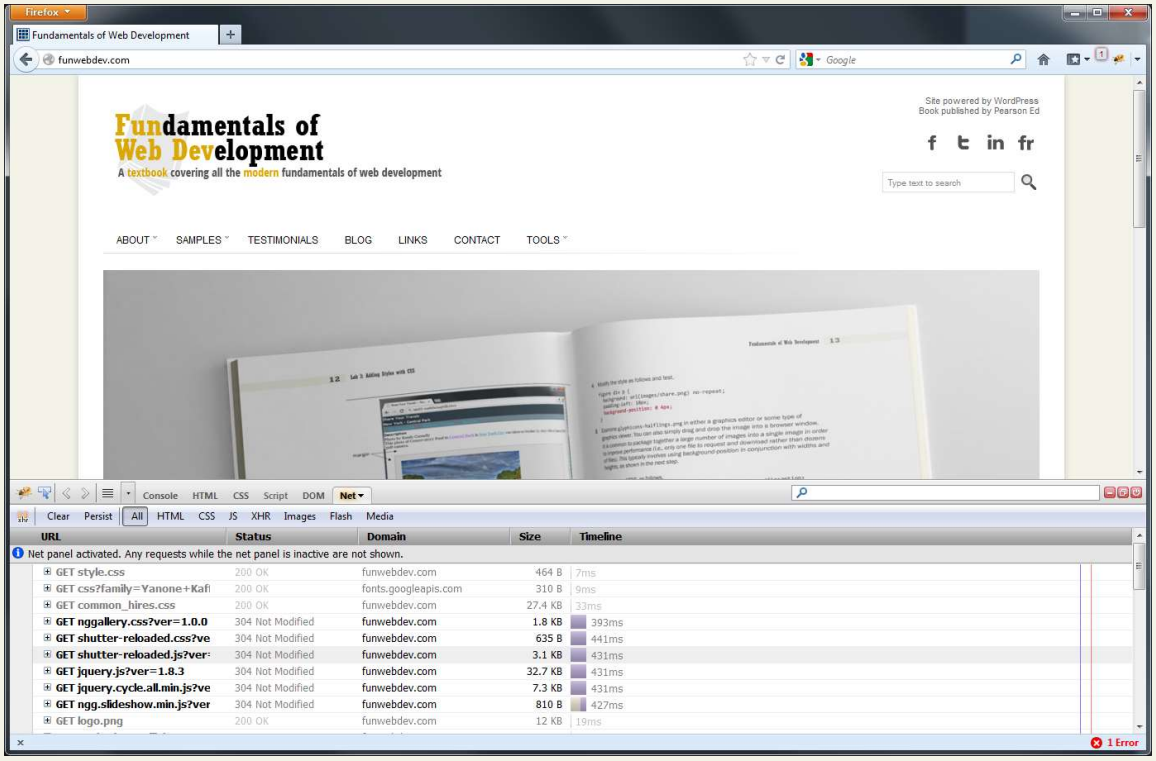
B. GET, Request Body

C. GET, Request Headers

D. POST, Request Headers

Browser Tools for HTTP

Modern browsers provide the developer with tools that can help us understand the HTTP traffic for a given page.



Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

Web Servers

A **web server** is, at a fundamental level, nothing more than a computer that responds to HTTP requests.

Real-world web servers are often more powerful than your own desktop computer

Webservers must choose an **application stack** to run a website. This application stack will include an

- operating system,
- web server software,
- a database,
- and a scripting language for dynamic requests

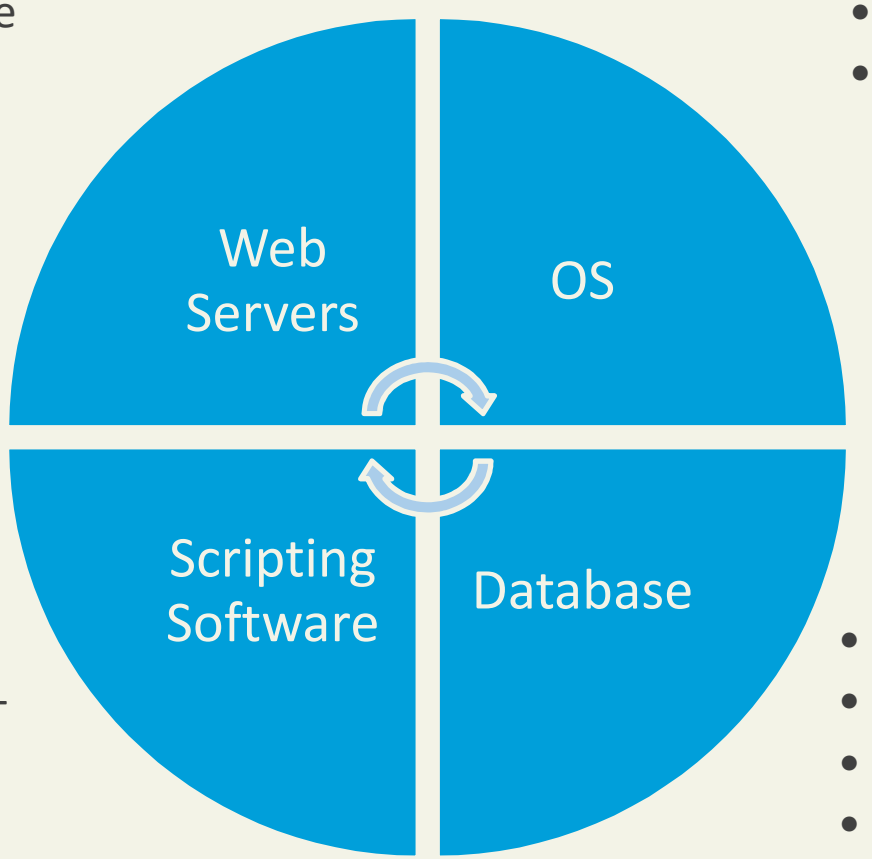
LAMP Software Stack

Throughout this textbook we will rely on the **LAMP software stack**, which refers to the

- Linux operating system
- Apache web server
- MySQL database,
- PHP scripting language

Web Server Technologies

- Apache
- Nginx
- IIS



- Windows
- Linux

- PHP
- ASP.NET
- Python
- Node.js

- MySQL
- PostgreSQL
- Sqlite
- Oracle
- MSSQL
- MongoDB

Web Server Technologies

What are we using in 2018?

Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

Next Topic:

Topic 2: HTML Basics