

EECS 368

Programming Language Paradigms

David O. Johnson

Fall 2022

Reminders

- Assignment 2 due: 11:59 PM, Monday, September 19
- Assignment 3 due: 11:59 PM, Monday, October 3

Any Questions?

In-Class Problem Solution

- 8-(9-12) In-Class Problem Solution.pptx

Any Questions?

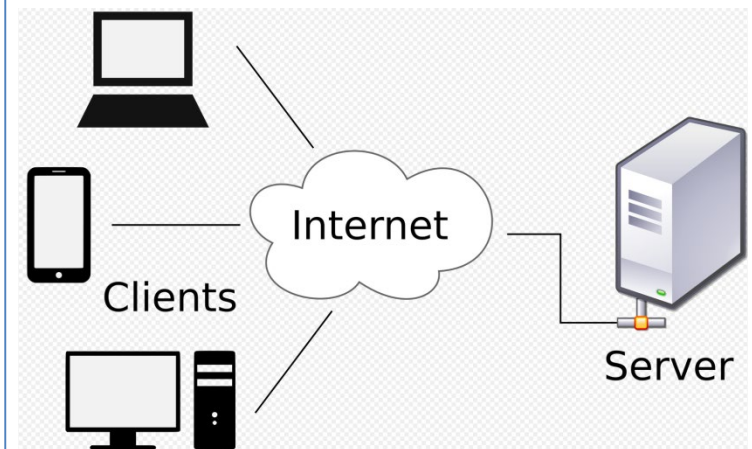
JavaScript and the Browser

- In EECS 168 and 268, you learned how to write programs that run on one computer.
- Up to this point in 368, you have learned how to write JavaScript programs that run on one computer.
- As you may recall, from the first lecture on JavaScript:
 - JavaScript was introduced in 1995 as a way to add programs to web pages in the Netscape Navigator browser.
 - The language has since been adopted by all other major graphical web browsers.
 - It has made modern web applications possible—applications with which you can interact directly without doing a page reload for every action.
- For the rest of our time studying JavaScript, we are going to learn how JavaScript is used in modern web applications!

Client-Server Architecture

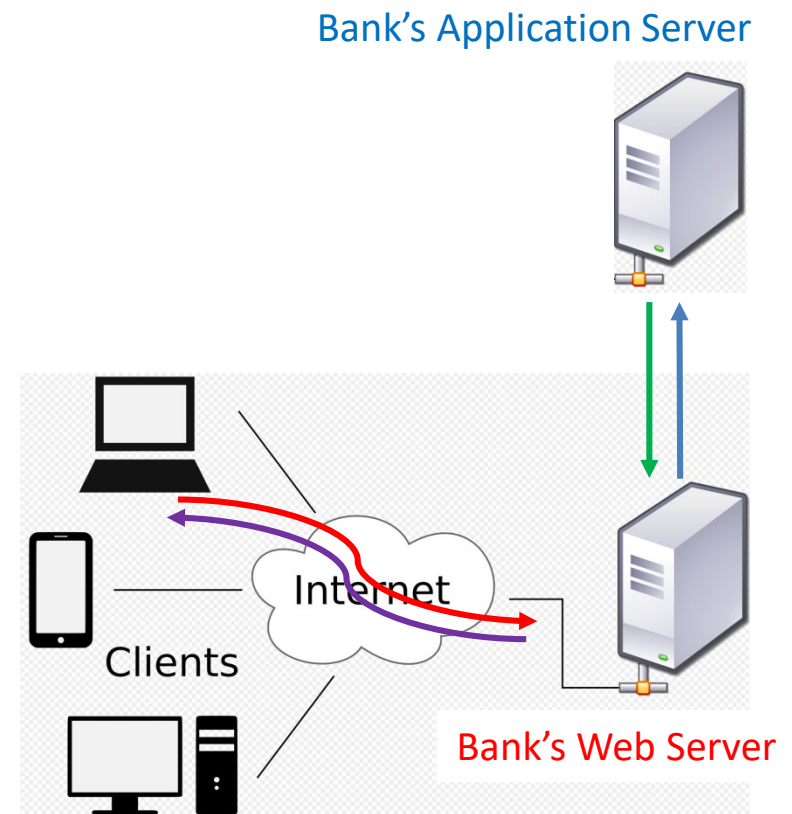
Modern web applications are based on the Client-Server Architecture

- **Servers:**
 - Provide resources or services
 - Await incoming requests from **clients**
 - For example: Canvas service where all your grades and my lectures are stored
- **Clients:**
 - Request a resource or service
 - Initiate communication sessions with **servers**
 - Canvas app on your phone that retrieves your grades and my lectures
- **Communicate over:**
 - Computer network (e.g., Internet)
- **Examples:**
 - Email
 - Network printing
 - Web or phone apps



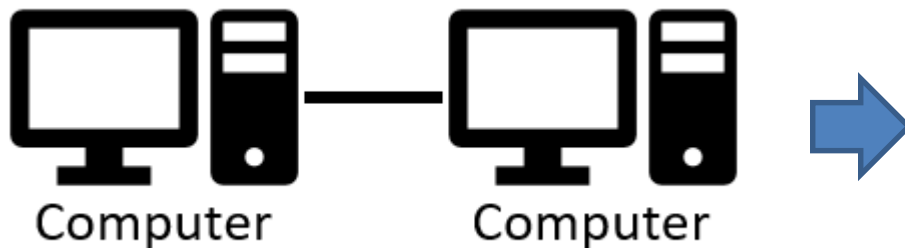
Client-Server Example

- Bank customer accesses online banking services with a web browser (the client)
- Client initiates a login request to the bank's Web Server.
- Customer's login credentials may be stored in a database on an Application Server
- Web Server accesses the Application Server as a client.
- Application Server verifies the customer's login credentials and provides the output to the Web Server.
- Finally, the Web Server returns the result to the client web browser for display.
- In each step of this sequence of client-server message exchanges, a computer processes a request and returns data.
- This is the request-response messaging pattern.



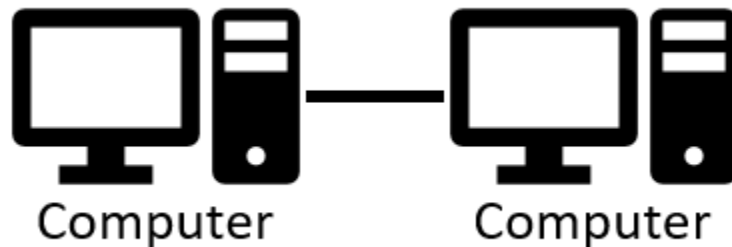
Computer Networks

- Computer networks have been around since the 1950s.
- If you put cables between two computers and allow them to send data back and forth through these cables, you can do all kinds of wonderful things.
- And if connecting two machines in the same building allows us to do wonderful things, connecting machines all over the planet should be even better.
- The technology to start implementing this vision was developed in the 1960s, and the resulting network is called the Internet.
- It has lived up to its promise.



Network Protocol

- A computer can use a network to shoot bits at another computer.
- For any effective communication to arise out of this bit-shooting, the computers on both ends must know what the bits are supposed to represent.
- The meaning of any given sequence of bits depends entirely on the kind of thing that it is trying to express and on the encoding mechanism used.
- A **network protocol** describes a style of communication over a network.
- There are protocols for sending email, for fetching email, for sharing files, and even for controlling computers that happen to be infected by malicious software.
- Most protocols are built on top of other protocols.



OSI Network Protocol Model

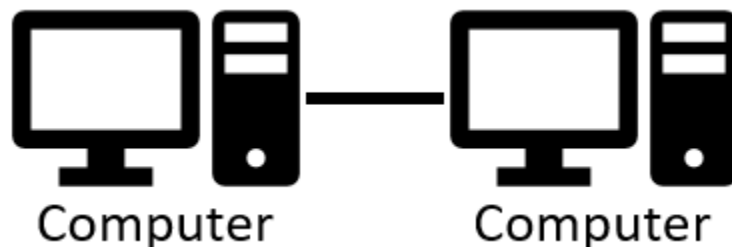
OSI model		
Layer		Function
Host layers	7 Application	High-level APIs, including resource sharing, remote file access
	6 Presentation	Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5 Session	Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4 Transport	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3 Network	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2 Data link	Reliable transmission of data frames between two nodes connected by a physical layer
	1 Physical	Transmission and reception of raw bit streams over a physical medium

Hypertext Transfer Protocol (HTTP)

- Hypertext Transfer Protocol (HTTP) is a protocol for retrieving named resources (chunks of information, such as web pages or pictures).
- It specifies that the side making the request should start with a line like this, naming the resource and the version of the protocol that it is trying to use:

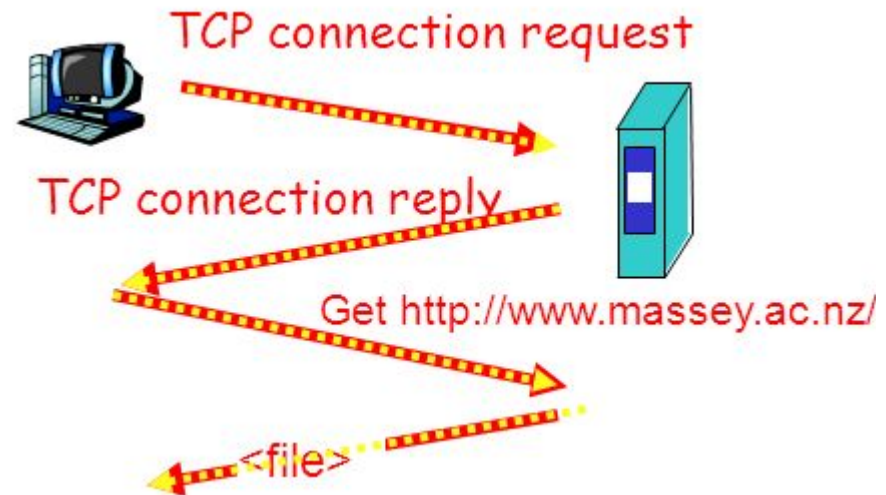
GET /index.html HTTP/1.1

- There are a lot more rules about the way the requester can include more information in the request and the way the other side, which returns the resource, packages up its content.
- HTTP treats the network as a streamlike device into which you can put bits and have them arrive at the correct destination in the correct order.
- We will learn a lot more about HTTP later.



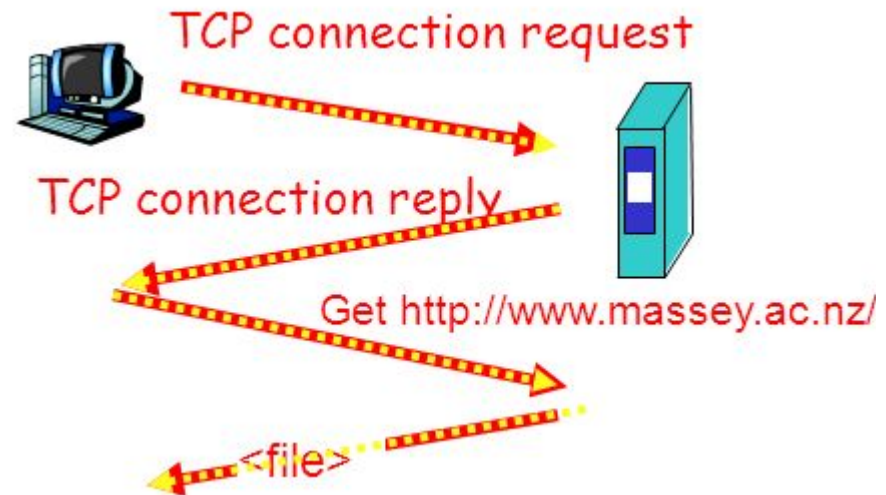
Transmission Control Protocol (TCP)

- Transmission Control Protocol (TCP) is a protocol that ensures you can treat the network as a streamlike device into which you can put bits and have them arrive at the correct destination in the correct order.
- All Internet-connected devices “speak” it, and most communication on the Internet is built on top of it.
- A TCP connection works as follows: one computer must be waiting, or listening, for other computers to start talking to it.
- To be able to listen for different kinds of communication at the same time on a single machine, each listener has a number (called a port) associated with it.
- Most protocols specify which port should be used by default.



Transmission Control Protocol (TCP)

- For example, when we want to send an email using the SMTP protocol, the machine through which we send it is expected to be listening on port 25.
- Another computer can then establish a connection by connecting to the target machine using the correct port number.
- If the target machine can be reached and is listening on that port, the connection is successfully created.
- The listening computer is called the server, and the connecting computer is called the client.
- Such a connection acts as a two-way pipe through which bits can flow—the machines on both ends can put data into it.
- Once the bits are successfully transmitted, they can be read out again by the machine on the other side.



HTTP & TCP in the OSI Protocol Model

OSI model		
Layer		Function
Host layers	7 Application	High-level APIs, including resource sharing, remote file access HTTP
	6 Presentation	Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5 Session	Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4 Transport	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing TCP
Media layers	3 Network	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2 Data link	Reliable transmission of data frames between two nodes connected by a physical layer
	1 Physical	Transmission and reception of raw bit streams over a physical medium

World Wide Web (www)

- The World Wide Web is a set of protocols and formats that allow us to visit web pages in a browser.
- The “Web” part in the name refers to the fact that such pages can easily link to each other, thus connecting into a huge mesh that users can move through.
- To become part of the Web, all you need to do is connect a machine to the Internet and have it listen on port 80 with the HTTP protocol so that other computers can ask it for documents.
- Each document on the Web is named by a Uniform Resource Locator (URL), which looks something like this:

http://eloquentjavascript.net/13_browser.html

```

|      |      |
protocol      server      path

```

- The first part tells us that this URL uses the HTTP **protocol** (as opposed to, for example, encrypted HTTP, which would be https://).
- Then comes the part that identifies which **server** we are requesting the document from.
- Last is a **path** string that identifies the specific document (or resource) we are interested in.

World Wide Web (www)

- Machines connected to the Internet get an IP address, which is a number that can be used to send messages to that machine, and looks something like:
149.210.142.219 or 2001:4860:4860::8888
- But lists of more or less random numbers are hard to remember and awkward to type, so you can instead register a domain name (e.g., ku.edu) for a specific address or set of addresses.
- If you type this URL into your browser's address bar, the browser will try to retrieve and display the document at that URL.
- First, your browser has to find out what IP address ku.edu refers to.
- Then, using the HTTP protocol, it will make a connection to the server at that address and ask for the resource: 13_browser.html.
- If all goes well, the server sends back a document, which your browser then displays on your screen.

Any Questions?

Hypertext Markup Language (HTML)

- HTML is the document format used for web pages.
- An HTML document contains text, as well as tags that give structure to the text, describing things such as links, paragraphs, and headings.
- The tags, wrapped in angle brackets (< and >), provide information about the structure of the document.
- The other text is just plain text.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My home page</title>
  </head>
  <body>
    <h1>My home page</h1>
    <p>
      Hello, I am Marijn and this is my home page.
    </p>
    <p>
      I also wrote a book! Read it
      <a href="http://eloquentjavascript.net">
        here
      </a>.
    </p>
  </body>
</html>
```

Hypertext Markup Language (HTML)

- The document starts with `<!doctype html>`, which tells the browser to interpret the page as modern HTML, as opposed to various dialects that were in use in the past.
- HTML documents have a `head` and a `body`.
- The `head` contains information about the document.
- The `body` contains the document itself.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My home page</title>
  </head>
  <body>
    <h1>My home page</h1>
    <p>
      Hello, I am Marijn and this is my home page.
    </p>
    <p>
      I also wrote a book! Read it
      <a href="http://eloquentjavascript.net">
        here
      </a>.
    </p>
  </body>
</html>
```

Hypertext Markup Language (HTML)

- In this case, the head declares:
 - It uses the UTF-8 encoding, which is a way to encode Unicode text as binary data.
 - The title of this document is “My home page”
 - The title is what will be displayed on the browser tab.
- The document’s body contains:
 - A heading (<h1>, meaning “heading 1”)
 - <h2> through <h6> produce subheadings
 - Two paragraphs (<p>).

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My home page</title>
  </head>
  <body>
    <h1>My home page</h1>
    <p>
      Hello, I am Marijn and this is my home page.
    </p>
    <p>
      I also wrote a book! Read it
      <a href="http://eloquentjavascript.net">
        here
      </a>.
    </p>
  </body>
</html>
```

Hypertext Markup Language (HTML)

Tags come in several forms:

- An element, such as the body, a paragraph, or a link, is started by an opening tag like `<p>` and ended by a closing tag like `</p>`.
- Some kinds of tags do not enclose anything and thus do not need to be closed. The metadata tag `<meta charset="utf-8">` is an example of this.
- Some opening tags, such as the one for the link (`<a>`), contain extra information in the form of `name="value"` pairs.
- These are called attributes.
- In this case, the destination of the link is indicated with `href="http://eloquentjavascript.net"`, where `href` stands for “hypertext reference”.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My home page</title>
  </head>
  <body>
    <h1>My home page</h1>
    <p>
      Hello, I am Marijn and this is my home page.
    </p>
    <p>
      I also wrote a book! Read it
      <a href="http://eloquentjavascript.net">
        here
      </a>.
    </p>
  </body>
</html>
```

Any Questions?

HTML and JavaScript

- In the context of this course, the most important HTML tag is `<script>`.
- This tag allows us to include a piece of JavaScript in a document.
`<h1>Testing alert</h1>`
`<script>alert("hello!");</script>`
- Such a script will run as soon as its `<script>` tag is encountered while the browser reads the HTML.
- This page will pop up a dialog when opened—the alert function resembles prompt, in that it pops up a little window, but only shows a message without asking for input.
- A script tag must always be closed with `</script>`, even if it refers to a script file and doesn't contain any code.
- If you forget this, the rest of the page will be interpreted as part of the script.

HTML and JavaScript

- Including large programs directly in HTML documents is often impractical.
- The `<script>` tag can be given an `src` attribute to fetch a script file (a text file containing a JavaScript program) from a URL.
`<h1>Testing alert</h1>`
`<script src="code/hello.js"></script>`
- The `code/hello.js` file included here contains the same program—`alert("hello!")`.
- When an HTML page references other URLs as part of itself ...
 - for example, an image file or a script
- ... web browsers will retrieve them immediately and include them in the page.

HTML and JavaScript

- Some attributes can also contain a JavaScript program.
- The <button> tag shown next (which shows up as a button) has an **onclick** attribute.
- The attribute's value will be run whenever the button is clicked.
`<button onclick="alert('Boom!');">DO NOT PRESS</button>`
- Use single quotes for the string in the onclick attribute...
- ...because double quotes are already used to quote the whole attribute.

Any Questions?

Summary

- Modern web applications are based on the **Client-Server Architecture**.
 - The **Client** program runs on one computer and the **Server** program runs on another computer.
 - The **Client** and **Server** and programs communicate over a **computer network**.
- **Computer networks** are composed of two or more computers with connections between them that allows them to send data back and forth over the connections.
- A **network protocol** describes a style of communication over a network.
- **Hypertext Transfer Protocol (HTTP)** is a protocol for retrieving named resources (chunks of information, such as web pages or pictures).
- **Transmission Control Protocol (TCP)** is a protocol that ensures you can treat the network as a streamlike device into which you can put bits and have them arrive at the correct destination in the correct order.
- The **World Wide Web (www)** is a set of protocols and formats that allow us to visit web pages in a browser.
- Each document on the Web is named by a **Uniform Resource Locator (URL)**.
- Machines connected to the Internet get an **IP address**, which is a number that can be used to send messages to that machine.
- **Hypertext Markup Language (HTML)** is the document format used for web pages.
- **HTML tags**, wrapped in angle brackets (< and >), provide information about the structure of the document.
- **JavaScript** can be embedded in HTML documents using the **<script>** tag.

Any Questions?

In-Class Problem

- A "Hello, World!" program generally is a computer program that outputs or displays the message "Hello, World!".
- Such a program is very simple in most programming languages.
- It is often used to illustrate the basic syntax of a programming language.
- It is often the first program written by people learning to code a new language.
- It can also be used as a sanity test to make sure that computer software intended to compile or run source code is correctly installed, and that the operator understands how to use it.
- Write a “Hello World!” program in HTML.
- For full credit, include:
 - Tag that tells the browser to interpret the page as modern HTML.
 - Header and body.
 - Tag that tells the browser that the character set is “utf-8”.
 - Tag that will display “Hello World!” on the browser tab.
 - Tags that will display “Hello World!” in the browser window.