

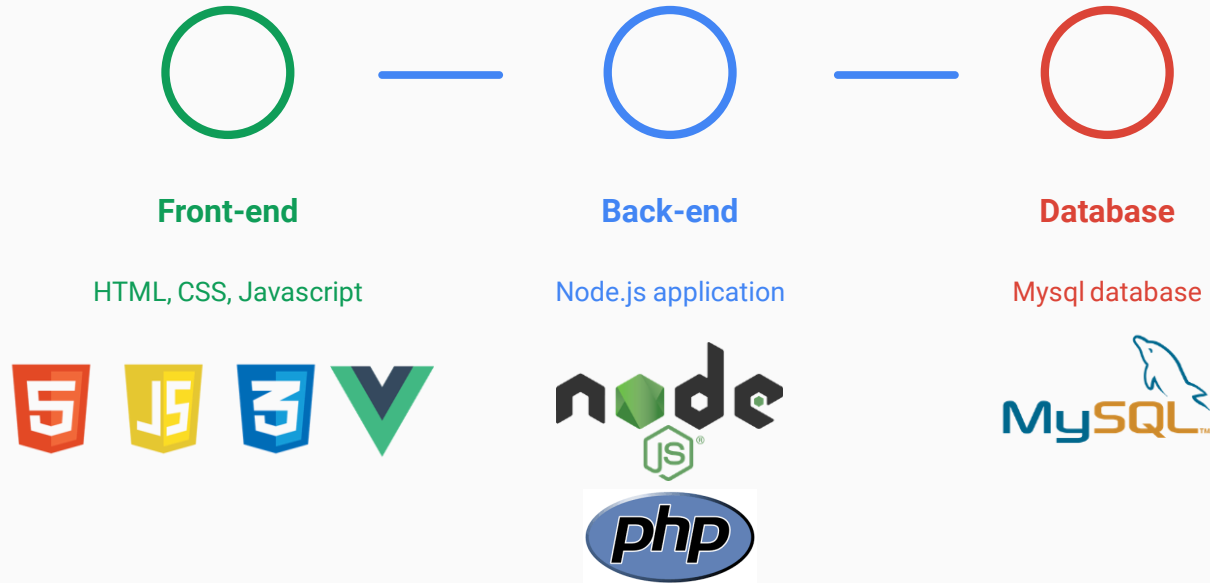
CSC336 - Web Technologies

Lecture # 2

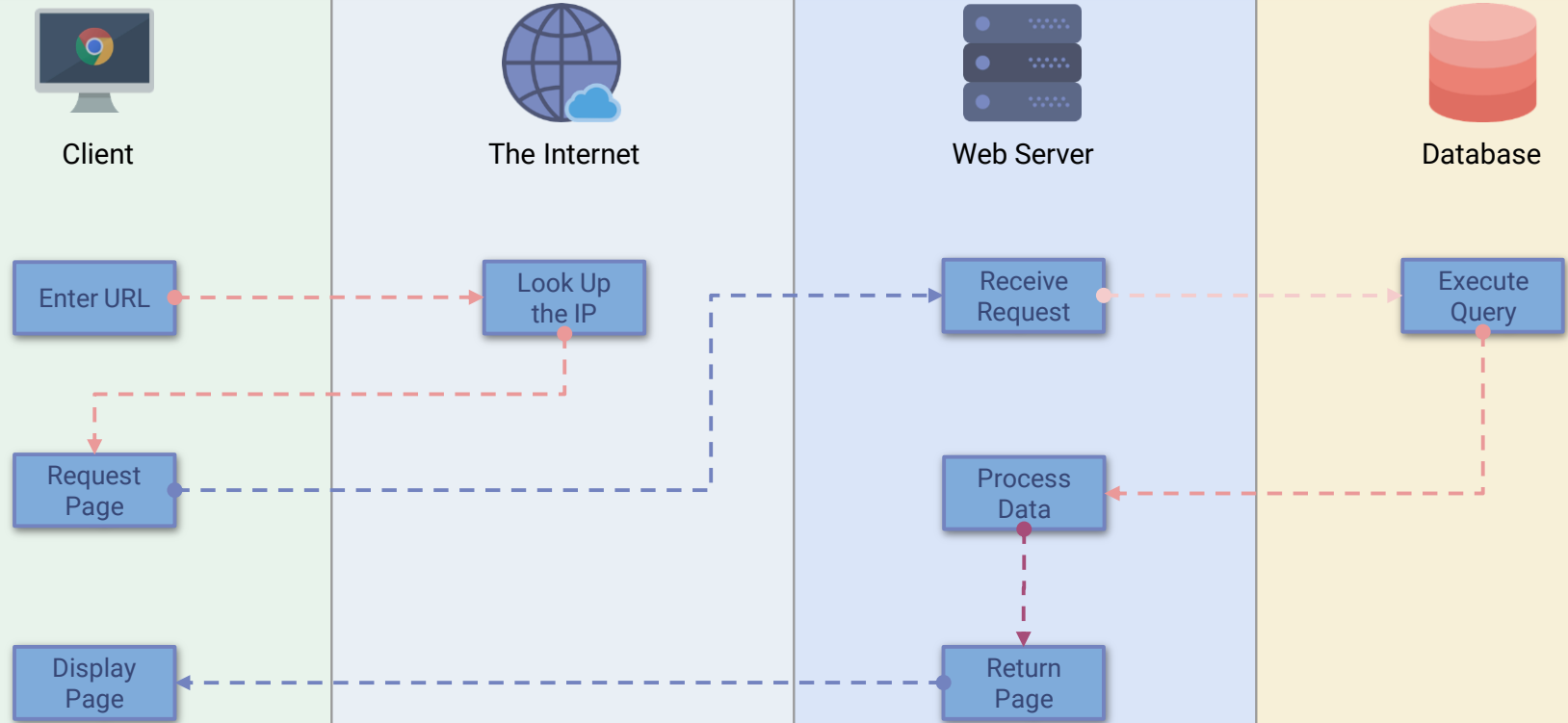


Lecture 1 Recap

Three-tier Architecture in Web



Client-Server communication



HTTP Request and Response

Request Message

```
GET /sample/example?param=1 HTTP/1.1
Host: www.example.com
Content-Type: application/json
Cache-Control: no-cache
```

Response Message

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 138

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

HTML

Hypertext Markup Language

HTML

Hypertext Markup Language (HTML) is the standard **markup language** for documents designed to be displayed in a **web browser**.^[1]



In computer text processing, a **markup language** is a system for **annotating** a **document** in a way that is **syntactically distinguishable** from the text.^[2]



[1] <https://en.wikipedia.org/wiki/HTML>

[2] https://en.wikipedia.org/wiki/Markup_language

index.html

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```



Page title

Search Google or type a URL

This is a heading

This is a paragraph.

This is another paragraph.

!doctype

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

```
1. <!doctype html>
```

HTML Elements

An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:

```
1. <tagname>Content goes here</tagname>
```

The HTML element is everything from the **start** tag to the **end** tag.

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"

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

```
1. 
```

<head>

- The <head> element is a container for metadata. HTML metadata is data about the HTML document.
- Metadata is not displayed.
- Metadata typically define the document title, character set, styles, scripts, and other meta information.

```
1. <head>
2.     <title>Page Title</title>
3.     <meta charset="UTF-8">
4.     <meta name="description" content="Web development">
5.     <meta name="keywords" content="HTML, CSS, JavaScript">
6.     <meta name="author" content="John Doe">
7. </head>
```

<body>

- The <body> element defines the document body.
- Contents inside body tag are displayed

```
1. <body>
2.     <h1>This is a Heading</h1>
3.     <p>This is a paragraph.</p>
4.     <p>This is another paragraph.</p>
5. </body>
```

Common tags

Headings

- Headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading. `<h6>` defines the least important heading.
- **Note:** Search engines use the headings to index the structure and content of your web pages.

1. `<h1>Heading 1</h1>`

2. `<h2>Heading 2</h2>`

3. `<h3>Heading 3</h3>`

4. `<h4>Heading 4</h4>`

5. `<h5>Heading 5</h5>`

6. `<h6>Heading 6</h6>`

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Paragraphs

- The HTML `<p>` element defines a paragraph:
- With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code. The browser will remove any extra spaces and extra lines when the page is displayed.

```
1. <p>This is a paragraph.</p>
2. <p>This is a paragraph.</p>
3. <p>
4.   This is first line.

5.   And this is  second line.
6. </p>
```

This is a paragraph.

This is a paragraph.

This is first line. And this is second line.

Formatting

Formatting elements were designed to display special types of text:

- `` - Bold text
- `` - Important text
- `<i>` - Italic text
- `` - Emphasized text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

1. ``This text is bold``
2. ``This text is strong``
3. `<i>`This text is italic`</i>`
4. ``This text is emphasized``
5. `<p>`This is `_{`subscript`}` and `^{`superscript`}``</p>`

This text is bold

This text is strong

This text is italic

This text is emphasized

This is _{subscript} and ^{superscript}

Formatting

Formatting elements were designed to display special types of text:

- `<small>` - Small text
- `<mark>` - Marked text
- `` - Deleted text
- `<ins>` - Inserted text

1. `<p>HTML <small>Small</small> Formatting</p>`
2. `<p>HTML <mark>Marked</mark> Formatting</p>`
3. `<p>My favorite color is blue red.</p>`
4. `<p>My favorite <ins>color</ins> is red.</p>`

HTML Small Formatting

HTML **Marked** Formatting

My favorite color is ~~blue~~ red.

My favorite color is red.

HTML Comments

- Comment tags are used to insert comments in the HTML source code.

1. `<!-- This is a comment -->`
2. `<p>This is a paragraph.</p>`
3. `<!-- Remember to add more information here -->`
4. `<!-- Do not display this image at the moment`
5. ``
6. `-->`

This is a paragraph.

Links

- The `href` attribute specifies the destination address of the link.
- The *link text (content of the tag)* is the visible part.
- Clicking on the link text will send you to the specified address.

```
1. <a href="https://www.w3schools.com/html/">Visit our  
   HTML tutorial</a>
```

[Visit our HTML tutorial](https://www.w3schools.com/html/)

Images

- In HTML, images are defined with the `` tag.
- The `` tag is empty, it contains attributes only, and does not have a closing tag.
- The `src` attribute specifies the URL (web address) of the image


```
1. 
```



Images

- The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).
- The value of the `alt` attribute should describe the image

```
1. 
```

 Flowers in Chania

Tables

- An HTML table is defined with the `<table>` tag.
- Each table row is defined with the `<tr>` tag. A table header is defined with the `<th>` tag. By default, table headings are bold and centered. A table data/cell is defined with the `<td>` tag.

```
1. <table>
2.   <tr>
3.     <th>Firstname</th>
4.     <th>Lastname</th>
5.     <th>Age</th>
6.   </tr>
7.   <tr>
8.     <td>Jill</td>
9.     <td>Smith</td>
10.    <td>50</td>
11.  </tr>
12.  <tr>
13.    <td>Eve</td>
14.    <td>Jackson</td>
15.    <td>94</td>
16.  </tr>
17.</table>
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

Lists

- An unordered list starts with the `` tag. Each list item starts with the `` tag.
- An ordered list starts with the `` tag. Each list item starts with the `` tag.

```
1. <ul>
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

- Coffee
- Tea
- Milk

```
1. <ol>
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

1. Coffee
2. Tea
3. Milk

Block vs Inline

- Every HTML element has a default display value depending on what type of element it is. The two display values are: **block** and **inline**.
- A **block-level** element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
- An **inline** element does not start on a new line and only takes up as much width as necessary.

1. `<div>Hello World</div>`
2. `Hello World`

Hello world

Hello World

Block vs Inline

Block tags

```
<address><article><aside><blockquote>  
<canvas><dd><div><dl><dt><fieldset>  
<figcaption><figure><footer><form><h1>-  
<h6><header><hr><li><main><nav><noscript><o  
l>  
<p><pre><section><table><tfoot><ul><video>
```

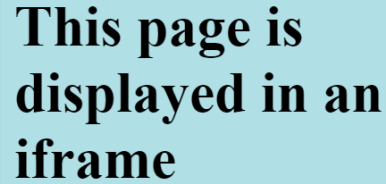
Inline tags

```
<a><abbr><acronym><b><bdo><big><br><button>  
<cite><code><dfn><em><i><img><input><kbd>  
<label><map><object><output><q><samp>  
<script><select><small><span><strong><sub>  
<sup><textarea><time><tt><var>
```

Iframes

- An iframe is used to display a web page within a web page.
- The `src` attribute specifies the URL (web address) of the inline frame page.
- Use the `height` and `width` attributes to specify the size of the iframe.

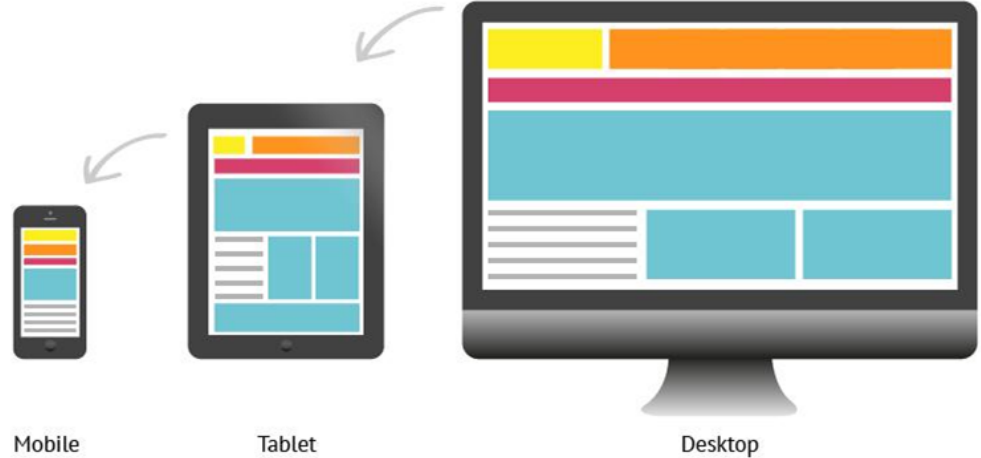
```
1. <iframe src="demo_iframe.htm" height="200"  
width="300"></iframe>
```



**This page is
displayed in an
iframe**

Responsive Web Design

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):



Responsive

- This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.

```
1. <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
```

Without viewport tag



With viewport tag



Forms

Form

- The `action` attribute defines the action to be performed when the form is submitted.
- The `method` attribute specifies the HTTP method (GET or POST) to be used when submitting the form data

```
1. <form action="/action_page.php" method="get">
2.   First name:<br>
3.   <input type="text" name="firstname"><br>
4.   Last name:<br>
5.   <input type="text" name="lastname"><br>
6.   <input type="submit" value="Submit">
7. </form>
```

First name:

Last name:

Input types

- The `value` attribute specifies the initial value for an input field
- The `readonly` attribute specifies that the input field is read only (cannot be changed)
- The `disabled` attribute specifies that the input field input field is unusable and its value will not be sent when submitting the form
- The `size` attribute specifies the size (in characters) for the input field
- The `maxlength` attribute specifies the maximum allowed length for the input field

```
<input type="text">
```

```
<input type="password">
```

```
<input type="checkbox">
```

```
<input type="radio">
```

```
<input type="file">
```

```
<input type="submit">
```

```
<input type="reset">
```

```
<input type="hidden">
```

```
<input type="number">
```

```
<input type="date">
```

```
<input type="time">
```

 No file chosen

Video

- Before HTML5, a video could only be played in a browser with a plug-in (like flash).
- The HTML5 `<video>` element specifies a standard way to embed a video in a web page.

1. `<video width="320" height="240" controls>`
2. `<source src="movie.mp4" type="video/mp4">`
3. `<source src="movie.ogv" type="video/ogg">`
4. Your browser does not support the video tag.
5. `</video>`



Audio

- Before HTML5, audio files could only be played in a browser with a plug-in (like flash).
- The HTML5 `<audio>` element specifies a standard way to embed audio in a web page.

```
1. <audio controls>
2.   <source src="horse.ogg" type="audio/ogg">
3.   <source src="horse.mp3" type="audio/mpeg">
4. Your browser does not support the audio element.
5. </audio>
```



HTML Accessibility

Write HTML with accessibility in mind.
Provide the user a good way to
navigate and interact with your site.
Make your HTML code as semantic as
possible, so that the code is easy to
understand for visitors and screen
readers.



Semantic HTML means using correct HTML elements for their correct purpose as much as possible. Semantic elements are elements with a meaning; if you need a button, use the `<button>` element (and not a `<div>`)[1]



[1] https://www.w3schools.com/html/html_accessibility.asp

GIT

Version Control System

GIT



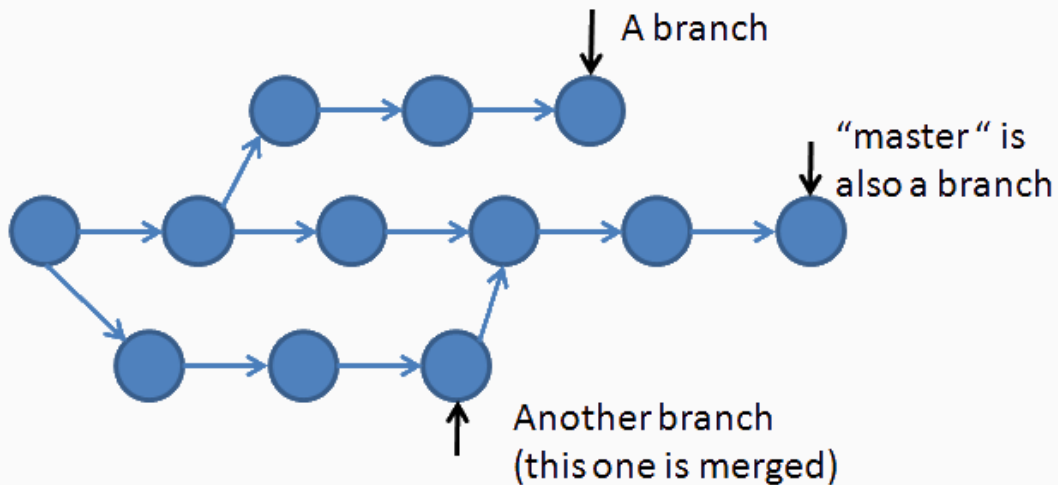
<https://git-scm.com/>

A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together.

GIT

Developers can review project history to find out:

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were changes needed?



GIT Commands

- » `git init` initializes a brand new Git repository and begins tracking an existing directory.
- » `git clone` creates a local copy of a project that already exists remotely.
- » `git add` stages a change.
- » `git commit` saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo.
- » `git status` shows the status of changes as untracked, modified, or staged.
- » `git branch` shows the branches being worked on locally.
- » `git merge` merges lines of development together.
- » `git pull` updates the local line of development with updates from its remote counterpart.
- » `git push` updates the remote repository with any commits made locally to a branch.

Questions?

Next: CSS