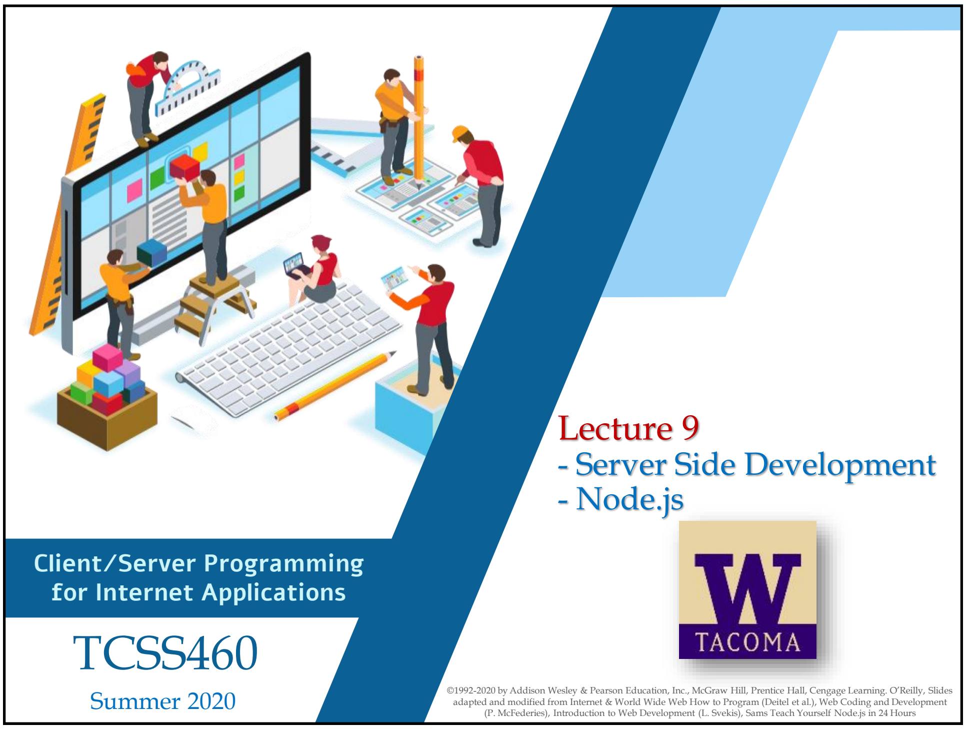
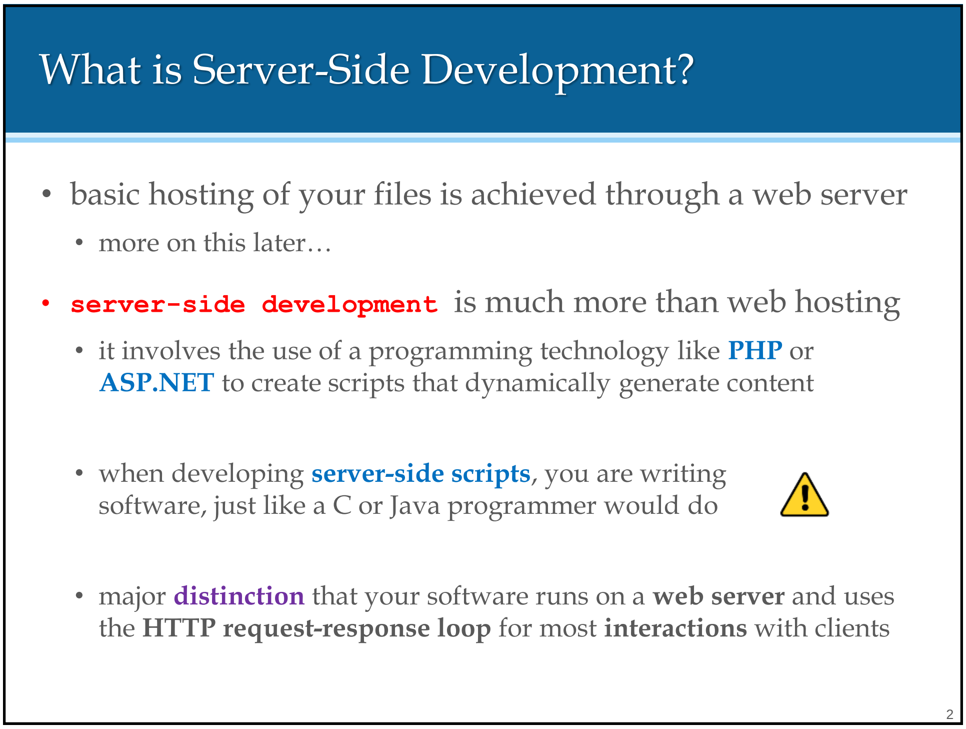


Lecture 9
- Server Side Development
- Node.js

Client/Server Programming for Internet Applications

TCSS460

Summer 2020



## What is Server-Side Development?

- basic hosting of your files is achieved through a web server
  - more on this later…
- `server-side development` is much more than web hosting
  - it involves the use of a programming technology like **PHP** or **ASP.NET** to create scripts that dynamically generate content
  - when developing **server-side scripts**, you are writing software, just like a C or Java programmer would do ⚠
  - major **distinction** that your software runs on a **web server** and uses the **HTTP request-response loop** for most **interactions** with clients

2

# Comparing Client and Server Scripts

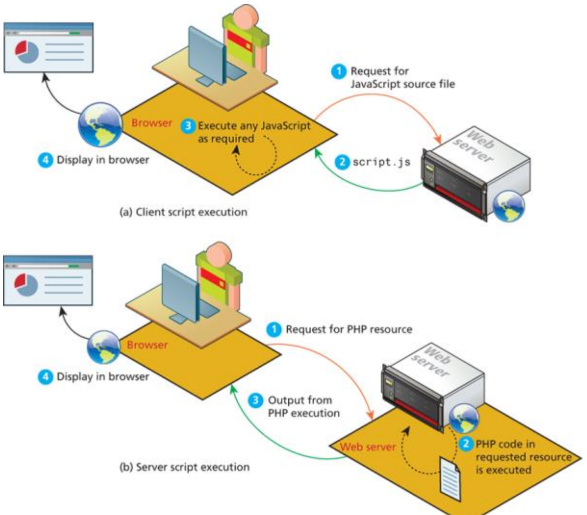
- earlier, we studied JavaScript, a client-side web programming language (a scripting language)
- fundamental difference between client and server scripts
  - client-side script: code is executed by the client's browser
  - server-side script: code is executed on the web server
    - *e.g. server sends JavaScript but no guarantee it will even execute*
    - **server-side** source code **remains hidden** from client as it is processed on the server
    - client never get to see the code, just the HTML output from the script

3

# Comparing Client and Server Scripts (cont'd)

Comparison of (a) client script execution and (b) server script execution

- important considerations
  - location of the script also impacts what resources it can access
  - server scripts **cannot** manipulate the **HTML** or **DOM** of a page in the client browser as the case with client scripts
  - a **server script** can **access resources** on the web server whereas the client cannot


1 Request for JavaScript source file
Browser
3 Execute any JavaScript as required
2 script.js
Web server
4 Display in browser
(a) Client script execution
1 Request for PHP resource
Browser
Web server
4 Display in browser
3 Output from PHP execution
Web server
2 PHP code in requested resource is executed
(b) Server script execution

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

4

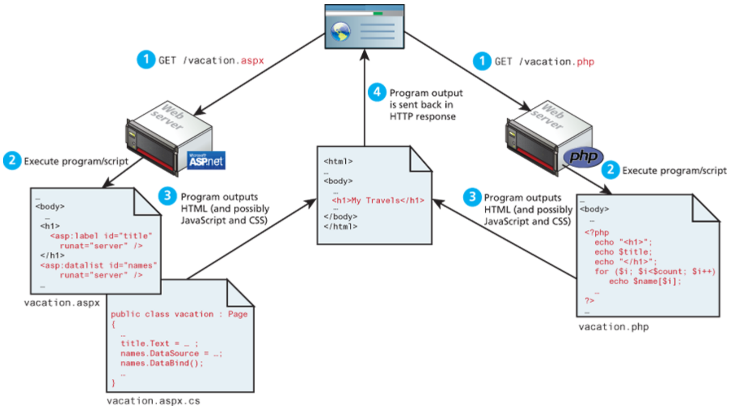
# Comparing Server-Side Technologies

| Technology | Description |
|---|---|
| **ASP (Active Server Pages)** | • was Microsoft's first server-side technology (also called ASP Classic)<br>• ASP programming code is interpreted at run time |
| **ASP.NET** | • part of Microsoft's .NET Framework (can use any .NET language)<br>• often used in larger corporate web application systems<br>• uses special markup called web server controls<br>• recent extension called ASP.NET MVC |
| **JSP (Java Server Pages)** | • uses Java as its programming language<br>• used in large enterprise web systems and is integrated into the J2EE environment |
| **Node.js** | • recent server environment that uses JavaScript on the server side<br>• node.js has its own web server software, thus eliminating the need for Apache, IIS, or some other web server software |
| **Perl** | • was the language typically used for early server-side web development |
| **PHP** | • dynamically typed language that can be embedded directly within HTML<br>• supports most common object-oriented features such as classes & inheritance |
| **Python** | • an object-oriented programming language has many uses<br>• used in a variety of web development frameworks (e.g. Django & Pyramid) |
| **Ruby on Rails** | • a web development framework that uses the Ruby programming language |

5

# Comparing Server-Side Technologies (cont'd)

- all server-side technologies share one goal in common: using programming logic, they generate HTML and possibly CSS and JavaScript on the server and send it back to the requesting browser


1 GET /vacation.aspx
1 GET /vacation.php
4 Program output is sent back in HTTP response
Web server
ASP.net
Web server
php
2 Execute program/script
2 Execute program/script
3 Program outputs HTML (and possibly JavaScript and CSS)
3 Program outputs HTML (and possibly JavaScript and CSS)
vacation.aspx
vacation.aspx.cs
vacation.php

```
…
<body>
  …
  <h1>
    <asp:label id="title"
       runat="server" />
  </h1>
  <asp:datalist id="names"
       runat="server" />
  …
```

```
public class vacation : Page
{
  …
  title.Text = … ;
  names.DataSource = …;
  names.DataBind();
  …
}
```

```
<html>
…
<body>
  …
  <h1>My Travels</h1>
  …
</body>
</html>
```

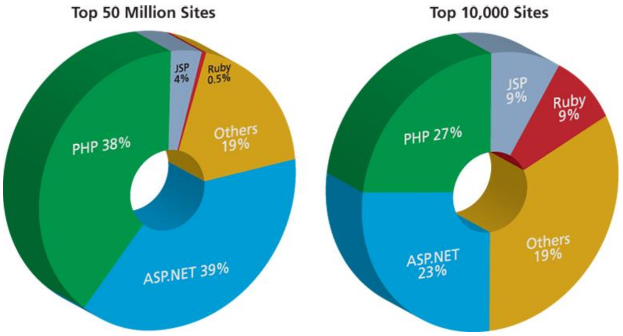
```
…
<body>
  …
  <?php
    echo "<h1>";
    echo $title;
    echo "</h1>";
    for ($i; $i<$count; $i++)
       echo $name[$i];
    …
  ?>
  …
```

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

6

## Comparing Server-Side Technologies (cont'd)

- **ASP.NET** (combined with ASP and ASP.NET MVC) and **PHP** appear to have the largest market share
  - PHP is the most commonly used web development technology, and will be the technology we will use in this module



**Market share of web development environments** *(data courtesy of **BuiltWith.com**)*

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

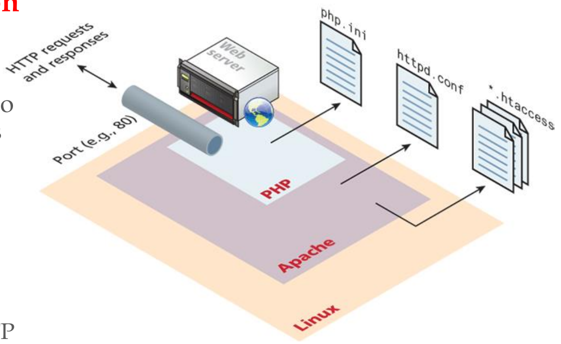
## A Web Server's Responsibilities

- **server** is responsible for answering all client requests
  - once a web server is configured and its IP address is associated through a DNS server, it can then start listening for and answering HTTP requests
- a **web server** has many responsibilities beyond responding to requests for HTML files
  - handling HTTP connections,
  - responding to requests for static and dynamic resources,
  - managing permissions and access for certain resources,
  - encrypting and compressing data,
  - managing multiple domains and URLs,
  - managing database connections,
  - cookies, and state, and
  - uploading and managing files.

# Apache and Linux

- Apache runs as a **daemon** on the server
  - a **daemon** is an executing instance of a program (also called a process) that runs in the background
  - a **daemon** waits for a specific event that will activate it
  - Apache daemon: **httpd**
    - it waits for incoming HTTP requests.
    - when a request arrives, Apache then uses **modules** to determine how to respond to the request

**Linux, Apache, and PHP together**



in Apache, a **module** is a compiled extension (usually written in the C programming language) to Apache that helps it handle requests

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

# PHP

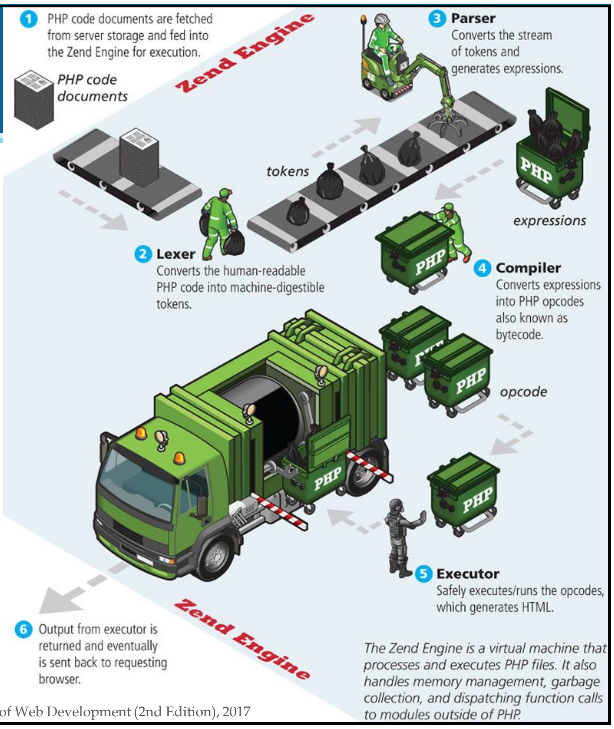
Let's briefly examine PHP

## What is PHP?

- **PHP** itself is written in the C programming language and is composed of three main modules:
  - **PHP core**: defines the main features of the PHP environment
    - e.g. variable handling, arrays, strings, classes, math, and other core features
  - **extension layer**: defines functions for interacting with services outside of PHP
    - e.g. libraries for MySQL (and other databases), FTP, SOAP web services, XML processing, etc.
  - **Zend Engine**: handles the reading of a requested PHP file, compiling it, and executing it
    - Zend Engine is a **virtual machine**

## PHP Internals

- to run the PHP examples, you will need specialized software that recognizes PHP files and execute them properly
- when you develop PHP code in this module, your local machine may likely be hosting both the browser software and web server software



1 PHP code documents are fetched from server storage and fed into the Zend Engine for execution.

2 **Lexer** Converts the human-readable PHP code into machine-digestible tokens.

3 **Parser** Converts the stream of tokens and generates expressions.

4 **Compiler** Converts expressions into PHP opcodes also known as bytecode.

5 **Executor** Safely executes/runs the opcodes, which generates HTML.

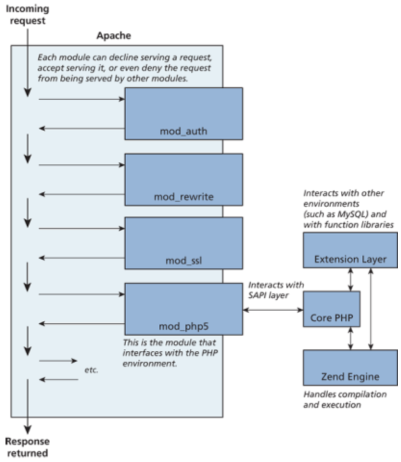
6 Output from executor is returned and eventually is sent back to requesting browser.

*The Zend Engine is a virtual machine that processes and executes PHP files. It also handles memory management, garbage collection, and dispatching function calls to modules outside of PHP.*

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

## Apache Modules and PHP

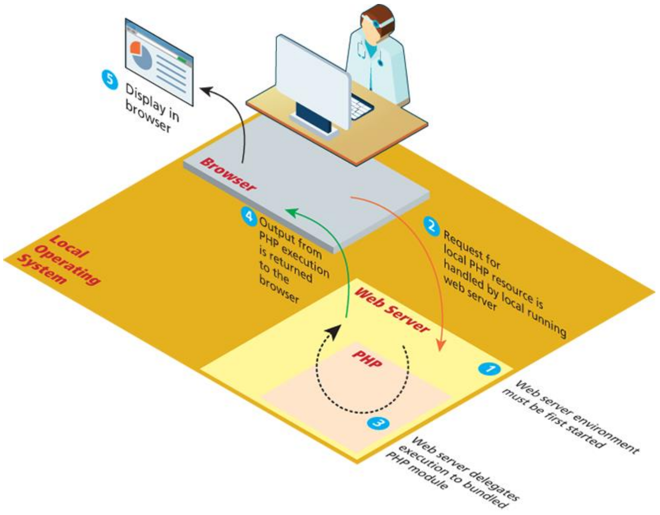
- each module is given an opportunity to handle some aspect of the request
  - some modules handle
    - authorization,
    - URL rewriting
    - specific extensions
- PHP is usually installed as an **Apache module**
  - can also be installed as CGI binary
- PHP module **mod_php5** is sometimes referred to as the **SAPI** layer
  - **Server Application Programming Interface**
    - **SAPI** handles the interaction between the PHP environment and web server environment


Incoming request
Apache
Each module can decline serving a request, accept serving it, or even deny the request from being served by other modules.
mod_auth
mod_rewrite
mod_ssl
mod_php5
This is the module that interfaces with the PHP environment.
etc.
Response returned
Interacts with SAPI layer
Interacts with other environments (such as MySQL) and with function libraries
Extension Layer
Core PHP
Zend Engine
Handles compilation and execution

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

## Can I Run PHP Locally?

Hosting a web server locally


5 Display in browser
Browser
Local Operating System
4 Output from PHP execution is returned to the browser
2 Request for local PHP resource is handled by local running web server
Web Server
PHP
1 Web server environment must be first started
3 Web server delegates execution to bundled PHP module

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

## Running PHP → Web Software Stacks

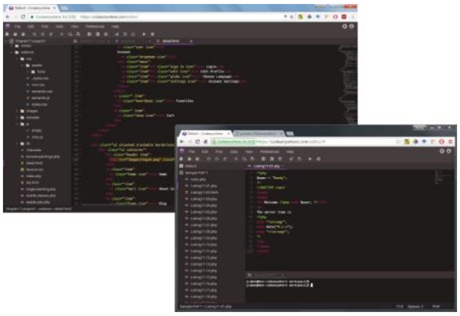
- one of the true benefits of the **LAMP** web development stack is that it can run on almost any computer platform
  - **AMP** part can run on most OS (including Windows and Mac OS)
    - **A** → Apache **M** → MySQL **P** → PHP (also used for Python)
- there are many different ways to install web software stacks
  - easiest & quickest: use all-in-one management software that bundles popular tools together
    - easyPHP (www.easyphp.org)
    - XAMPP (www.apachefriends.org)
    - MAMP for Mac (www.mamp.info)

## Running PHP → cloud-based

- an alternative to running PHP locally, you can use **cloud-based integrated development environment (IDE)**
  - examples
    - cloud9 (c9.io) → by Amazon
    - codeanywhere (www.codeanywhere.com)

AWS Cloud9

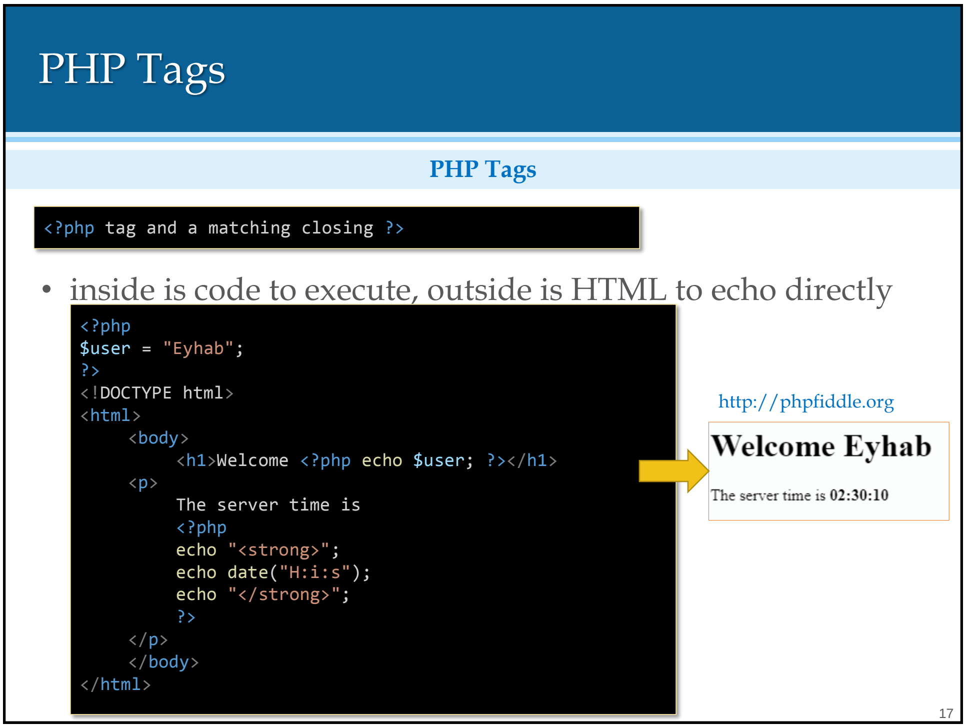
codeanywhere



to interpret a file and echo its output directly to the console

```
php example1.php
```

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017



# PHP Tags

## PHP Tags

```
<?php tag and a matching closing ?>
```

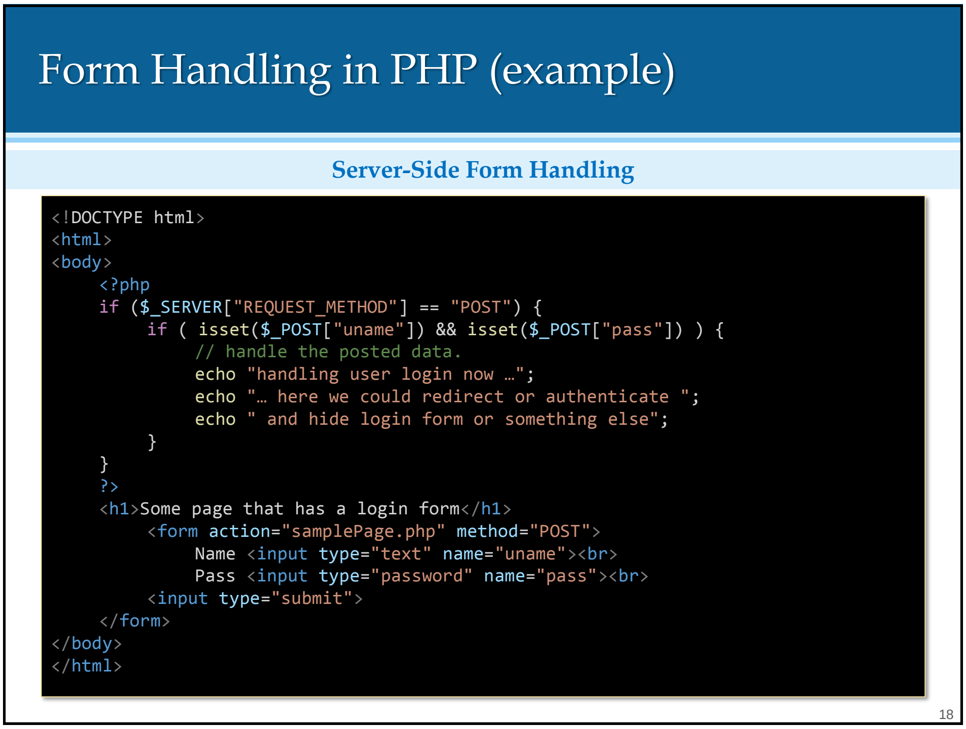
- inside is code to execute, outside is HTML to echo directly

```
<?php
$user = "Eyhab";
?>
<!DOCTYPE html>
<html>
     <body>
          <h1>Welcome <?php echo $user; ?></h1>
     <p>
          The server time is
          <?php
          echo "<strong>";
          echo date("H:i:s");
          echo "</strong>";
          ?>
     </p>
     </body>
</html>
```

http://phpfiddle.org

**Welcome Eyhab**

The server time is **02:30:10**



# Form Handling in PHP (example)

## Server-Side Form Handling

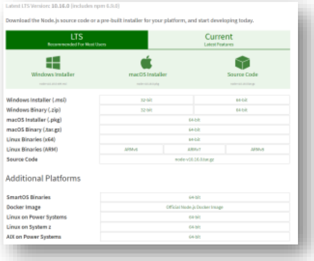
```
<!DOCTYPE html>
<html>
<body>
     <?php
     if ($_SERVER["REQUEST_METHOD"] == "POST") {
          if ( isset($_POST["uname"]) && isset($_POST["pass"]) ) {
               // handle the posted data.
               echo "handling user login now …";
               echo "… here we could redirect or authenticate ";
               echo " and hide login form or something else";
          }
     }
     ?>
     <h1>Some page that has a login form</h1>
          <form action="samplePage.php" method="POST">
               Name <input type="text" name="uname"><br>
               Pass <input type="password" name="pass"><br>
          <input type="submit">
     </form>
</body>
</html>
```

# Node.js

Let’s examine Node.js

## What is Node.js?

- **Node.js** is a programming platform for running primarily server-side code
  - can create small scripts to do tasks on a file system or large-scale web applications
  - ideal for **data intensive applications**
  - based on JavaScript
- official page: http://nodejs.org



Latest LTS Version: **10.16.0** (includes npm 6.9.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

| | | | |
|---|---|---|---|
| Windows Installer (.msi) | 32-bit | 64-bit | |
| Windows Binary (.zip) | 32-bit | 64-bit | |
| macOS Installer (.pkg) | 64-bit | | |
| macOS Binary (.tar.gz) | 64-bit | | |
| Linux Binaries (x64) | 64-bit | | |
| Linux Binaries (ARM) | ARMv6 | ARMv7 | ARMv8 |
| Source Code | node-v10.16.0.tar.gz | | |

Additional Platforms

| | |
|---|---|
| SmartOS Binaries | 64-bit |
| Docker Image | Official Node.js Docker Image |
| Linux on Power Systems | 64-bit |
| Linux on System z | 64-bit |
| AIX on Power Systems | 64-bit |

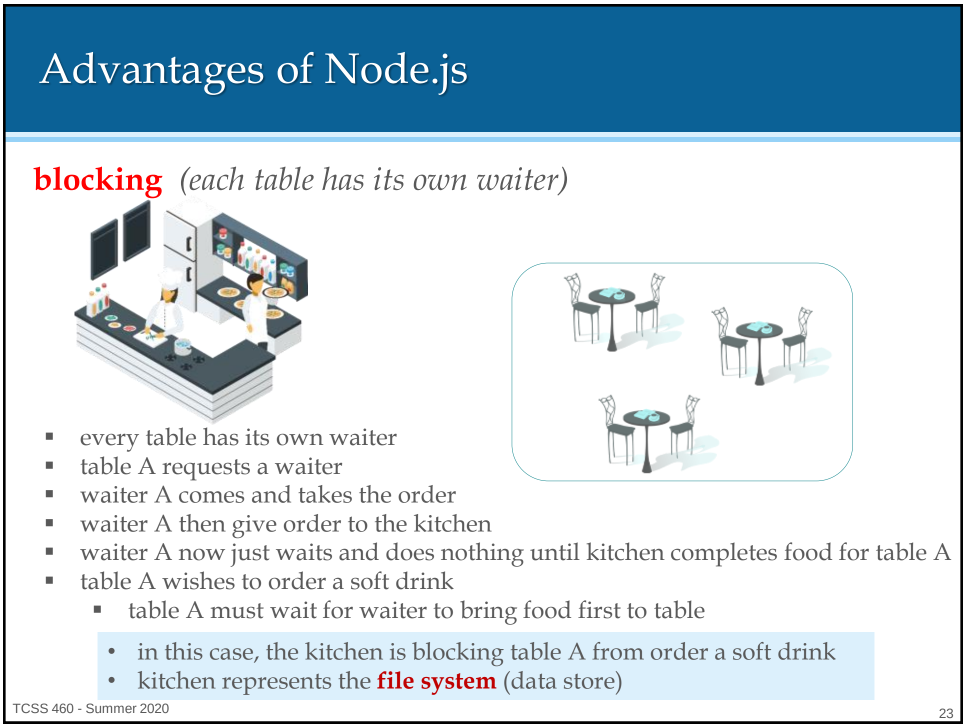
Download for Windows (x64)

## What is Node.js? (cont'd)

- introduced in 2009 by Ryan Dahl
  - around the same time, Apple and Google were investing heavily on improving browser technologies
    - Microsoft later joined with the chakra (JavaScript Engine) in 2015
- why browsers?
  - most services are offered today through browsers
    - e.g. Gmail, Google docs, etc.
- Google engineers created **V8**
  - JavaScript engine for Google's Chrome web browser
  - optimized software that was designed for the web
  - encourage adoption of V8 → it was open sourced under BSD license

## What is Node.js? (cont'd)

- Ryan Dahl used Google's V8 to create a **server-side JavaScript platform**
  - V8 was very well-optimized → runs very fast
  - V8 was designed for the web → HTTP and REST
  - JavaScript is a well-known to developers
- core: **event-driven server-side JavaScript environment**
  - can create server-side applications using JavaScript using the same approach for server-side scripting languages such as PHP

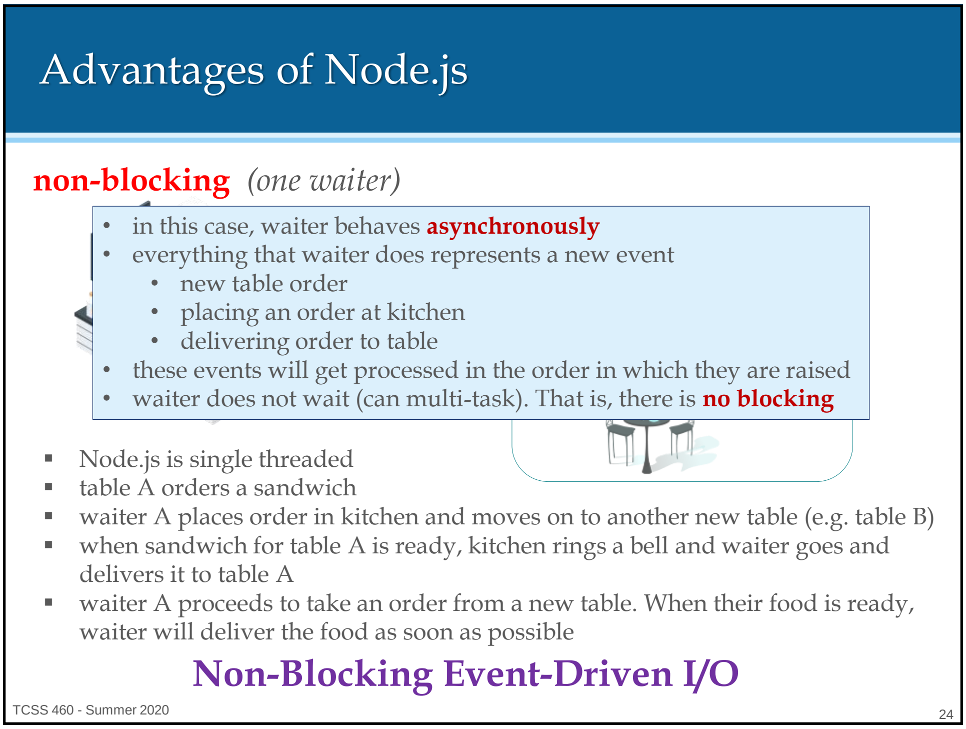


## Advantages of Node.js

**blocking** *(each table has its own waiter)*

- every table has its own waiter
- table A requests a waiter
- waiter A comes and takes the order
- waiter A then give order to the kitchen
- waiter A now just waits and does nothing until kitchen completes food for table A
- table A wishes to order a soft drink
  - table A must wait for waiter to bring food first to table

- in this case, the kitchen is blocking table A from order a soft drink
- kitchen represents the **file system** (data store)



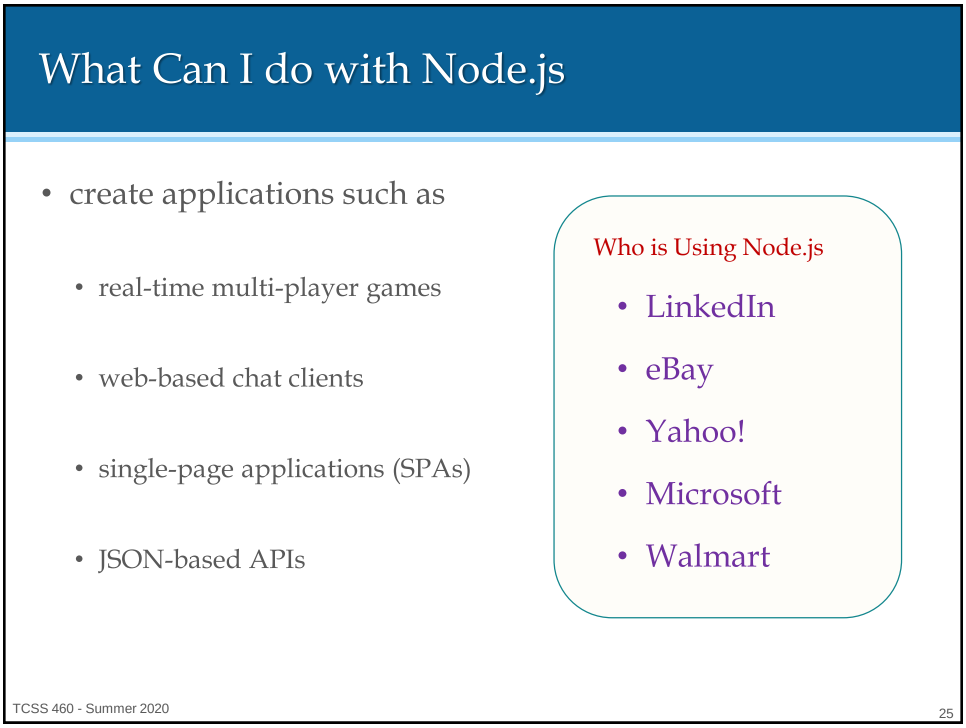
## Advantages of Node.js

**non-blocking** *(one waiter)*

- in this case, waiter behaves **asynchronously**
- everything that waiter does represents a new event
  - new table order
  - placing an order at kitchen
  - delivering order to table
- these events will get processed in the order in which they are raised
- waiter does not wait (can multi-task). That is, there is **no blocking**

- Node.js is single threaded
- table A orders a sandwich
- waiter A places order in kitchen and moves on to another new table (e.g. table B)
- when sandwich for table A is ready, kitchen rings a bell and waiter goes and delivers it to table A
- waiter A proceeds to take an order from a new table. When their food is ready, waiter will deliver the food as soon as possible

**Non-Blocking Event-Driven I/O**

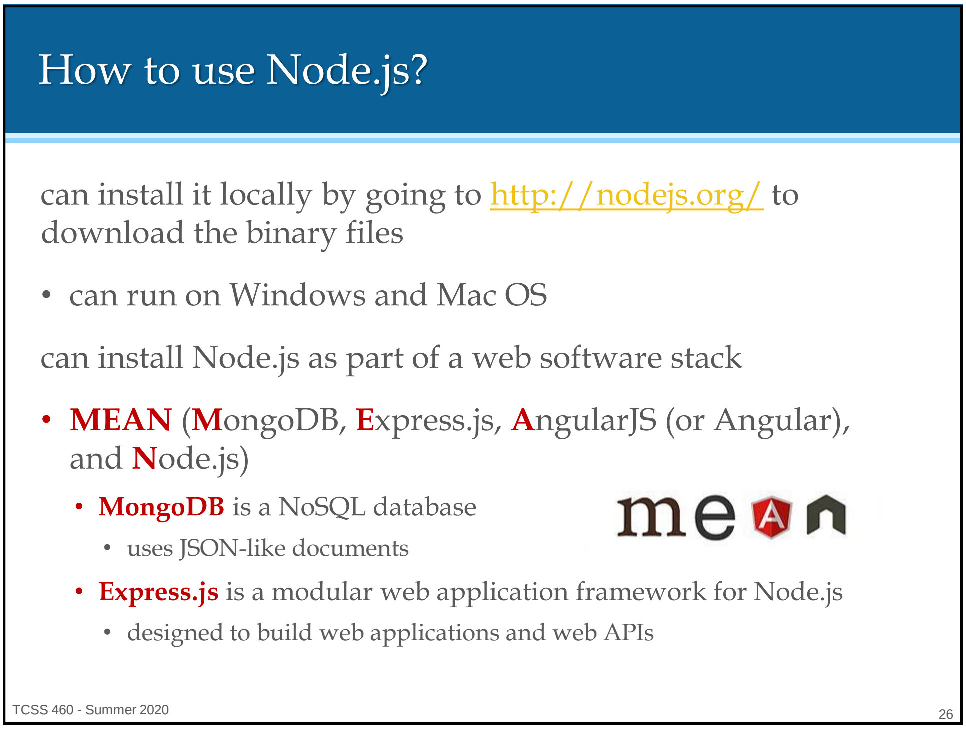


## What Can I do with Node.js

- create applications such as
  - real-time multi-player games
  - web-based chat clients
  - single-page applications (SPAs)
  - JSON-based APIs

Who is Using Node.js

- LinkedIn
- eBay
- Yahoo!
- Microsoft
- Walmart



## How to use Node.js?

can install it locally by going to http://nodejs.org/ to download the binary files

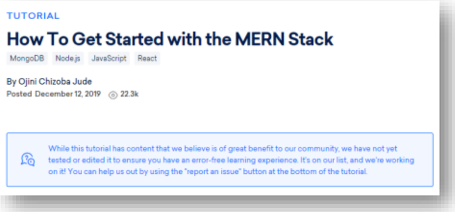
- can run on Windows and Mac OS

can install Node.js as part of a web software stack

- **MEAN** (**M**ongoDB, **E**xpress.js, **A**ngularJS (or Angular), and **N**ode.js)
  - **MongoDB** is a NoSQL database
    - uses JSON-like documents
  - **Express.js** is a modular web application framework for Node.js
    - designed to build web applications and web APIs

## How to use Node.js? (cont'd)

- **MERN**
  - **M**ongoDB
  - **E**xpress.js
  - **React** and
  - **N**ode.js



TUTORIAL

**How To Get Started with the MERN Stack**

MongoDB Node.js JavaScript React

By Ojini Chizoba Jude
Posted December 12, 2019 22.3k

While this tutorial has content that we believe is of great benefit to our community, we have not yet tested or edited it to ensure you have an error-free learning experience. It's on our list, and we're working on it! You can help us out by using the "report an issue" button at the bottom of the tutorial.

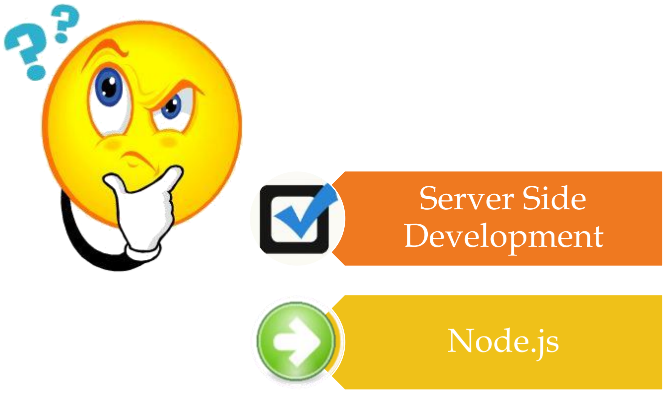
- **MEVN**
  - **M**ongoDB
  - **E**xpress.js
  - **Vue.js** and
  - **N**ode.js

Full-Stack Web Development with Vue.js and Node

Aneeta Sharma

December 5, 2018
366 pages

## Module Topics



Server Side Development

Node.js