




---

---

---

---

---

---

---

---

## Evolution of Internet and WWW

### Internet

- **what is it?** a global network of computers
- evolved from **Advanced Research Projects Agency (ARPANET)**
- ARPANET allowed multiple users to send/receive data simultaneously over the same communication path (e.g. cable or phone lines)

### Packet Switching

- **what is it?** a technique in which digital data is sent in small **packets** (bundles)
- each **packet** contains address, error-control and sequencing information
  - **address:** allow packets to be routed to their destinations
  - **sequencing:** helped in reassembling the packets (could arrive out of order)

### TCP/IP

- **what is it?** combined set of communication protocols for the Internet
- **TCP → Transmission Control Protocol**
  - ensures messages were properly routed from sender to receiver
  - message arrive intact
- **IP → Internet Protocol**
  - each computing device has a unique IP address (IPv4 and IPv6)

TCSS 460 - Summer 2020

---

---

---

---

---

---

---

---

## Evolution of Internet and WWW

### TCP/IP continued

#### IPv4

- **Internet Protocol version 4 (IPv4)** is the most deployed Internet protocol that is used to identify devices on a network using hierarchical addressing scheme
  - IPv4 uses 32-bit address scheme that allows a total of  $2^{32}$  or **4,294,967,296** addresses
  - provide low-cost and low-speed communication for power constrained devices
- as more and more devices get connected to the Internet, these addresses have been exhausted (about the year 2011)
- IPv4 is succeeded by IPv6
- **Internet Protocol version 6 (IPv6)** is the newest version of Internet protocol and successor to IPv4
  - IPv6 uses 128-bit address scheme that allows a total of  $2^{128}$  or **3.4x10<sup>38</sup>** addresses
  - IPv6 offers some new features not present in IPv4, for example:

TCSS 460 - Summer 2020

---

---

---

---

---

---

---

---

## Evolution of Internet and WWW (cont'd)

### World Wide Web

- **WWW** is a platform that enables computer users to execute web-based applications and to locate and view multimedia-based content
- **HyperText Markup Language (HTML)**

### HTTPS

- **what is it?** **Hypertext Transfer Protocol Secure**
- standard for transferring encrypted data on the web
- combines **HTTP with Secure Socket Layer (SSL)** and **Transport Layer Security (TLS)** to secure communications and identification of information over the web

TCSS 460 - Summer 2020

4

## Web Basics

### web page

- HTML document that describes to a web browser the content and structure
- extension: **.html** or **.htm**

### Hyperlinks

- a mechanism to relate documents to each other
- often, **hyperlinked text** appears underlined and in a different color
- hyperlinks are widely used to reference resources on the Internet
- paths created by hyperlinking create the effect of the "web"
- hyperlinks can reference other **web pages, email addresses, files**, etc.

TCSS 460 - Summer 2020

5

## Web Basics (cont'd)

### URLs

- a **Uniform Resource Locator (URL)** is contains information that directs a browser to the resource to be accessed

- resources can be

- documents,
- images,
- downloadable files
- electronic mailboxes
- programs,
- services,
- among others

#### example 1 (URL)

• <http://myname.mypassword@httpandbook.org:80/mydir/myfile.html?myvar=myvalue#myfrag>

http	scheme (protocol)
myname	username (optional)
MyPass	password (optional)
httpandbook.org	network location (host)
80	port (optional)
/mydir/myfile.html	path (resource)
myvar=myvalue	query string (optional)
myfrag	fragment (optional)

TCSS 460 - Summer 2020

6

## Web Basics (cont'd)

### How the web works?

- **Step 1:** tell the browser the "resource" you want to visit

e.g.

← → www.washington.edu/index.html

- **Step 2:** browser decodes URL

- decoding means two things:
  1. check the prefix of the URL (usually http or https)
  2. get the URL's hostname and ask the **Domain Name System (DNS)** to translate this into unique location (IP address)

e.g.

```
Decoding    http://washington.edu/index.html ...
results:
  prefix: http://
  hostname: washington.edu
  web server IP address: 128.208.178.140
```

TCSS 460 - Summer 2020

3

## Web Basics (cont'd)

### How the web works?

- **Step 3:** browser contacts web server and requests web page
  - browser sets up a communications channel with the server and uses this channel to send along a request of the web page

e.g.

```
Dear 128.208.178.140,

Greetings! At your earliest convenience, please send
me the washington.edu web page located at index.html

Here is some additional information you may need:
- accept: text/html
- user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
  x.y; rv:42.0) Gecko/20100101 Firefox/42.0
- Method: GET

Sincerely,
My Browser
```

TCSS 460 - Summer 2020

4

## Web Basics (cont'd)

### How the web works?

- **Step 4:** web server decodes the "http" request
  - decoding http requests involves a number of operations
    1. if web server is shared among multiple user accounts, server begins by locating the user account that owns requested resource
    2. server then uses the page address to find the directory that holds the page (and the file in which the page is stored)

e.g.

```
Decoding    http://washington.edu/index.html ...
results:
  user account: uofwashington
  directory: uwashington_folder
  filename: index.html
```

TCSS 460 - Summer 2020

5

## Web Basics (cont'd)

### How the web works?

- **Step 5:** web server sends the web page representation (response) to browser

e.g.

```
Dear My Browser,
Your request has been processed successfully. Here is the
web file you requested.
Washington.edu Web Server
```

- **Step 6:** web browser decodes the web page file

- decoding the page file means

- looking for text to display
- instructions on how to display that text
- look for other resources by the page (e.g. images, fonts, etc.)

e.g.

```
Decoding index.html...
Results:
  text: received
  formatting: request style.css
  images: request logo.jpg, pic1.jpg, ...
  audio: none
  video: none
  data: none
```

TCSS 460 - Summer 2020

10

## Web Basics (cont'd)

### How the web works?

- **Step 7:** if web page requires more resources, browser asks the server to pass along those resources

e.g.

```
Dear 128.208.178.140,
We need more resources from you please. Can you send us the
following files:
- style.css
- logo.jpg
- pic1.jpg
```

- **Step 8:** for each requested resource, the web server locates the associated files and sends the response back to the browser

e.g.

```
Dear My Browser,
Your request is being processed. We will send you the
requested resources shortly. Sincerely,
Washington.edu Web Server
```

TCSS 460 - Summer 2020

11

## Web Basics (cont'd)

### How the web works?

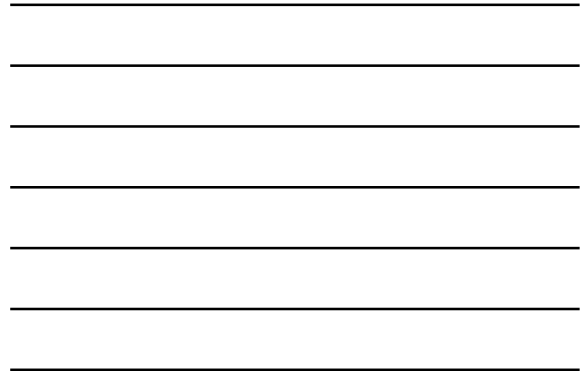
- **Step 9:** web browser renders the HTML file to display text, images and other resources in a graphical representation

e.g.



TCSS 460 - Summer 2020

12



---

---

---

---

---

---

## 16

## 17

## 18

## Web Basics (cont'd)

### HTTP Methods (cont'd)

HTTP provides a standard set of methods that form the interface for a resource

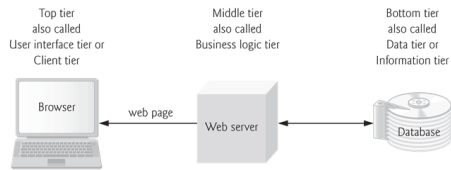
- **PUT**
  - requests that the server replaces the state of the target resource
- **DELETE**
  - requests that the server remove the entity located at specified URL
- **OPTIONS**
  - requests the server return information about its capabilities
- **PATCH**
  - requests that the server do a partial update of the entity at specified URL
- **TRACE**
  - requests that the server return the request it received

TCSS 460 - Summer 2020 <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

19

## Multitier Application Architecture

- web-based applications are often multitier applications
  - sometimes referred to as **n-tier** applications
- idea: functionality is divided into separate tiers



example of a three-tier architecture

TCSS 460 - Summer 2020

20

## Multitier Application Architecture (cont'd)

### bottom tier (or data / information tier)

responsible for maintaining applications (current) data

### middle tier

- controls interactions between application's clients and its data (intermediary)
- **controller logic** processes client requests (such as requests to view a product catalog) and retrieves data from the database
- **presentation logic** then processes data from the information tier and presents the content to the client
- **business logic** enforces business rules
  - how clients should access data
  - how applications should process data

### top tier (or client tier)

application's user interface (typically a web browser), which gathers input and displays output

web applications typically present data to clients as HTML documents

TCSS 460 - Summer 2020

21

## Client-Side Scripting

### client-side scripting

- runs on the **client side** (typically a browser) → uses JavaScript
- can be used to
  - interact with the browser
  - enhance web pages
  - adds client/server communication between browser and server
- **limitations:**
  - **browser dependency:** scripting host must support the scripting language and capabilities
  - **restricted local file system access:** scripts are restricted from arbitrarily accessing the local hardware and file system for security reasons
  - **data integrity:** sensitive information, such as passwords or other personally identifiable data, should not be on the client
  - **security issues:** placing certain operations in JavaScript on the client can open web applications to security issues

TCSS 460 - Summer 2020

22

## Server-Side Scripting

### server-side scripting

- server-side scripts offer programmers more flexibility in many ways
- this enables clients to obtain the most current flight information from database
- **advantages:**
  - server-side scripting languages have a wider range of programmatic capabilities than client-side equivalents
  - server-side scripts have access to server-side software that **extends** server functionality
    - Microsoft web server: **Internet Server Application Program Interface**
    - Apache HTTP Servers: uses **modules**

TCSS 460 - Summer 2020

23

## World Wide Web Consortium (W3C)

- 1994: Tim-Berners Lee founded the W3C whose mission is
  - "to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web"
- Open Standards Principles ...
  - **web for all**
  - **web on everything**
- W3C is also a standards organization
  - Web technologies standardized by W3C are called **Recommendations**
  - **current and upcoming W3C recommendations include**
    - HTML 5
    - Cascading Style Sheets (CSS)
    - Extensible Markup Language (XML)

TCSS 460 - Summer 2020

24



## Goals of This Lecture

- You should be comfortable with the basic elements of the web and how the concepts of the Internet/WWW originated
- You should be able to know the behind the scenes steps (in general) of how the web works
- You should be comfortable with the HTTP messaging such as HTTP request and response
- You should be able to define client-side and server-side scripting as well as how to create multitier applications (e.g. 3 tier, n-tier). You should be comfortable with the limitations and the advantages of creating n-tier applications
- You should know the difference between the Internet and World Wide Web

TCSS 460 - Summer 2020

25

---

---

---

---

---

---

---

---

## Module Topics



- How Web Works
- HTML
- CSS

TCSS 460 - Summer 2020

26

---

---

---

---

---

---

---

---