



Web Development

Download FREE Notes for Computer Science and related resources only at

Kwiknotes.in

Don't forget to check out our social media handles, do share with your friends.



History of the Internet and World Wide Web:

Internet History:

- The Internet's origins can be traced back to the 1960s when the U.S. Department of Defense created ARPANET, a network of computers.
- ARPANET was the precursor to the modern Internet and allowed for communication between computers at different locations.

World Wide Web (WWW):

- In 1989, British computer scientist Tim Berners-Lee invented the World Wide Web.
- The WWW introduced concepts like URLs (Uniform Resource Locators), HTML (Hypertext Markup Language), and HTTP (Hypertext Transfer Protocol).

Search Engines, News Groups, and Email:

Search Engines:

- Search engines like Google, Yahoo, and Bing help users find information on the web.
- They use web crawlers to index web pages and algorithms to rank search results.

News Groups:

- Usenet newsgroups are discussion groups on the Internet where users can post and read messages on various topics.

Email and Protocols:

- Email (electronic mail) is a common method of online communication.
- SMTP (Simple Mail Transfer Protocol) and IMAP (Internet Message Access Protocol) are used for sending and receiving emails.

Web Portals, Browsers, and URLs:

Web Portals:

- Web portals like Yahoo, MSN, and AOL serve as a starting point for accessing various online services.
- They provide news, email, search, and more in one place.

Web Browsers:

- Web browsers like Chrome, Firefox, Safari, and Edge are applications used to access websites and web content.
- They render HTML, CSS, and JavaScript to display web pages.

URLs (Uniform Resource Locators):

- URLs are web addresses that specify the location of resources on the Internet.
- They consist of a protocol (e.g., http://), domain name, and optional path.

Web Applications and HTML:

Introduction to Web Applications:

- Web applications are interactive software programs accessible through web browsers.
- Examples include online shopping, social media, and email services.

HTML (Hypertext Markup Language):

- HTML is the standard markup language for creating web pages.
- It uses tags to structure and format content.
- Example HTML structure:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>My Web Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
```

Client-Side vs. Server-Side Scripting:

Client-Side Scripting:

- Client-side scripting involves executing code on the user's device (browser).
- JavaScript is a common client-side scripting language.
- It's used for interactivity and user interface enhancements.

Server-Side Scripting:

- Server-side scripting involves executing code on a web server.
- It's used for processing data, generating dynamic content, and interacting with databases.
- Examples include PHP, Python, and Ruby.

Web Servers and Installation:

Web Servers:

- Web servers are software or hardware systems that store and deliver web content to users.
- Popular web servers include Apache, Nginx, and Microsoft Internet Information Services (IIS).

Local Servers and Remote Servers:

- Local servers are used for development and testing on a local machine.
- Remote servers are hosted on the Internet and serve websites to users.

Installing Web Servers:

- You can install web servers like Apache and Nginx on your local machine for development.
- For production, hosting providers offer web server solutions.

Certainly! Here are detailed notes on Internet Information Server (IIS), Personal Web Server (PWS), and static web development:

Internet Information Server (IIS) and Personal Web Server (PWS):

Internet Information Server (IIS):

- Internet Information Server, or IIS, is a web server software developed by Microsoft.
- It is commonly used for hosting websites on Windows servers.
- IIS supports various web technologies, including ASP.NET and PHP.

Personal Web Server (PWS):

- Personal Web Server, often abbreviated as PWS, is a lightweight web server software developed by Microsoft.
- It was designed for personal and small-scale web development and testing on Windows.
- PWS provides a local web server environment on your computer.

Static Web Development using HTML:

Introduction to HTML (Hypertext Markup Language):

- HTML is the standard markup language for creating web pages.
- It uses tags to structure and format content.

HTML Document Structure Tags:

- HTML documents have a basic structure with `<html>`, `<head>`, and `<body>` elements.
- `<html>`: The root element of an HTML page.
- `<head>`: Contains metadata like the page title.
- `<body>`: Contains the visible content.

HTML Comments:

- Comments in HTML are created with `<!--` and `-->` and are not visible to users.
- They are used for adding notes or explanations in the HTML code.

Text Formatting:

- HTML tags like `` (bold), `<i>` (italic), and `<u>` (underline) are used for text formatting.

Inserting Special Characters:

- Special characters, such as © and ®, can be inserted using HTML entities (e.g., `©` and `®`).

Anchor Tag:

- The anchor tag `<a>` is used to create hyperlinks to other web pages or resources.
- Example: `Visit Example`

Adding Images and Sound:

- Images are displayed using the `` tag, while sound can be added with the `<audio>` tag.
- Example for an image: ``

Lists (Types of Lists):

- HTML supports ordered lists (``), unordered lists (``), and definition lists (`<dl>`).
- Items within lists are defined with `` (list item) tags.

Tables:

- Tables are created with the `<table>` element.
- Rows are defined with `<tr>` (table row) tags, and data cells are created with `<td>` (table data) or `<th>` (table header) tags.

Frames and Floating Frames:

- HTML frames are used to divide a web page into multiple sections.
- Floating frames provide floating content within a page.
- Frames are less commonly used in modern web development.

Developing Forms:

- HTML forms are used for user input and data submission.
- Form elements include ``, ``, and `<select>`.- Forms use the `<form>` element to encapsulate form controls.</div><div data-bbox="111 395 568 419" data-label="Section-Header"><h3>Static Website vs. Dynamic Website Development:</h3></div><div data-bbox="111 425 885 490" data-label="List-Group">- A static website's content remains fixed and doesn't change based on user interactions.- Dynamic websites use server-side scripting to generate content on-the-fly, often driven by a database.</div><div data-bbox="111 525 417 551" data-label="Section-Header"><h3>Cascading Style Sheet (CSS):</h3></div><div data-bbox="111 565 640 611" data-label="List-Group">- CSS is used for styling and formatting web pages.- It separates the presentation (style) from the content (HTML).</div><div data-bbox="111 619 333 639" data-label="Section-Header"><h4>Some important topics :</h4></div><div data-bbox="111 647 227 665" data-label="Section-Header"><h5>1. Selectors:</h5></div><div data-bbox="125 675 519 749" data-label="List-Group">- Select elements by tag: <code>p { property: value; }</code>- Select by class: <code>.myclass { property: value; }</code>- Select by ID: <code>#myid { property: value; }</code></div><div data-bbox="111 784 238 803" data-label="Section-Header"><h5>2. Properties:</h5></div><div data-bbox="125 811 574 886" data-label="List-Group">- Change text color: <code>color: red;</code>- Set background color: <code>background-color: #00FF00;</code>- Adjust font size: <code>font-size: 16px;</code></div>

3. Box Model:

- Set padding: ``padding: 10px;``
- Define borders: ``border: 1px solid #000;``
- Adjust margins: ``margin: 20px;``

4. Positioning:

- Position elements absolutely: ``position: absolute;``
- Center an element: ``margin: 0 auto;``

5. Pseudo-classes:

- Style links on hover: ``a:hover { property: value; }``
- Style the first child: ``li:first-child { property: value; }``

6. Media Queries:

- Define responsive styles: ``@media (max-width: 768px) { /* styles */ }``

7. Comments:

- Add comments: ``/* This is a comment */``

8. Font Styling:

- Set font family: ``font-family: Arial, sans-serif;``
- Make text bold: ``font-weight: bold;``

9. Transforms:

- Rotate an element: ``transform: rotate(45deg);``
- Scale an element: ``transform: scale(1.5);``

10. Animations:

- Create a fade-in effect: ``animation: fadeIn 2s ease-in;``

JavaScript (JS):

JavaScript is a versatile programming language used for adding interactivity to web pages. It allows you to manipulate the Document Object Model (DOM), handle user interactions, and more.

1. Alert Dialog:

You can use JavaScript to display alert dialogs to the user.

eg:

```
alert("Hello, World!");
```

2. Variables:

JavaScript variables are used to store data.

eg:

```
var greeting = "Hello, World!";  
alert(greeting);
```

3. Functions:

Functions allow you to encapsulate a block of code for reuse.

eg:

```
function sayHello() {  
    alert("Hello, World!");  
}  
sayHello();
```

4. DOM Manipulation:

JavaScript can manipulate the DOM to change content or styles.

eg:

```
document.getElementById("myElement").innerHTML = "New Content";
```

5. Event Handling:

You can use JavaScript to respond to user actions, like button clicks.

Eg: **html**

```
<button id="myButton" onclick="handleClick()">Click Me</button>
```

javascript

```
function handleClick() {  
    alert("Button Clicked!"); }  
}
```

6. Conditional Statements:

JavaScript supports if statements for making decisions.

eg:

```
var age = 18;  
if (age >= 18) {  
    alert("You are an adult.");  
} else {  
    alert("You are a minor.");  
}
```

7. Loops:

Loops are used for repetitive tasks.

eg:

```
for (var i = 1; i <= 5; i++) {  
    console.log("Count: " + i);  
}
```

8. External Scripts:

You can include external JavaScript files in your HTML.

eg:

html

```
<script src="myscript.js"></script>
```

(Where "myscript.js" contains your JavaScript code.)

Introduction to PHP:

- PHP (Hypertext Preprocessor) is a widely used server-side scripting language.
- It is embedded within HTML and is executed on the web server.
- PHP is used to create dynamic web pages, process form data, interact with databases, and more.
- It's open-source and has a large community of developers.

Start and End Tags of PHP:

- PHP code is enclosed in special tags: `<?php` to start and `?>` to end.
- You can also use short tags `<?` to start and `?>` to end, but it's recommended to use the full tags for compatibility.

Data Types in PHP:

- PHP supports various data types, including integers, floats, strings, booleans, arrays, objects, and more.
- You don't need to declare data types explicitly; PHP dynamically determines the data type.

Variables:

- Variables in PHP are used to store data. They are case-sensitive.
- To create a variable, use the `$` symbol followed by the variable name.
- Variable names must start with a letter or underscore, followed by letters, numbers, or underscores.

Constants:

- Constants are used to store values that cannot be changed during the script's execution.
- To define a constant, use the `define()` function.

Operators and Expressions:

- PHP supports various operators, including arithmetic, assignment, comparison, logical, and more.
- Expressions are formed by combining values and variables using operators.

Printing Data on PHP Page:

- Use `echo` or `print` to display data on a PHP page.
- For example:

```
$message = "Hello, World!";  
echo $message;
```

Control Statements:

1. if Statement: Used for conditional execution.

```
if (condition) {  
    // Code to execute if condition is true  
}
```

2. switch-case Statement: A multi-way branch statement.

```
switch (value) {  
    case 1:  
        // Code for case 1  
        break;  
    case 2:  
        // Code for case 2  
        break;  
    default:  
        // Code to execute if no cases match  
}
```

3. for Loop: Used for iterative tasks.

```
for ($i = 0; $i < 5; $i++) {  
    // Code to execute in each iteration }
```

4. while Loop: Continues executing as long as a condition is true.

```
while (condition) {  
    // Code to execute in each iteration  
}
```

5. do-while Loop: Similar to the while loop but always executes the code block at least once.

```
do {  
    // Code to execute in each iteration  
} while (condition);
```

Arrays:

- Arrays are used to store multiple values in a single variable.
- PHP supports indexed arrays, associative arrays, and multidimensional arrays.
- Example of initializing an indexed array:

```
$fruits = array("apple", "banana", "cherry");
```

Working with Forms: Get and Post Methods:

- HTML forms are used to collect user input.
- Forms can use two methods to send data to the server: GET and POST.
- **GET Method:** Appends data to the URL. Suitable for small amounts of data.
- **POST Method:** Sends data in the request body. Suitable for larger data or sensitive information.

Query Strings

- When using the GET method, data is passed as query strings in the URL.
- Query strings are in the form of `key=value` pairs, separated by `&`.
- For example: `http://example.com/page.php?name=John&age=30`

HTML Form Controls and PHP:

- HTML forms consist of various input controls like text fields, radio buttons, checkboxes, dropdown lists, etc.
- PHP can process the data submitted through these controls.
- Example HTML form:

```
<form action="process.php" method="POST">  
  Name: <input type="text" name="name"><br>  
  Age: <input type="number" name="age"><br>  
  <input type="submit" value="Submit">  
</form>
```

- **PHP can access form data using the `$_POST` superglobal.**

```
$name = $_POST['name'];  
$age = $_POST['age'];
```

Maintaining User State: Cookies, Sessions, and Application State:

- To maintain user state across multiple requests, PHP provides mechanisms like cookies, sessions, and application state.

Cookies:

- Cookies are small pieces of data sent from a web server and stored on the user's device.
- They can be used to store user-specific information like preferences, login status, etc.
- PHP sets cookies using `setcookie()` and reads them from `$_COOKIE`.

Sessions:

- Sessions are server-side storage for user data.
- A unique session ID is used to identify a user.
- Sessions are often used for user authentication and storing temporary data.
- Sessions are initiated using `session_start()`.
- Data is stored and retrieved from `$_SESSION`.

Application State:

- Application state involves storing data globally for all users.
- It is useful for sharing data between different parts of the application.
- PHP provides options like using global variables or using a database for application state.

Certainly! Here are detailed notes on PHP database connectivity with MySQL:

PHP Database Connectivity: Introduction to MySQL:

- MySQL is a popular open-source relational database management system.
- It's widely used for storing, retrieving, and managing structured data.

Creating a Database and Other Operations on a Database:

- In MySQL, you can create a new database using SQL commands or through a database management tool.
- Example SQL command to create a new database named "mydb":

```
CREATE DATABASE mydb;
```
- Once the database is created, you can perform various operations like creating tables, altering tables, inserting data, updating data, and deleting data using SQL commands.

Connecting to a Database:

- To work with a MySQL database in PHP, you need to establish a connection.
- PHP provides functions like `mysqli_connect()` or PDO (PHP Data Objects) for connecting to MySQL databases.

Using a Particular Database:

- After connecting to MySQL, you can select the specific database you want to work with using the `USE` SQL statement.

- Example:

```
$db = mysqli_connect("localhost", "username", "password");  
mysqli_select_db($db, "mydb");
```

Sending Queries to the Database:

- You can execute SQL queries in PHP to interact with the database.
- Two main methods to execute queries are `mysqli_query()` and prepared statements.
- Example using `mysqli_query()`:

```
$query = "SELECT name, age FROM users WHERE id = 1";  
$result = mysqli_query($db, $query);
```