# Introduction to Web Application Development, for CS437/637

INSTRUCTOR: BETTY O’NEIL

CLASS MEETS MW 4:00-5:15 ONLINE WITH ZOOM

# Plan for course (more details on syllabus)

1. Introduction (today)
2. HTML (HTML5) and CSS using Duckett, HTML&CSS
3. PHP: using Murach &Harris (2ed or 3ed), PHP and MySQL
4. Javascript: using Purewal, Learning Web App Development (O'Reilly)
5. REST web services, using web resources

HW1: get started as soon as possible!

Password for slides…<br>
Call roll…

# Introduction: Internet vs. World Wide Web

**Internet** is an interconnected ***network*** of thousands of networks and millions of computers (sometimes called host computers or just hosts).

The **World Wide Web**, or **Web** for short, is one of the Internet's most popular ***services***, providing access to over one billion Web pages.

The Web runs over the Internet, using its ability to transfer data reliably (using TCP).

# How the Internet evolved

The Internet, then known as the ARPANET (Advanced Research Projects Agency Network), began in 1969, soon file transfers, emails were flowing.

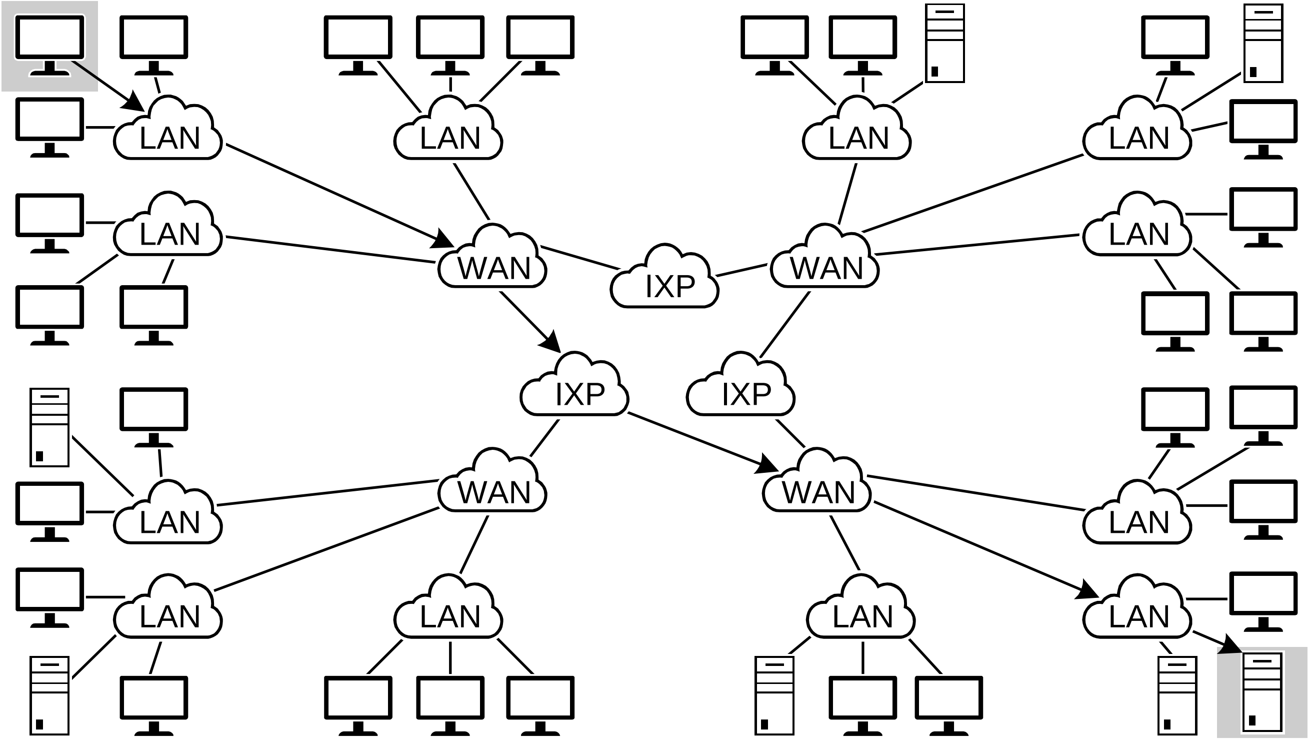
The current TCP/IP protocols date from 1983. The Internet was already international then, but restricted to non-commercial use.

In 1990, it was opened to commercial use….

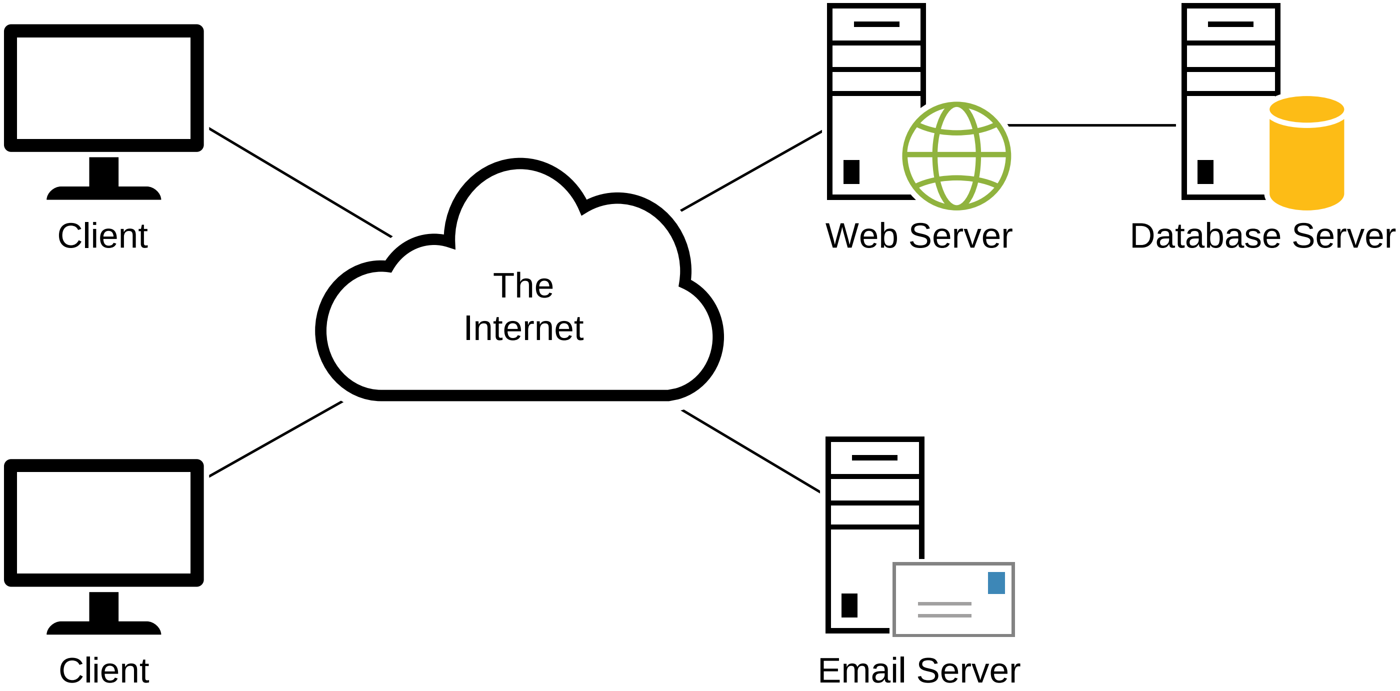
# A Hierarchy of Networks

- Every computer that is connected to the Internet is part of a network.
- You may use a cable or DSL modem to an **Internet Service Provider (ISP)**. At work, you may be part of a **local area network (LAN)** using an ISP that your company has contracted with.
- When you connect to your ISP, you become part of their network. The ISP may then connect to a larger network and become part of their network. The Internet is simply a *network of networks*.

# The architecture of the Internet



# Luckily, the Internet can be viewed like this for our purposes…



# Internet Domain Naming System

- Provides unique ids for all the hosts directly connected to the Internet
  - Example hostname = www.cs.umb.edu, a unique hostname across the Internet.

- Domain naming system (in use since 1983):
  - Top-level domain name:  edu
  - UMB's domain name:  umb.edu
  - Departmental subdomain:  cs.umb.edu
  - Full hostname = www.cs.umb.edu, specifying the departmental web server host

- The Internet's Domain Naming Service (DNS) servers translate a human-readable domain name into the machine-readable IP address.
  - The IP address of www.cs.umb.edu is 158.121.106.222,  also unique across the Internet

# The World Wide Web

- The Web runs over the Internet, using its ability to transfer data reliably (using TCP) and specify hosts with domain names and IP addresses.

- The crucial web protocol is HTTP (Hypertext Transfer Protocol)

- The Web dates from the early 90s.

- Since the Internet was already an international presence, so was the Web.

# Web Accessibility

- Web accessibility means that people with disabilities can use the Web.
- More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web.
- HTML, especially HTML5, plays an important role by providing a standard way of describing what’s on the screen.
- Our web apps will send HTML5 to our users, and so provide base-level accessibility.

# Uniform Resource Locator (URL)

URLs specify locations on the Internet, along with the protocol needed for accessing something there.

We say URLs specify “resources”.

A simple URL has the following format:

**<protocol> :// <hostname or IP address> / <path>**

Example: https://https://www.cs.umb.edu/cs637(our class homepage)

Here hostname = www.cs.umb.edu

Using IP address: https://158.121.106.224/cs637

(however, this no longer works because of heightened security for our web server)

# What is a web application?

**App vs. web app:**

An app (desktop or mobile) is a program that runs directly on the OS of the device.

- Thus has sub-species: Android app, ioS app, Linux app, Windows app, MacOS app, etc., and each works only on its own OS.
- An organization that wants all its users to use “its app” needs to implement all these versions.

A web app runs inside a web browser. The web browser itself is an app on the desktop or mobile device.

- Thus we would expect sub-species Chrome app, Safari app, Firefox app
- But browsers are much more standardized than OSs, so we just say “web app” and expect it to run on any of these modern browsers.
- An organization that wants all its users to use “its webapp” should only need to implement one version (possibly with some conditional code if using non-standardized features).

# A web application has a website

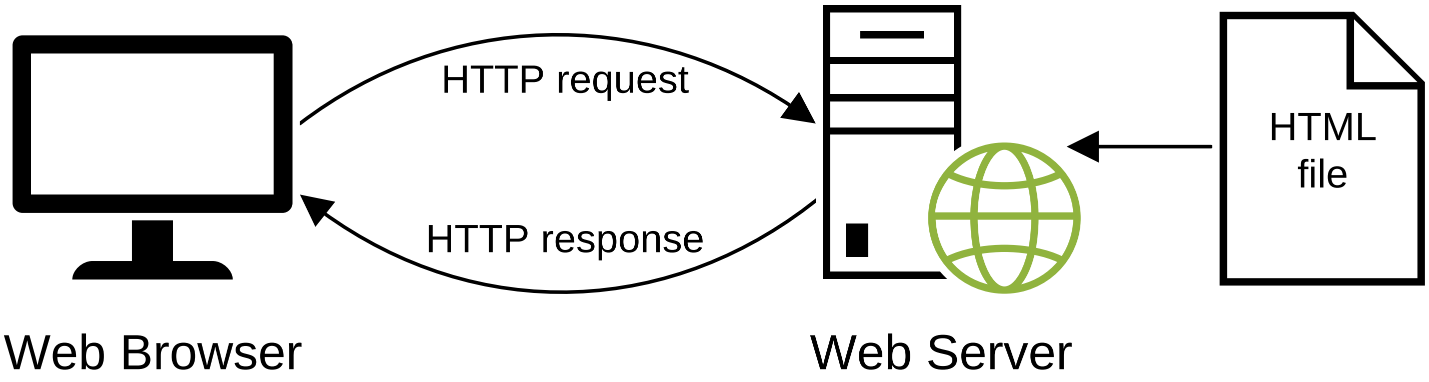
A web app pulls its pages, etc., from a website on a server connected to the Internet (or possibly a smaller network).

A web app can accept user input and send it back to its server for processing (the PHP way), or process it right in the browser (the Javascript way).

A web app depends on its server for saving data persistently.

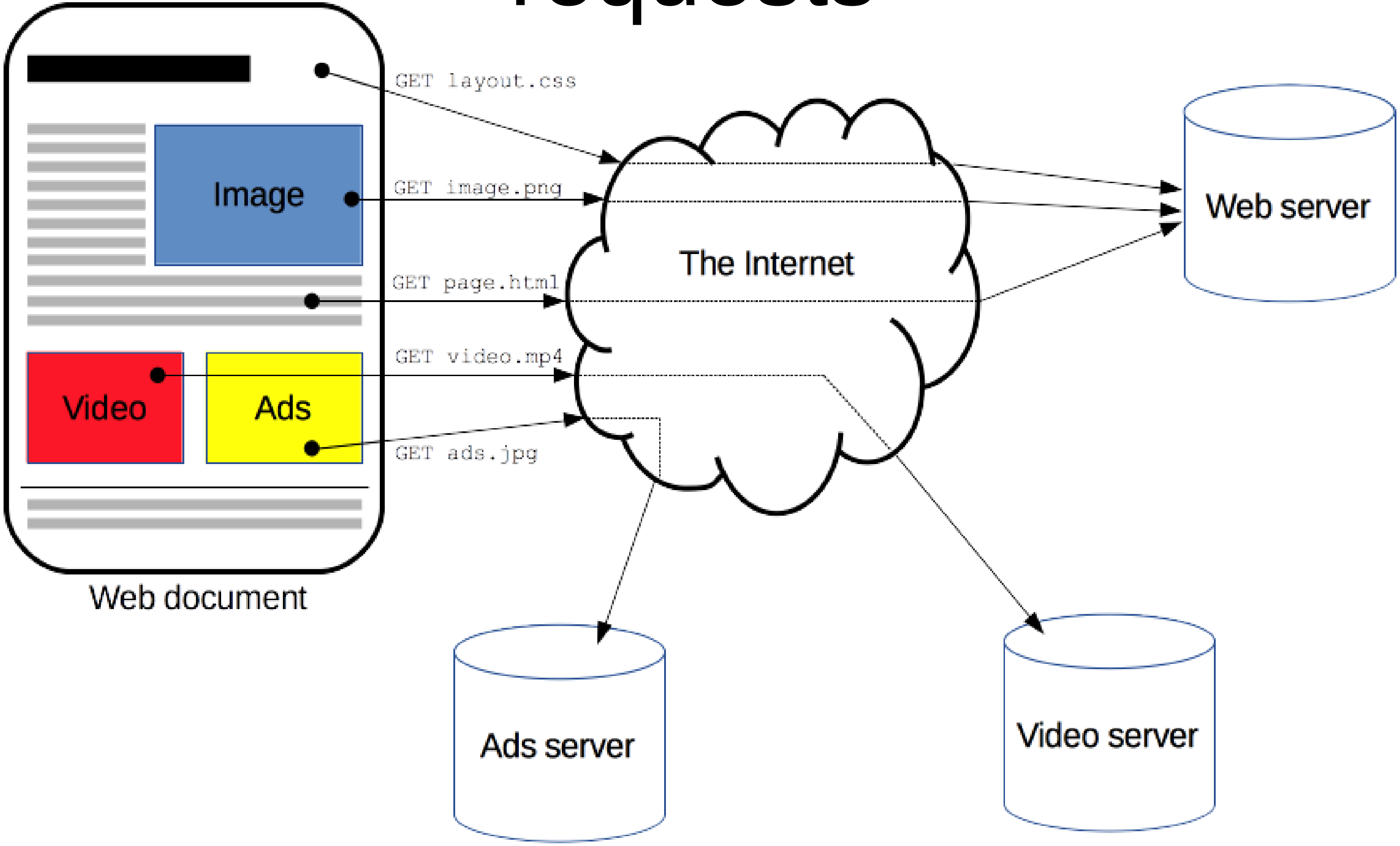
A web app UI is much like an app UI: buttons, forms, text input, etc., but may be slowed down by the network.

# How static web pages are processed



- A *static* web page is a simple HTML file, no PHP or Javascript involved.
- HTTP is the network protocol that delivers HTML and many other data formats to the client web browser, including Javascript code.
- Note the web browser is very likely to be in a mobile device today, so this picture is a little dated.

# One page can involve many HTTP requests



From https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

# A simple HTTP request

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

# A simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

## A simple HTTP request

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

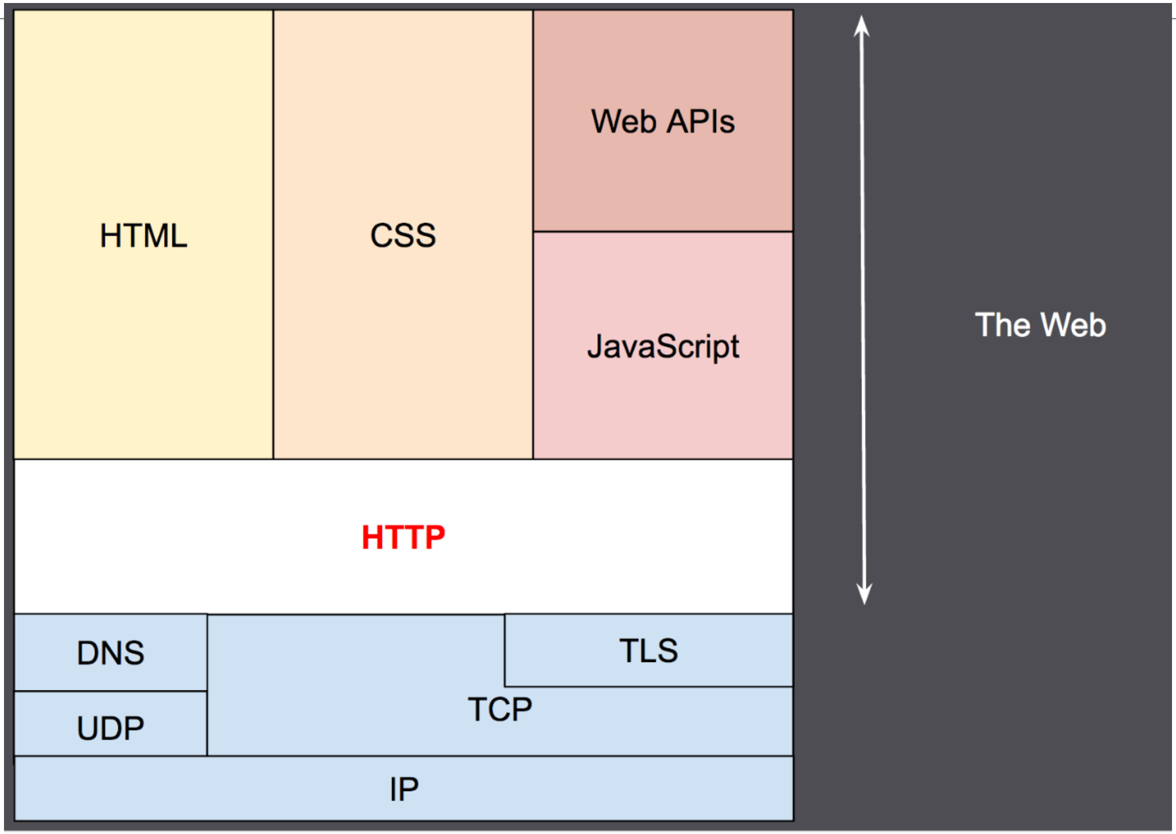
## A simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

- If this HTML text was just read from a file on the server, this is *static HTML.*
- If a program helped generate it, it is *dynamic HTML*.

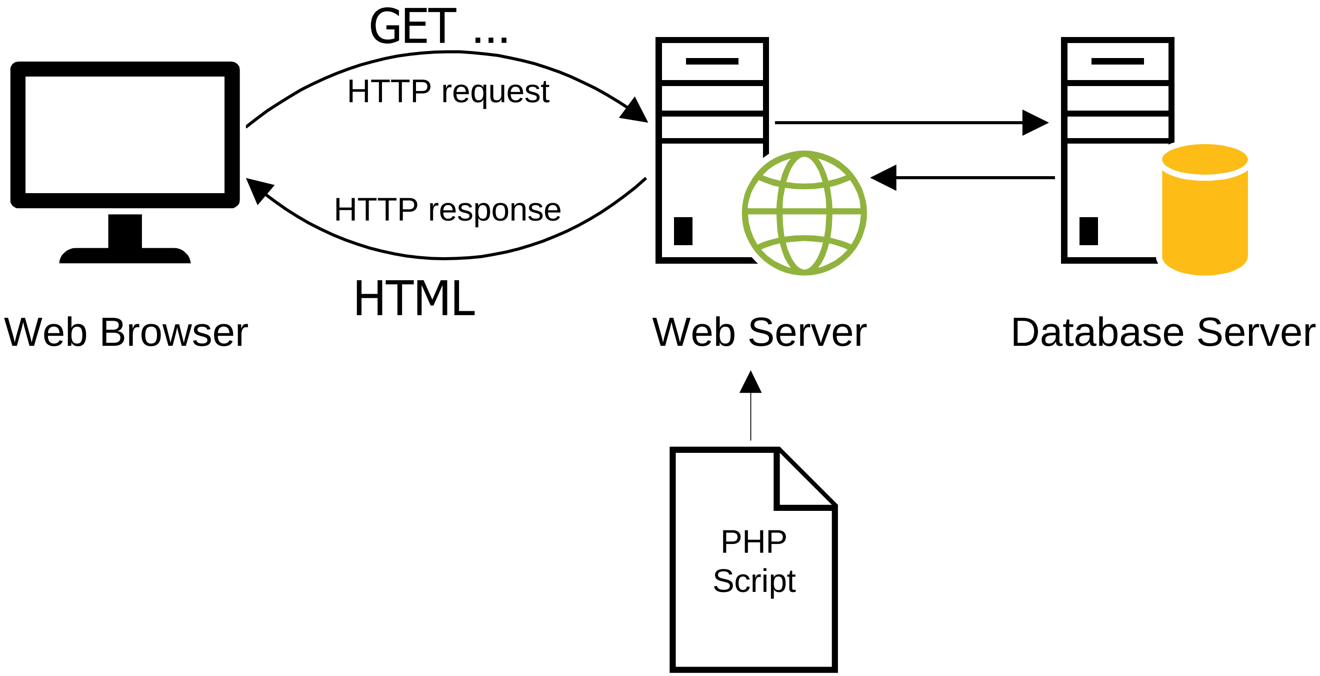
# The response HTML* is carried by HTTP, itself carried by TCP, itself carried by IP



*HTML or CSS or Javascript code, also images, etc.

From https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

# How dynamic web pages are processed with PHP



- The PHP code is not seen by the client, only the *server-generated* HTML.
- Any user click/taps cause another HTTP request to the server for processing.

# Javascript Execution Steps

1. User browses to something.html
2. Server sends HTML to the client, as we have just seen
   - In that HTML: link to URL for Javascript code
3. Browser finds link, does second HTTP request for Javascript (JS) code, again to a server.
   - JS code is loaded into browser memory
4. The JS code can detect load event, start executing to set things up for itself.
   - Can build more HTML elements, for example.
5. Later, when user clicks on something, it can be handled by JS immediately
   - No server involvement unless persistent changes are needed.

# Clients: Desktops and Mobile Devices

**Similarities**

- Both have rectangular hi-res color screens
- Both can accept keyboard input
- Both can accept click/tap selection of buttons, etc.
- Both can accept drag and drag-and-drop user gestures
- Both can accept double-click, but this is almost never used in webapps, just in apps

➢ Based on these similarities, we want to be able to write webapps that run on both using the same codebase.

# Clients: Desktops and Mobile Devices

**Differences**

- Not all desktop screens are touch screens
- Can’t “hover” with touch screens
- Can’t “pinch” or “spread” with non-touch screens
- Other less common touch-screen gestures: long press, flick

- ➢ Using just PHP, we can’t detect any of these, or drag or double-click either, so we will concentrate on click/tap, and text input
- ➢ Using Javascript, we can detect the others…

# Key terms

- HyperText Markup Language (HTML)
- static web page
- HTTP request
- HTTP response.
- dynamic web page
- PHP interpreter
- database server
- render
- round trip

## Web browsers

- Chrome
- Firefox
- Internet Explorer/Edge
- Safari
- Opera

## Web servers

- Apache
- IIS (Microsoft’s Internet Information Services)

## Server-side languages

- PHP
- JSP
- ASP.NET
- Python
- Ruby

## Client-side languages

- Javascript (can also run on server side)

# Highlights in the history of PHP

| Version | Year | Description |
|---|---|---|
| 2 | 1995 | Personal Home Page |
| 3 | 1998 | PHP: Hypertext Processor |
| 4 | 2000 | Introduced the Zend Engine |
| 5 | 2004 | Introduced the Zend Engine II.<br>Improved support for OOP<br>Added the PHP Data Objects extension |
| 6 | -- | PHP6, meant to have native Unicode, abandoned |
| 7 | 12/3/2015 | Zend Engine 3, types for scalar arguments, 64-bit integer support on Windows, … |

Note: we'll use version 7.0 on pe07 (or 7.2 on XAMPP)

## PHP Advantages

- Considered easy to use, focuses on web
- Free, open-source (implemented in C)
- Provides decent performance
    - In particular, efficient with server memory use
- Available on many platforms other than mobile devices

## PHP disadvantages

- PHP strings are not in Unicode, so hard to provide non-English-language websites. (There are library calls for Unicode)
- "Too easy" so "lots of bad PHP exists", including insecure sites.
- Missing large-software features, like Java's package system.
- Not available for execution *inside* mobile platforms.
- Has no equivalent to JDBC, a widely supported generic database access API for Java. Most PHP webapps just use mysql.

**PHP Uses**

- Small sites
- Prototypes for larger sites
- Server-side code for Javascript/Ajax running in web pages
- Server-side code for providing data for programs running in mobile devices
- Note no significant use *inside* mobile devices.

# PHP vs. Javascript

Javascript (JS for short) has become very important for web apps, especially ones that need to work nicely on smartphones.

JS runs in the browser, so is closer to the user than PHP or Java, which run only on the server.

JS can detect the advanced user gestures like pinch/spread.

JS now can run on the server too (with nodejs), providing a single language for both sides.

JS is a completely different language from Java or PHP, themselves in the same family. It does use Java-like syntax for conditionals, loops, etc.

# Javascript and PHP

Websites often use a combination of JS and PHP (or JS and Java) along with HTML and CSS.

- The actions that need fast response to user requests, like game playing or device control, are in JS
- Ancillary support, like UI for changing a password, or finding documentation, are in PHP.
- Server support, like accessing a database, can be in PHP or JS.

The user often doesn’t know what language is in use.

- Either language can do buttons, forms, links, etc.
- But if there’s a bug running around the screen, that’s JS.
  - Or the restaurant menu folds up for you (this also uses advanced CSS).
- On the other hand, you don’t need JS for a photo carousel, so motion by itself doesn’t necessitate JS.

## Database servers

- MySQL and its descendent MariaDB
- Oracle
- DB2
- MS SQL Server
- MongoDB (a No-SQL database)
- Apache HBase (also No-SQL, part of Hadoop project)

# Highlights in the history of MySQL

| Version | Year | Description |
|---|---|---|
| 3.23 | 1995 | The original version of MySQL |
| 4.0 | 2003 | Introduced support for unions. |
| 4.1 | 2004 | Introduced support for subqueries and prepared statements. |
| 5.0 | 2005 | Introduced support for stored procedures, triggers, views, and transactions. |
| 5.1 | 2008 | Introduced support for row-based replication and server log tables. |
| 5.5 | 2010 | Default storage engine now supports referential integrity. |
| 5.6 | 2011 | Introduced support for big data. |

We’ll use MySQL version 5.6+, or MariaDB

# MySQL notes

- MySQL is owned and sponsored by MySQL AB, a for-profit firm.
- In 2008, Sun Microsystems acquired MySQL AB.
- In 2009, Oracle Corporation acquired Sun Microsystems.
- In 2009, many of the original developers of MySQL left MySQL AB and begin working on different forks of the open-source code. One of the most popular of these forks is MariaDB.
- In 2012, tech writers reported that Oracle was holding back MySQL Server test cases and no longer synchronizing their changes with the public source repository. Also, several Linux distributions, Wikipedia and Google, began to replace MySQL with MariaDB.

XAMPP comes with MariaDB now, so we'll be using that on our development systems, and mysql itself on the Linux servers at cs.umb.edu.

# Hello World in PHP

- Source file hello.php:

```
<?php
echo 'Hello World!';
?>
```

- Local execution: use php command
- Note no compilation step! It’s a scripting language.

```
$ php hello.php
Hello World!
$
```

- Normal execution: put hello.php in right place for web server execution, say in c:\xampp\htdocs
- Browse to http://localhost/hello.php, see:

Hello World!

## PHP derives lots of its syntax from C, so does Java and JS… (but not Python)

- C-syntax control structures, increment operators
- But PHP variable names are prefixed with $
- Java-type comments or # line comments

```
// calculate the future value
$future_value = $investment;
for ($i = 1; $i <= $years; $i++) {
  $future_value =
     $future_value + ($future_value *
                  $interest_rate * .01);
}
```

- Could rewrite last assignment, as in Java, with +=:

```
$future_value +=
           $future_value*$interest_rate*.01;
```

# But PHP has differences too

- Good news!  Associative Arrays
- C/Java: only integers can be used as array indexes
- PHP, JS and other scripting languages:
    - ✓ We can use strings as “keys” in array lookup!
    - ✓ This includes Python, which has Dictionaries

PHP Example, pg. 321

```
$tax_rates = array();  // or = [];
$tax_rates[‘NC’] = 7.75;
$tax_rates[‘CA’] = 8.25; // Akk!
…
$payment =
    $price*(1+$tax_rates[$state]);
```

# PHP is loosely, dynamically typed

```
$x = 12;  // an integer
$x = 2.3;  // now a double
$x = null;  // now a null
$x = ‘abc’;  // now a string
```

- $x remembers its type, determined from the value on the rhs of the assignment.

- Good programming practice: types should be kept as static as possible

- Netbeans complains on multiple assignments to a variable, even of the same type values.

## PHP is made for web apps

It maintains global arrays for easy access to HTTP request parameters, HTTP headers, session variables, and cookies.

Don’t worry if this doesn’t make sense yet!

It deallocates or saves away memory data after each request cycle is done, to minimize memory footprint. In other words, it assumes it is sharing the system with many other requestors.

Next time: look at HTML via Duckett’s book…
I’ll send out another Zoom link by UMB email.

# HTML&CSS

design and build websites