CSE417: WEB ENGINEERING

Daffodil International University

You Will Be Able To

- ✔ Understand the HTTP
- ✔ Understand HTML5 elements
- ✔ Choose the right tools
- ✔ Know about different character set

Contents

✔ HTTP

✔ Tools to use

✔ HTML Elements

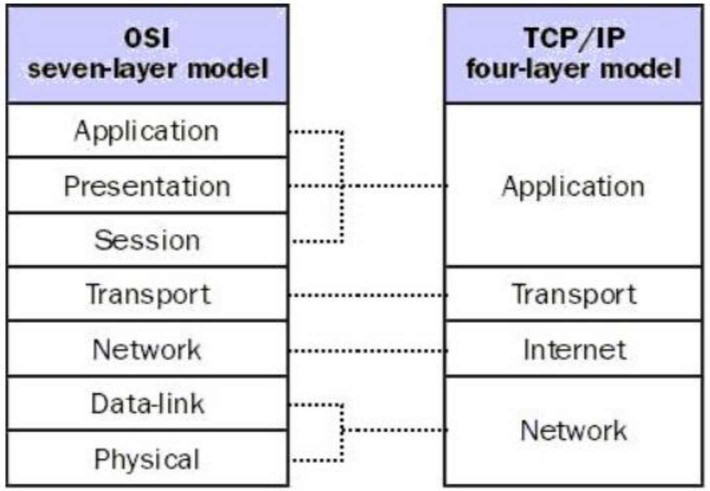
✔ HTML Characters

WEB

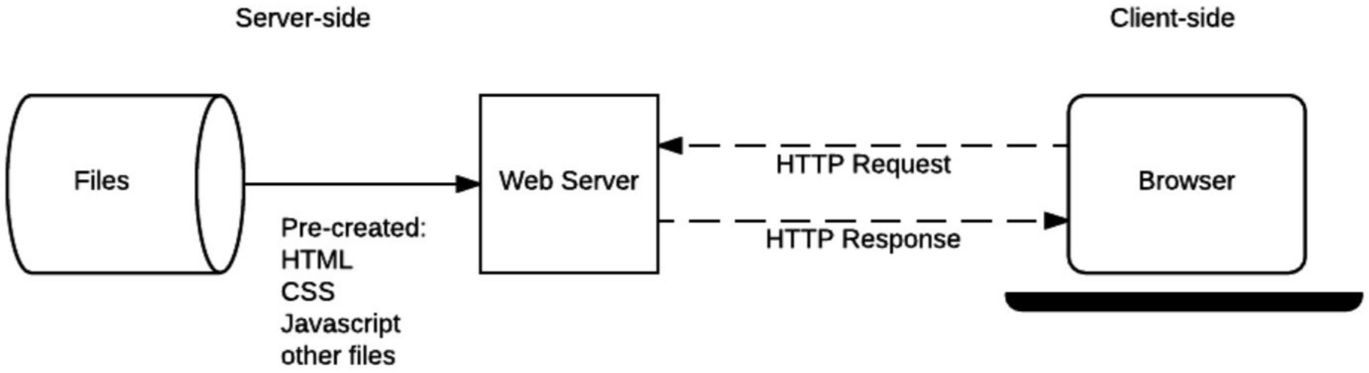
- World Wide Web [New]
  - An infrastructure that allows to easily develop, deploy, and use distributed systems
- Distributed systems
  - A system in which components located on networked computers communicate and coordinate their actions by passing messages in order to achieve a common goal
- The web uses the **Hypertext Transfer Protocol** to communicate (Communication) **protocol**
  - A defined system that allows two or more entities to transmit information via any kind of variation of a physical quantity
  - It defines the rules, syntax, semantics and synchronization of communication

HTTP

- From CSE313 we already know,
  - Web clients (web browsers) and web servers use HTTP (Hypertext Transfer Protocol) to communicate with each other.
  - More generally, HTTP is an application-layer protocol for



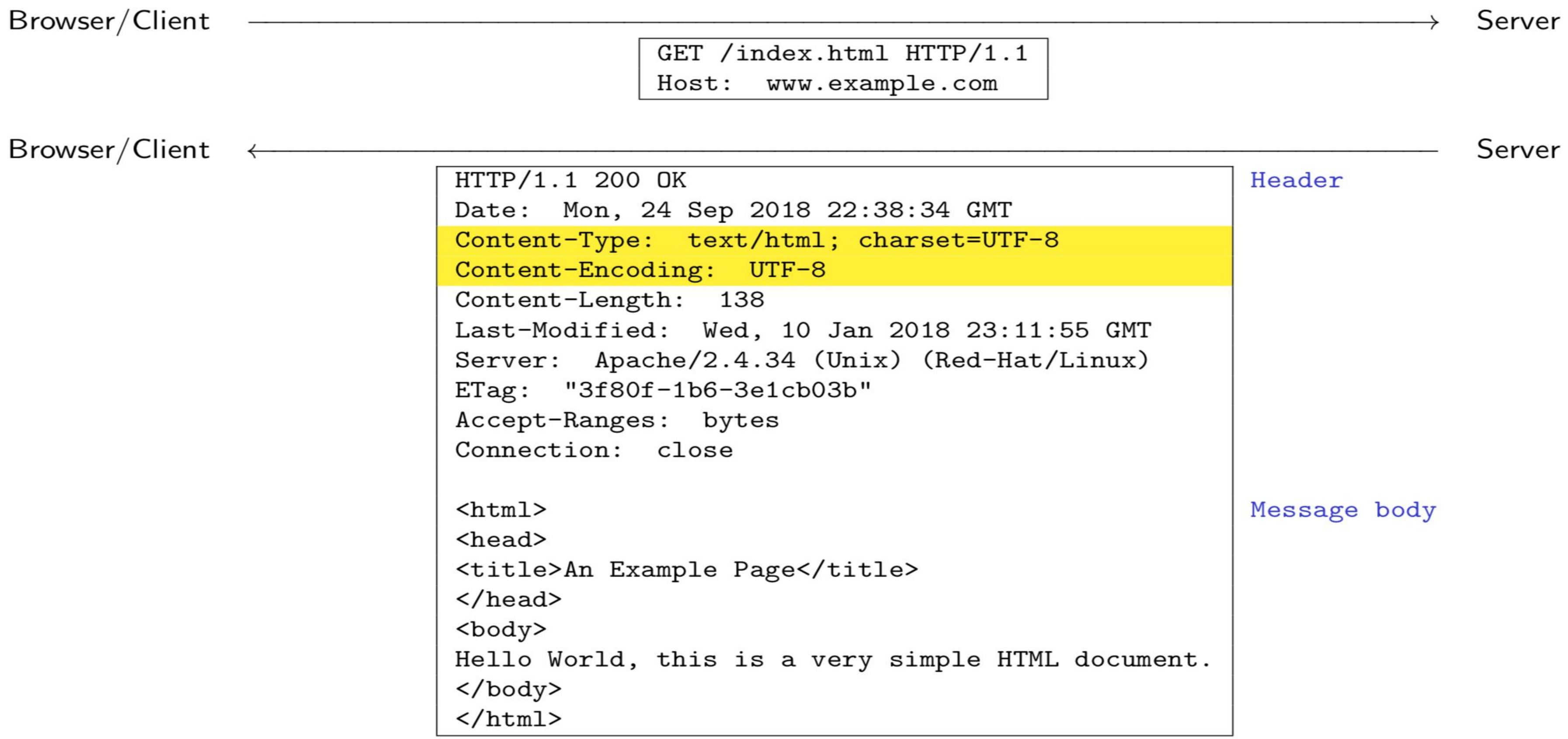
OSI seven-layer model	TCP/IP four-layer model
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data-link	Network
Physical	



History of HTTP

- 1991 – HTTP 0.9
- 1996 – HTTP/1.0
- 1997 – HTTP/1.1
- 1997-2014
  - Minor improvements and clarifications of HTTP/1.1 are developed
- 2015 – HTTP/2
  - Major revision of HTTP with focus on efficiency and privacy improvements
  - HTTP/2 allows a server to push resources to client even before they are requested
  - HTTP/2 puts more emphasis on encrypted connections

HTTP Requests



```
GET /index.html HTTP/1.1
Host:  www.example.com
```

```
HTTP/1.1 200 OK
Date:  Mon, 24 Sep 2018 22:38:34 GMT
Content-Type:  text/html; charset=UTF-8
Content-Encoding:  UTF-8
Content-Length:  138
Last-Modified:  Wed, 10 Jan 2018 23:11:55 GMT
Server:  Apache/2.4.34 (Unix) (Red-Hat/Linux)
ETag:  "3f80f-1b6-3e1cb03b"
Accept-Ranges:  bytes
Connection:  close

<html>
<head>
<title>An Example Page</title>
</head>
<body>
Hello World, this is a very simple HTML document.
</body>
</html>
```

Wikipedia Contributors: Wikipedia, The Free Encyclopedia, 16 September 2018 23:26
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol. [accessed 13 Sep 2017]

Hypertext and HTML

- The HTML specifies the static part/ elements of a web page
- Hypertext documents contain links to other hypertext documents, creating an associative trail that readers can choose to follow
- Markup , more general term for tags which provide additional information about document structure, content type, formatting, etc.

HTML is an evolving…

- HTML is an evolving standard (as new technology/tools are added)
  - HTML 1 (Berners-Lee, 1989): very basic, limited integration of multimedia
    - in 1993, Mosaic added many new features (e.g., integrated images)
  - HTML 2.0 (IETF, 1994): tried to standardize these & other features, but late
    - in 1994-96, Netscape & IE added many new, divergent features
  - HTML 3.2 (W3C, 1996): attempted to unify into a single standard
    - but didn't address newer technologies like Java applets & streaming video
  - HTML 4.0 (W3C, 1997): current standard
    - attempted to map out future directions for HTML, not just react to vendors
  - XHTML 1.0 (W3C, 2000): HTML 4.01 modified to conform to XML standards
  - XHTML 1.1 (W3C, 2001): "Modularization" of XHTML 1.0

In this course…

You should design your webpages to the XHTML 1.0 Strict standard (or better).

I will be enforcing this standard in your assessments (as much as possible).

More will be said when I discuss Document Type Declarations (soon).

Web Development Tools

- Many high-level tools exist for creating Web pages

  e.g., Microsoft FrontPage, Netscape Composer, Adobe PageMill, Macromedia DreamWeaver, HotDog, … also, many applications have "save to HTML" options (e.g., Word)

  …for most users who want to develop basic, static Web pages, these are fine (but they still may produce poorly structured HTML code)

- Assembly language vs. high-level language analogy

- So, why are we learning low-level HTML using a basic text editor?
  - may want low-level control
  - may care about size/readability of pages
  - may want to "steal" page components and integrate into existing pages
  - may want dynamic features such as scripts or applets
  - remote editing of web pages may only be possible using a basic text editor

Choosing the right tools?

- What Not To Use in this course:
  - Any drag and drop website
  - ‘Save as html’ – this generates poor html code
  - No software that can convert your design to code
- What To Use:
  - Any Text-editors and/or IDEs. ie, Notepad, Notepad++, SublimeText, PHP Storm, VSCode etc.
  - You CAN NOT use CSS frameworks like Bootstrap
    - **<u>You can if and only if</u>** you get approval for a higher level projects (you are already a developer!)

Elements and Attributes(1/5)

- The HTML5 specification defines a set of elements, attributes, and attribute values and their meanings (semantics)
  - (there are more than 100 different elements alone)
- Authors of HTML documents should not use elements, attributes, or attribute values for purposes other than their intended semantic purpose
  - otherwise documents might not be processed correctly (still, most authors violate this rule)

Elements and Attributes(2/5)

- HTML5 follows the separation of concerns design principle:
  - a system should be divided into parts with functionality that overlaps as little as possible
  - in HTML5 semantics and presentation are (mostly) separated
- For the full specification of the most recent version see
  - S. Faulkner, A. Eicholz, T. Leithead, A. Danilo, S. Moon, editors: HTML 5.2. W3C Recommendation, 14 December 2017. https://www.w3.org/TR/html52/ (accessed 09 September 2019)

Elements and Attributes(3/5)

- Most elements consist of a start tag and a matching end tag, with some content in between
- The general form of a start tag

  <tagName attrib1="value1" ... attribN="valueN">
- A end tag / closing tag takes the form

  </tagName>
- Examples:

  <*title >My first HTML document </title >*
  <a href="http://cgi.csc.liv.ac.uk/">CS Website </a>

Elements and Attributes(4/5)

- **HTML Attributes**
  - All HTML elements can have **attributes**
  - Attributes provide **additional information** about an element
  - Attributes are always specified in **the start tag**
  - Attributes usually come in name/value pairs like: **name="value"**
- Example

  <a href="https://www.w3schools.com">This is a link</a>

Elements and Attributes(5/5)

- Nested HTML Elements
  - HTML elements can be nested (elements can contain elements).
  - All HTML documents consist of nested HTML elements.
- Suggestion
  - Use Lowercase Attributes
  - Quote Attribute Values

Character Encodings

- Computers operate on bits (0/1) and sequences of bits
- To store a text, it needs to be encoded as a sequence of bits
    - To retrieve a text, a sequence of bits needs to be decoded back to a sequence of characters
- Early examples of such encodings are
    - 7-bit ASCII (American Standard Code for Information Interchange)
    - 8-bit ANSI (American National Standards Institute)
        - 8-bit Windows-1252
        - 8-bit Mac OS Roman
- However these allow to encode at most 256 characters
    - the languages of the world contain many more characters

Character Encodings

- UTF-8 is a modern solution to this problem:
  - (Almost?) every known character is mapped to a sequence of 1x8 bits to 4x8 bits
- Within UTF-8, ANSI characters retain their encoding
- When two systems exchange texts, then they need to know / agree which encoding they are using
  - A HTTP header uses ASCII
  - A HTTP message body can use an arbitrary encoding

Char	ASCII	ANSI	UTF-8	Mac OS Roman
a ˆa ¨a α	1100001	01100001 11100010 11100100	01100001 11100010 11100100 11001001:10100011	01100001 10001001 10001010

HTML Characters

- The HTML5 specification defines a large number of named characters with the general format &name;
  - allows access to non-ASCII and reserved characters
  - Named char, *&lt; &gt*; rendered as ‘<‘ and ‘>’ respectively
- Arbitrary characters can also be accessed using *&#dec*; and *&#xhex*; where *dec* and *hex* are decimal and hexadecimal encodings for a character
  - *&#x0003C;* rendered as ‘<’
  - *&#x00026; as ‘&’*

Exercise

- Why do we need protocols?
- Write two applications of HTTP.
- **READINGS**
    - M Schafer: Ch. 1, 2, 3, 4
    - HTTP
        - Kurose: Ch 2 [Page 98-114]
    - https://www.w3schools.com/html

Acknowledgement

- This module is designed and created with the help from following sources-
    - https://cgi.csc.liv.ac.uk/~ullrich/COMP519/
    - http://www.csc.liv.ac.uk/~martin/teaching/comp519/