MAA OMWATI DEGREE COLLEGE (HASSANPUR)

Subject:- Web Application Development

Program

Class:- BCA 5TH SEM

IMPORTANT NOTES FOR EXAM

UNIT-1

Introduction to the World Wide Web (WWW)

---

1. Concept of World Wide Web (WWW) --

Definition

The **World Wide Web (WWW)** is a system of interconnected web pages and resources that can be accessed through the Internet using a web browser. It was invented by **Tim Berners-Lee** in **1989** to make it easier to share and access information. The WWW uses **Hypertext**, **Hyperlinks**, **HTTP/HTTPS**, and **URLs** to connect documents stored on web servers.

Key Points

- WWW is an information-sharing system on the Internet.
- Uses **web pages** written mainly in **HTML**.
- Web pages are connected through **hyperlinks**.
- Every web page has a unique **URL (Uniform Resource Locator)**.
- Users access web pages using web browsers.
- Information is stored on web servers.
- Uses **HTTP/HTTPS** for communication.

Advantages

- Easy access to information.
- Global communication.
- Supports multimedia (text, images, audio, video).
- Easy navigation using hyperlinks.
- Supports online services and e-commerce.

Example

Opening **https://www.wikipedia.org**

in a browser displays web pages stored on web servers.

Exam Definition:
The **World Wide Web (WWW)** is a collection of interconnected web pages and resources accessed over the Internet using web browsers and HTTP/HTTPS protocols.

---

2. Internet and WWW --

Definition

The **Internet** is a global network of interconnected computers that communicate using the **TCP/IP protocol**. The **World Wide Web (WWW)** is one of the services that runs on the Internet and provides access to web pages through web browsers.

Key Points

- The Internet is the communication network.
- The WWW is a service that operates on the Internet.
- Internet supports services like Email, FTP, VoIP, and WWW

- .
- WWW uses HTTP/HTTPS protocols.
- Internet uses TCP/IP for communication.
- WWW provides information in the form of web pages.

Difference Between Internet and WWW

Internet	World Wide Web (WWW)
Global network of computers	Collection of web pages
Uses TCP/IP	Uses HTTP/HTTPS
Supports many services	Only web-based services
Infrastructure	Information system

Example

When you use **Google Chrome** to open **www.google.com**

, the **Internet** provides the connection, while the **WWW** provides the web page.

Exam Definition:
The **Internet** is a worldwide network of computers, whereas the **World Wide Web (WWW)** is a collection of web pages accessed through the Internet.

---

3. HTTP Protocol: Request and Response --

Definition

HTTP (HyperText Transfer Protocol) is the communication protocol used for transferring web pages and other resources between a web browser (client) and a web server. It follows a **request-response model**.

HTTP Request

A browser sends an HTTP request to the server asking for a web page or resource.

Common Request Methods

- **GET** – Retrieve data.
- **POST** – Send data.
- **PUT** – Update data.
- **DELETE** – Remove data.

HTTP Response

The server processes the request and sends back a response containing:

- Status code (200, 404, 500, etc.)
- Headers
- Requested web page or data

Common HTTP Status Codes

- **200 OK** – Request successful.
- **404 Not Found** – Resource not found.
- **500 Internal Server Error** – Server error.
- **301 Moved Permanently** – Page redirected.

Example

Typing **www.google.com**

in a browser sends an HTTP request. Google's server returns the homepage as an HTTP response.

Exam Definition:
HTTP is the protocol used for communication between a web browser and a web server using a request-response mechanism.

---

4. Web Browser and Web Server --

(A) Web Browser

Definition

A **web browser** is software that allows users to access, retrieve, and display web pages from the Internet. It interprets HTML, CSS, and JavaScript to present websites in a user-friendly format.

Examples

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari
- Opera

Functions

- Sends HTTP requests.
- Displays web pages.
- Stores cookies and cache.
- Supports bookmarks and downloads.

(B) Web Server

Definition

A **web server** is software or hardware that stores website files and delivers them to users upon request through HTTP/HTTPS.

Examples

- Apache HTTP Server
- Nginx
- Microsoft IIS
- LiteSpeed

Functions

- Stores web pages.
- Processes client requests.
- Sends HTTP responses.
- Manages website traffic.

Difference

Web Browser	Web Server
Client software	Server software/hardware
Requests web pages	Delivers web pages
Runs on user device	Runs on server machine

Exam Definition:
A web browser requests and displays web pages, while a web server stores and delivers web pages to users over HTTP/HTTPS.

---

5. Features of the Latest Version of the Web (Web 3.0 / Modern Web) --

Definition

The latest generation of the web, often called the **Modern Web** or **Web 3.0**, focuses on intelligent, secure, decentralized, and user-centric web applications. It integrates technologies such as Artificial Intelligence (AI), cloud computing, blockchain, semantic web, and responsive design to provide faster and smarter user experiences.

Key Features

1. Responsive Design

- Websites automatically adjust to mobile phones, tablets, and desktops.

2. Artificial Intelligence (AI)

- Personalized search results and recommendations.
- Smart chatbots and virtual assistants.

3. Cloud Integration

- Data stored and accessed from cloud services.
- Easy synchronization across devices.

4. High Security

- HTTPS encryption.
- Multi-factor authentication.
- Better privacy protection.

5. Semantic Web

- Machines understand data meaning for more accurate search results.

6. Rich Multimedia Support

- High-quality video, audio, animations, and interactive content.

7. Progressive Web Applications (PWAs)

- Websites that work like mobile applications.
- Offline access and fast loading.

Advantages

- Faster browsing.
- Better security.
- Mobile-friendly.
- Personalized user experience.
- Improved accessibility.

Example

Modern websites such as YouTube, Amazon, and Gmail use AI, cloud computing, responsive design, and HTTPS to deliver secure and personalized services.

Exam Definition:
The latest version of the web provides intelligent, secure, mobile-friendly, and cloud-enabled web services using modern technologies such as AI, semantic web, and responsive design.

Quick Revision Table

Topic	One-Line Definition
Concept of WWW	A collection of interconnected web pages accessed through the Internet.
Internet and WWW	The Internet is the global network; the WWW is a web-based information service that runs on it.
HTTP Protocol	A request-response protocol used for communication between web browsers and web servers.
Web Browser	Software used to access and display web pages.
Web Server	Software or hardware that stores and delivers web pages to browsers.
Features of the Latest Web	Includes responsive design, AI, cloud integration, HTTPS, semantic web, and progressive web applications (PWAs).

Important Exam Points

WWW: Invented by **Tim Berners-Lee** in **1989**.

- **Internet vs WWW:** Internet = network; WWW = service using that network.
- **HTTP:** Understand request methods (GET, POST, PUT, DELETE) and common status codes (200, 404, 500, 301).
- **Browser vs Server:** Browser requests pages; server stores and sends them.
- **Modern Web:** AI, cloud integration, semantic web, HTTPS, responsive design, and PWAs are key features.

Web Design – Exam Notes

---

1. Concepts of Effective Web Design --

Definition

Effective web design is the process of creating websites that are attractive, easy to use, accessible, and provide a good user experience. A well-designed website combines visual design, functionality, performance, and usability to meet user needs.

Key Principles

- **Simplicity:** Keep the design clean and avoid unnecessary elements.
- **Consistency:** Use similar colors, fonts, and layouts throughout the website.

- **Usability:** Users should easily find information and complete tasks.
- **Accessibility:** Website should be usable by people with different abilities.
- **Fast Loading:** Pages should load quickly to improve user experience.
- **Responsive Design:** Website should work properly on mobiles, tablets, and desktops.
- **Visual Hierarchy:** Important information should be highlighted.
- **Quality Content:** Provide accurate, clear, and useful information.

Advantages

- Improves user satisfaction.
- Increases website traffic.
- Builds trust and credibility.
- Helps users navigate easily.
- Improves search engine ranking.

Exam Definition:
Effective web design is the practice of creating websites that are visually appealing, user-friendly, accessible, and efficient.

---

2. Web Design Issues: Browser Compatibility --

Definition

Browser compatibility refers to the ability of a website to function correctly across different web browsers such as Chrome, Firefox, Edge, and Safari. Different browsers may interpret HTML, CSS, and JavaScript differently, causing display or functionality problems.

Key Points

- Websites should be tested on multiple browsers.
- HTML and CSS standards should be followed.
- Avoid browser-specific code.
- Use responsive web design techniques.
- Ensure JavaScript works properly on different browsers.
- Check fonts, images, forms, and animations.

Common Problems

- Incorrect page layout.
- Missing images or features.
- Different font appearance.

- JavaScript errors.
- Poor mobile compatibility.

Solutions

- Perform cross-browser testing.
- Use modern web standards.
- Keep browsers updated.
- Use browser developer tools.

Example

A website may look correct in Chrome but display incorrectly in an older version of Internet Explorer.

Exam Definition:
Browser compatibility ensures that a website works properly and provides the same user experience across different web browsers.

---

3. Bandwidth and Web Design --

Definition

Bandwidth is the amount of data that can be transferred through a network connection in a given time. Website design must consider bandwidth limitations because large files can slow down page loading, especially for users with slow Internet connections.

Key Points

- Large images and videos consume more bandwidth.
- Optimized websites load faster.
- Use compressed images and files.
- Avoid unnecessary animations and heavy scripts.
- Use Content Delivery Networks (CDNs) for faster delivery.
- Optimize HTML, CSS, and JavaScript files.

Effects of Poor Bandwidth Management

- Slow page loading.
- Higher user abandonment.
- Increased data usage.
- Poor user experience.

Solutions

- Compress images.
- Reduce file sizes.
- Use efficient coding.
- Enable browser caching.

Example

An e-commerce website with optimized product images loads faster than one with large uncompressed images.

Exam Definition:
Bandwidth consideration in web design involves optimizing website content to reduce data usage and improve loading speed.

---

4. Cache in Web Design --

Definition

A cache is a temporary storage area that saves frequently accessed website files such as images, CSS, and JavaScript. Browsers use caching to reduce loading time by storing copies of website resources locally.

Key Points

- Improves website loading speed.
- Reduces server requests.
- Saves bandwidth.
- Enhances user experience.
- Stores frequently used resources.
- Helps websites perform better.

Types of Cache

1. **Browser Cache**
    - Stores files on the user's device.
2. **Server Cache**
    - Stores frequently requested data on the server.
3. **CDN Cache**
    - Stores website content on multiple global servers.

Problems

- Users may see outdated content.
- Cache management is required.

Solutions

- Use cache expiration dates.
- Clear cache when updates are made.
- Use proper cache-control settings.

Exam Definition:
Caching is a technique of storing frequently used web resources temporarily to improve website speed and reduce server load.

---

5. Display Resolution in Web Design --

Definition

Display resolution refers to the number of pixels displayed on a screen horizontally and vertically. Web designers must consider different screen sizes and resolutions to ensure websites appear correctly on all devices.

Key Points

- Different devices have different screen resolutions.
- Websites should support desktop, tablet, and mobile screens.
- Responsive design adjusts layout automatically.
- Images should be optimized for different displays.
- Text should remain readable on all devices.

Common Resolutions

- Desktop: 1920 × 1080 pixels
- Tablet: 768 × 1024 pixels
- Mobile: 360 × 800 pixels (approx.)

Design Solutions

- Use flexible layouts.
- Use CSS media queries.
- Avoid fixed-width designs.
- Test websites on multiple devices.

Importance

- Improves usability.
- Provides consistent appearance.
- Increases mobile accessibility.

Exam Definition:
Display resolution consideration ensures that websites display correctly and remain usable across different screen sizes and devices.

---

6. Look and Feel of a Website --

Definition

The look and feel of a website refers to its visual appearance and the overall experience it provides to users. It includes colors, fonts, images, graphics, layout, and overall design style.

Key Elements

- **Color Scheme:** Creates visual identity and improves readability.
- **Typography:** Proper fonts improve understanding and appearance.
- **Images and Graphics:** Enhance communication and attraction.
- **White Space:** Makes content easier to read.
- **Consistency:** Maintains professional appearance.
- **Brand Identity:** Represents company values and style.

Importance

- Creates a positive first impression.
- Builds user trust.
- Improves navigation.
- Makes content attractive.
- Supports brand recognition.

Good Practices

- Use readable fonts.
- Avoid excessive colors.
- Maintain consistent design.
- Use high-quality images.

Exam Definition:
The look and feel of a website describes its visual design elements that influence user perception and overall experience.

---

7. Page Layout and Linking --

Definition

Page layout is the arrangement of text, images, menus, and other elements on a webpage. Linking connects different web pages and resources using hyperlinks, allowing users to navigate through a website.

Page Layout Elements

- Header: Contains logo and main information.
- Navigation Bar: Provides links to important sections.
- Content Area: Displays main information.
- Sidebar: Shows additional information.
- Footer: Contains contact details and copyright information.

Linking Types

1. **Internal Links**
    - Connect pages within the same website.
2. **External Links**
    - Connect to other websites.
3. **Anchor Links**
    - Move users to specific sections of a page.

Importance

- Improves navigation.
- Organizes information.
- Helps users find content quickly.
- Improves search engine optimization (SEO).

Exam Definition:
Page layout organizes webpage elements, while linking connects web pages and resources for easy navigation.

---

8. User-Centric Design --

Definition

User-centric design is a design approach that focuses on the needs, preferences, and behaviors of users. The goal is to create websites that are easy, efficient, and enjoyable to use.

Key Principles

- Understand user requirements.
- Design according to user expectations.
- Provide simple navigation.
- Make information easy to find.
- Ensure accessibility.
- Collect user feedback.
- Improve design continuously.

Features

- Simple interface.
- Clear menus.
- Readable content.
- Fast response time.
- Mobile-friendly design.

Advantages

- Better user satisfaction.
- Increased engagement.
- Reduced learning time.
- Improved usability.
- Higher conversion rates.

Example

An online shopping website provides search filters, simple checkout, and clear product information based on customer needs.

Exam Definition:
User-centric design is an approach that creates websites by focusing on user needs, usability, and overall experience.

---

9. Sitemap --

Definition

A sitemap is a structured list or diagram that shows the organization and relationship between pages of a website. It helps designers plan website structure and helps search engines understand website content.

Types of Sitemaps

1. HTML Sitemap

- Created for website visitors.
- Helps users navigate pages.

2. XML Sitemap

- Created for search engines.
- Helps search engines index web pages.

Key Points

- Shows website hierarchy.
- Improves navigation planning.
- Helps SEO.
- Identifies missing pages.
- Organizes website content.

Advantages

- Better website planning.
- Easier maintenance.
- Improved search engine visibility.
- Better user experience.

Example

A university website sitemap may contain Home → Departments → Courses → Faculty pages.

Exam Definition:
A sitemap is a visual or structured representation of website pages that helps users and search engines understand website organization.

---

10. Planning and Publishing Website --

Definition

Website planning is the process of defining goals, content, structure, and design before creating a website. Website publishing is the process of making the completed website available on the Internet.

Planning Steps

- Define website purpose.
- Identify target audience.
- Prepare content.
- Create sitemap.
- Design page layout.
- Select domain name and hosting.
- Test website functionality.

Publishing Steps

- Register domain name.
- Select web hosting service.
- Upload website files.
- Configure server settings.
- Test website online.
- Maintain and update content.

Key Points

- Proper planning reduces errors.
- Testing improves quality.
- Hosting makes website accessible.
- Regular updates maintain performance.

Exam Definition:
Website planning involves designing and organizing a website, while publishing makes the website available to Internet users.

---

11. Designing Effective Navigation --

Definition

Website navigation is the system that allows users to move between different pages and find information easily. Effective navigation improves usability and user satisfaction.

Key Principles

- Keep menus simple and clear.
- Use meaningful labels.
- Maintain consistent navigation throughout the website.
- Provide search functionality.
- Include breadcrumbs for easy movement.
- Use logical page structure.
- Highlight current page location.

Types of Navigation

- **Global Navigation:** Main menu available on all pages.
- **Local Navigation:** Links within a specific section.
- **Breadcrumb Navigation:** Shows user's location in website hierarchy.
- **Footer Navigation:** Additional links at the bottom of pages.

Advantages

- Helps users find information quickly.
- Reduces confusion.
- Improves user experience.
- Increases website engagement.

Exam Definition:
Effective navigation is the design of website menus and links that allow users to easily access information and move between pages.

Quick Revision Table

Topic	Key Idea
Effective Web Design	Creates attractive, usable, accessible, and efficient websites.
Browser Issues	Ensures websites work correctly across different browsers.
Bandwidth	Website optimization reduces data usage and improves speed.
Cache	Stores resources temporarily to improve loading time.
Display Resolution	Ensures proper display on different screen sizes.
Look and Feel	Visual appearance and user experience of a website.
Page Layout & Linking	Organizes content and connects web pages.
User-Centric Design	Focuses on user needs and usability.
Sitemap	Shows website structure and page relationships.
Planning & Publishing	Steps for creating and launching a website.
Navigation Design	Helps users move easily through a website.

Unit – II

HTML and Web Technologies – Exam Notes

---

1. Basics of HTML --

Definition

HTML (HyperText Markup Language) is the standard markup language used to create and design web pages. It defines the structure and content of a webpage using elements called **tags**. HTML documents are interpreted by web browsers and displayed as web pages.

Key Points

- HTML was created by **Tim Berners-Lee** in 1991.
- HTML uses tags enclosed within angle brackets (< >).
- HTML documents contain elements, attributes, and content.
- It defines webpage structure such as headings, paragraphs, images, links, and forms.
- HTML works together with CSS for styling and JavaScript for interactivity.

Basic HTML Structure

- `<html>` – Root element of the webpage.
- `<head>` – Contains metadata and page information.
- `<title>` – Defines webpage title.
- `<body>` – Contains visible webpage content.

Advantages

- Easy to learn and use.
- Supported by all browsers.
- Platform independent.
- Supports multimedia and interactive content.

Exam Definition:
HTML is a markup language used to create and structure content on web pages using tags and elements.

---

2. HTML Formatting and Fonts --

Definition

HTML formatting is used to change the appearance and presentation of text on web pages. Formatting tags help make content readable and visually attractive.

Common Formatting Tags

- `<b>` – Displays bold text.
- `<i>` – Displays italic text.
- `<u>` – Underlines text.
- `<strong>` – Indicates important text.
- `<em>` – Emphasizes text.
- `<sup>` – Creates superscript text.
- `<sub>` – Creates subscript text.
- `<br>` – Inserts a line break.

Font Formatting

The `<font>` tag was previously used to change:

- Font style.
- Font size.
- Font color.

Modern HTML uses **CSS** instead of the `<font>` tag.

Importance

- Improves readability.
- Highlights important information.
- Creates better webpage appearance.

Example Uses

- Bold headings.
- Highlighted keywords.
- Mathematical expressions using superscript and subscript.

Exam Definition:
HTML formatting controls the appearance of text using formatting elements and CSS styling.

---

3. Commenting Code in HTML --

Definition

Comments are non-executable statements added inside HTML code to explain the purpose of code sections. Browsers ignore comments and do not display them on webpages.

Syntax

```
<!-- This is an HTML comment -->
```

Key Points

- Used for documentation.
- Helps developers understand code.
- Makes maintenance easier.
- Can temporarily disable code during testing.
- Does not affect webpage output.

Advantages

- Improves code readability.
- Helps teamwork among developers.
- Makes debugging easier.
- Provides information about complex sections.

Types of Comments

- Single-line comments.
- Multi-line comments.

Example

```
<!-- Navigation menu starts here -->
<nav>
</nav>
```

Exam Definition:
HTML comments are hidden notes in HTML code used to explain and organize webpage code without affecting its display.

---

4. Colors in HTML --

Definition

HTML colors are used to define the visual appearance of webpage elements such as text, backgrounds, and borders. Colors improve website design and user experience.

Methods of Applying Colors

1. Color Names

Example:

```
<p style="color:red;">Text</p>
```

2. Hexadecimal Values

Uses six-digit color codes.
Example:
`#FF0000` represents red.

3. RGB Values

Defines colors using Red, Green, and Blue values.
Example:
`rgb(255,0,0)`

4. CSS Colors

Modern websites use CSS properties:

- `color`
- `background-color`
- `border-color`

Key Points

- Improves readability.
- Creates visual identity.
- Highlights important content.
- Supports branding.

Example

A company website uses specific colors to maintain brand recognition.

Exam Definition:
HTML colors define the appearance of webpage elements using color names, hexadecimal codes, RGB values, or CSS properties.

---

5. Hyperlinks in HTML --

Definition

A hyperlink is a connection between one webpage and another resource. It allows users to navigate between pages, websites, files, or specific sections of a page.

HTML Tag

The anchor tag `<a>` is used to create hyperlinks.

Example:

```
<a href="https://example.com">Visit Website</a>
```

Types of Hyperlinks

1. **Internal Links**

- Connect pages within the same website.

2. **External Links**

- Connect to other websites.

3. **Email Links**

- Opens email applications.

4. **Bookmark Links**

- Move to specific sections of a webpage.

Attributes

- `href` – Specifies destination.
- `target` – Defines where to open the link.

Importance

- Improves website navigation.
- Connects information resources.
- Enhances user experience.
- Supports SEO.

Exam Definition:
A hyperlink is an HTML element that connects webpages or resources and allows users to navigate through the web.

---

6. Lists in HTML --

Definition

HTML lists are used to organize information in a structured format. Lists improve readability and help present related items clearly.

Types of Lists

1. Ordered List (`<ol>`)

- Displays items with numbers.
- Used when sequence matters.

Example:

1. HTML
2. CSS
3. JavaScript

2. Unordered List (`<ul>`)

- Displays items with bullets.
- Used when order is not important.

3. Description List (`<dl>`)

- Contains terms and descriptions.

Important Tags

- `<li>` – List item.
- `<dt>` – Definition term.

- `<dd>` – Definition description.

Applications

- Menus.
- Instructions.
- Product features.
- Navigation links.

Exam Definition:
HTML lists organize information into ordered, unordered, or descriptive formats using list elements.

---

7. Tables in HTML --

Definition

HTML tables are used to display data in rows and columns. They are useful for presenting structured information such as records, schedules, and reports.

Important Tags

- `<table>` – Creates a table.
- `<tr>` – Creates a table row.
- `<td>` – Creates table data.
- `<th>` – Creates table heading.
- `<caption>` – Adds table title.

Features

- Organizes data clearly.
- Supports row and column formatting.
- Can include images and links.
- CSS improves table design.

Attributes

- Border.
- Width.
- Height.
- Cell padding.
- Cell spacing.

Example Uses

- Student marks table.
- Employee records.
- Product comparison.

Exam Definition:
An HTML table is a structure used to represent data in rows and columns using table-related tags.

---

8. Images in HTML --

Definition

Images make webpages attractive and improve communication. HTML provides the `<img>` tag to insert images into web pages.

Syntax

```
<img src="image.jpg" alt="Image">
```

Important Attributes

- **src:** Specifies image location.
- **alt:** Provides alternative text.
- **width:** Sets image width.
- **height:** Sets image height.

Supported Formats

- JPEG/JPG.
- PNG.
- GIF.
- SVG.
- WebP.

Key Points

- Improves visual presentation.
- Supports multimedia websites.
- Alternative text improves accessibility.
- Images should be optimized for faster loading.

Best Practices

- Use compressed images.
- Provide meaningful alt text.
- Maintain proper dimensions.

Exam Definition:
HTML images are added using the `<img>` tag to display graphical content on web pages.

---

9. HTML Forms --

Definition

HTML forms are used to collect information from users and send it to a server for processing. Forms are commonly used in registration, login, feedback, and search systems.

Form Tag

`<form>` defines an HTML form.

Common Form Elements

- Text box (`<input type="text">`)
- Password field
- Radio button
- Checkbox
- Submit button
- Drop-down list (`<select>`)
- Text area (`<textarea>`)

Important Attributes

- `action` – Specifies where data is sent.
- `method` – Defines data transfer method (GET/POST).

Advantages

- Collects user data.
- Enables user interaction.
- Supports online applications.
- Provides input validation.

Exam Definition:
HTML forms are webpage elements used to collect and submit user information to a server.

---

10. XHTML --

Definition

XHTML (Extensible HyperText Markup Language) is an advanced version of HTML based on XML rules. It combines HTML features with XML strictness to create well-structured web documents.

Key Features

- All tags must be properly closed.
- Tags must be written in lowercase.
- Attributes must be quoted.
- Documents must have proper structure.
- Elements must be properly nested.

Advantages

- More organized code.
- Better compatibility with XML tools.
- Easier processing by machines.
- Improves document quality.

Difference from HTML

HTML	XHTML
Less strict	More strict
Flexible syntax	Requires proper syntax
Based on SGML	Based on XML

Exam Definition:
XHTML is an XML-based version of HTML that follows strict rules for creating well-formed web documents.

---

11. Meta Tags --

Definition

Meta tags provide information about a webpage to browsers and search engines. They are placed inside the `<head>` section of an HTML document.

Common Meta Tags

- **charset:** Defines character encoding.
- **description:** Provides webpage summary.
- **keywords:** Specifies search keywords.
- **viewport:** Controls mobile display.
- **author:** Identifies webpage creator.

Importance

- Improves search engine optimization (SEO).
- Controls browser behavior.
- Defines webpage information.
- Supports mobile responsiveness.

Example

```
<meta name="description" content="HTML Tutorial">
```

Exam Definition:
Meta tags are HTML elements that provide metadata information about webpages to browsers and search engines.

---

12. Character Entities in HTML --

Definition

Character entities are special codes used to display reserved characters or symbols in HTML documents. They prevent browsers from interpreting special characters as HTML code.

Common Entities

- `&lt;` → <
- `&gt;` → >
- `&amp;` → &
- ` ` → Space
- `&copy;` → Copyright symbol

- `&reg;` → Registered symbol

Key Points

- Used for special characters.
- Prevents HTML errors.
- Supports symbols and mathematical characters.
- Improves webpage compatibility.

Exam Definition:
HTML character entities are codes used to represent special characters and symbols that cannot be directly written in HTML.

---

13. Frames and Framesets in HTML --

Definition

Frames allow a browser window to be divided into multiple sections where each section displays a different HTML document. Frames were used to display multiple pages simultaneously.

Important Tags

- `<frameset>` – Defines frame structure.
- `<frame>` – Defines individual frames.

Types

- Row frames.
- Column frames.

Advantages

- Multiple pages displayed together.
- Faster navigation.
- Separate content areas.

Disadvantages

- Poor SEO support.
- Difficult bookmarking.
- Not supported in modern HTML5.

Modern Alternative

CSS layouts and `<iframe>` are used instead.

Exam Definition:
Frames divide a browser window into multiple sections, each displaying a separate HTML document.

---

14. Browser Architecture and Website Structure --

Browser Architecture

A web browser consists of several components:

- **User Interface:** Allows interaction with users.
- **Browser Engine:** Controls communication between UI and rendering engine.
- **Rendering Engine:** Displays HTML and CSS content.
- **JavaScript Engine:** Executes JavaScript code.
- **Networking Component:** Handles HTTP requests.
- **Data Storage:** Stores cookies, cache, and history.

Website Structure

A website usually contains:

- Home page.
- Navigation system.
- Content pages.
- Images and multimedia.
- Forms.
- Footer and contact information.

Key Points

- Browser converts HTML code into visual pages.
- Website structure improves navigation.
- Proper organization improves user experience.

Exam Definition:
Browser architecture defines the internal components that display web content, while website structure defines the organization of website pages.

---

15. Latest Version of HTML – HTML5 --

Definition

HTML5 is the latest major version of HTML used for developing modern web applications. It introduces new elements, multimedia support, improved graphics, and better application features.

Key Features of HTML5

- **Semantic Elements:**
  Tags like `<header>`, `<nav>`, `<section>`, `<article>`, and `<footer>` improve structure.
- **Multimedia Support:**
  Native audio and video elements:
  - `<audio>`
  - `<video>`
- **Canvas Element:**
  Supports drawing graphics using JavaScript.
- **Geolocation API:**
  Provides location-based services.
- **Web Storage:**
  Stores data locally using localStorage and sessionStorage.
- **Responsive Design Support:**
  Better support for mobile devices.
- **Web Workers:**
  Allows background processing.

Advantages

- Faster web applications.
- Reduced dependency on plugins.
- Better multimedia support.
- Improved security and performance.

Exam Definition:
HTML5 is the latest version of HTML that provides semantic elements, multimedia support, graphics, storage, and advanced web application features.

Quick Revision Table

Topic	Key Idea

Topic	Key Idea
HTML Basics	Language used to create webpage structure.
Formatting & Fonts	Controls text appearance and style.
Comments	Hidden notes used for code explanation.
Colors	Defines webpage visual appearance.
Hyperlinks	Connect webpages and resources.
Lists	Organize information using ordered/unordered formats.
Tables	Display data in rows and columns.
Images	Add graphical content to webpages.
Forms	Collect user information.
XHTML	Strict XML-based HTML version.
Meta Tags	Provide webpage information to browsers/search engines.
Character Entities	Display special symbols safely.
Frames	Divide browser window into sections.
Browser Architecture	Components responsible for displaying web content.
HTML5	Modern HTML version with multimedia and advanced features.

Stylesheets (CSS) – Exam Notes

---

1. Need for CSS (Cascading Style Sheets) --

Definition

CSS (Cascading Style Sheets) is a stylesheet language used to control the presentation and design of HTML documents. It separates the content of a webpage from its visual appearance, making websites easier to design and maintain.

Need for CSS

- **Separates Content and Design:** HTML manages webpage structure, while CSS controls appearance.
- **Consistency:** A single CSS file can apply the same style to multiple pages.
- **Improves Website Appearance:** Provides control over colors, fonts, layouts, and animations.
- **Reduces Code Repetition:** Avoids writing the same formatting code repeatedly.
- **Faster Loading:** External CSS files can be cached by browsers.

- **Easy Maintenance:** Changes can be applied to the entire website by modifying one stylesheet.
- **Responsive Design:** Helps create websites that work on different screen sizes.

Advantages

- Saves development time.
- Improves website performance.
- Provides better user experience.
- Supports modern web design.

Exam Definition:
CSS is used to style and format HTML documents by controlling the layout, colors, fonts, and overall appearance of webpages.

---

2. Introduction to CSS --

Definition

CSS stands for **Cascading Style Sheets**. It is a web technology used to define how HTML elements appear on a webpage. CSS controls the visual presentation of websites, including colors, fonts, spacing, and positioning.

Key Points

- Developed by **W3C (World Wide Web Consortium)**.
- Works together with HTML and JavaScript.
- Uses selectors and properties to apply styles.
- Supports responsive web design.
- Can create animations and interactive effects.
- Improves webpage organization.

Types of CSS

1. Inline CSS

- Style is written directly inside HTML elements.
- Used for individual elements.

2. Internal CSS

- CSS is written inside the `<style>` tag in the HTML document.

3. External CSS

- CSS is stored in a separate `.css` file.
- Best method for large websites.

Benefits

- Better control over webpage design.
- Easy updates.
- Consistent appearance.

Exam Definition:
CSS is a stylesheet language used to describe the presentation and layout of HTML-based webpages.

---

3. Basic Syntax and Structure of CSS --

Definition

CSS syntax defines how styles are written and applied to HTML elements. A CSS rule consists of a **selector** and a **declaration block**.

Basic Syntax

```
selector {
    property: value;
}
```

Components

1. Selector

Identifies the HTML element to style.

Examples:

```
p { }
h1 { }
.classname { }
#idname { }
```

2. Property

Defines the style feature to change.

Examples:

- color
- font-size
- margin
- background

3. Value

Specifies the setting of a property.

Example:

```
color: blue;
```

CSS Structure

- Selectors target elements.
- Properties define styling.
- Values provide specific settings.

Example

```
h1 {
color:red;
font-size:30px;
}
```

Advantages

- Simple syntax.
- Easy to understand.
- Supports reusable styles.

Exam Definition:
CSS syntax consists of selectors, properties, and values that define how HTML elements are displayed.

---

4. Using CSS (Methods of Applying CSS) --

Definition

CSS can be applied to HTML documents using three different methods: inline, internal, and external stylesheets.

1. Inline CSS

CSS is written directly inside an HTML tag using the `style` attribute.

Example:

```
<p style="color:red;">Text</p>
```

Advantages

- Quick styling.
- Useful for small changes.

Disadvantages

- Difficult to maintain.

---

2. Internal CSS

CSS is written inside the `<style>` element in the HTML `<head>` section.

Advantages

- Good for single-page websites.

---

3. External CSS

CSS is stored in a separate file and linked using:

```
<link rel="stylesheet" href="style.css">
```

Advantages

- Reusable.
- Faster maintenance.
- Best for large websites.

Exam Definition:
CSS can be used through inline, internal, or external methods to apply styles to HTML elements.

---

5. Background Images, Colors, and Properties --

Definition

CSS provides background properties to control the appearance of webpage backgrounds using colors and images.

Background Properties

1. background-color

Sets the background color.

Example:

```
body {
background-color:white;
}
```

2. background-image

Adds an image as a background.

Example:

```
body {
background-image:url("image.jpg");
}
```

3. background-repeat

Controls image repetition.

- repeat
- no-repeat
- repeat-x
- repeat-y

4. background-position

Controls image placement.

5. background-size

Controls image size.

Key Points

- Improves webpage appearance.
- Supports gradients and images.
- Allows customized layouts.
- Enhances visual design.

Advantages

- Creates attractive websites.
- Provides branding options.
- Improves user experience.

Exam Definition:
CSS background properties control webpage backgrounds using colors, images, positions, and repetition settings.

---

6. Manipulating Texts Using CSS --

Definition

CSS text properties are used to control the appearance, alignment, spacing, and decoration of text on webpages.

Important Text Properties

1. color

Changes text color.

2. text-align

Controls alignment:

- left
- right

- center
- justify

3. text-decoration

Adds decoration:

- underline
- overline
- line-through

4. text-transform

Changes text case:

- uppercase
- lowercase
- capitalize

5. letter-spacing

Controls space between characters.

6. line-height

Controls spacing between lines.

7. text-shadow

Adds shadow effects.

Importance

- Improves readability.
- Creates attractive headings.
- Maintains design consistency.
- Enhances user experience.

Exam Definition:
CSS text properties control the appearance, alignment, spacing, and decoration of webpage text.

---

7. Using Fonts in CSS --

Definition

CSS fonts control the style, size, and appearance of text on webpages. Proper font selection improves readability and website design.

Font Properties

1. font-family

Defines the typeface.

Examples:

- Arial
- Times New Roman
- Verdana

2. font-size

Controls text size.

3. font-style

Defines:

- normal
- italic
- oblique

4. font-weight

Controls thickness:

- normal
- bold

5. font-variant

Controls small capital letters.

Web Fonts

CSS supports external fonts through:

- Google Fonts

- @font-face

Advantages

- Improves readability.
- Creates professional appearance.
- Supports branding.
- Provides design flexibility.

Exam Definition:
CSS font properties control the style, size, weight, and appearance of text displayed on webpages.

---

8. Borders and Boxes in CSS --

Definition

CSS borders and box models are used to design the structure and appearance of HTML elements. Every HTML element is treated as a rectangular box.

CSS Box Model Components

1. **Content**

- Actual text or image.

2. **Padding**

- Space between content and border.

3. **Border**

- Surrounds the element.

4. **Margin**

- Space outside the border.

Border Properties

- border-width
- border-style
- border-color

- border-radius

Border Styles

- solid
- dotted
- dashed
- double

Importance

- Creates attractive layouts.
- Controls element spacing.
- Improves webpage structure.

Exam Definition:
The CSS box model describes the structure of HTML elements using content, padding, border, and margin.

---

9. Margins and Padding in CSS --

Definition

Margins and padding control spacing around HTML elements. They help create proper layouts and improve readability.

Margin

Margin is the space outside an element's border.

Properties:

- margin-top
- margin-right
- margin-bottom
- margin-left

Example:

```
div {
margin:20px;
}
```

Padding

Padding is the space between content and border.

Properties:

- padding-top
- padding-right
- padding-bottom
- padding-left

Example:

```
div {
padding:15px;
}
```

Difference

Margin	Padding
Outside border	Inside border
Controls distance between elements	Controls content spacing

Advantages

- Creates clean layouts.
- Improves readability.
- Controls element positioning.

Exam Definition:
Margins control outer spacing of elements, while padding controls inner spacing between content and border.

---

10. Lists Using CSS --

Definition

CSS list properties are used to customize the appearance and layout of HTML lists.

List Properties

1. list-style-type

Defines bullet style.

Examples:

- circle
- square
- decimal

2. list-style-image

Uses an image as a bullet.

3. list-style-position

Controls bullet position:

- inside
- outside

Key Points

- Improves list appearance.
- Creates custom navigation menus.
- Controls bullet design.
- Supports horizontal lists.

Example Applications

- Navigation bars.
- Product features.
- Menus.

Exam Definition:
CSS list properties control the style, position, and appearance of ordered and unordered lists.

---

11. Positioning Using CSS --

Definition

CSS positioning controls how HTML elements are placed on a webpage. It determines the location of elements relative to their normal position or other elements.

Types of Positioning

1. Static Position

- Default position.
- Elements appear normally.

2. Relative Position

- Moves element relative to its original position.

3. Absolute Position

- Positioned relative to the nearest positioned parent.

4. Fixed Position

- Element remains fixed during scrolling.

5. Sticky Position

- Element behaves normally until a position is reached, then becomes fixed.

Important Properties

- top
- bottom
- left
- right
- z-index

Advantages

- Creates advanced layouts.
- Supports responsive design.
- Controls element placement.

Exam Definition:
CSS positioning defines the location and arrangement of webpage elements using different positioning methods.

---

12. CSS2 --

Definition

CSS2 (Cascading Style Sheets Level 2) is the second version of CSS introduced by W3C. It added advanced styling features and improved control over webpage layouts.

Key Features

- Support for media types.
- Improved positioning methods.
- Layering using z-index.
- Better font handling.
- Improved selectors.
- Support for absolute and relative positioning.
- Introduction of media queries concept.

Advantages

- More flexible layouts.
- Better control over design.
- Improved browser support.
- Enhanced webpage presentation.

Limitations

- Limited responsive design capabilities.
- Less support for animations compared to CSS3.

Exam Definition:
CSS2 is a stylesheet standard that introduced advanced styling features such as positioning, media support, and improved selectors.

---

13. Latest Version of CSS – CSS3 --

Definition

CSS3 is the latest major version of CSS that provides advanced styling features for modern web development. It introduced powerful tools for animations, responsive design, and visual effects.

Key Features

1. Advanced Selectors

- Allows precise element selection.

2. Media Queries

- Supports responsive websites for different devices.

3. Animations and Transitions

- Creates movement and visual effects.

4. Transformations

- Supports rotation, scaling, and movement.

5. Flexbox

- Provides flexible layouts.

6. Grid Layout

- Creates advanced two-dimensional layouts.

7. Rounded Corners

- Using `border-radius`.

8. Shadows

- Text and box shadows.

9. Gradients

- Creates smooth color transitions.

Advantages

- Better user interface design.
- Mobile-friendly websites.
- Less dependency on images and plugins.
- Improved performance.

Exam Definition:
CSS3 is the modern CSS standard that provides responsive layouts, animations, transformations, and advanced styling features.

Quick Revision Table

Topic	Key Idea
Need for CSS	Separates webpage content from design.
Introduction to CSS	Language used for styling HTML pages.
CSS Syntax	Uses selectors, properties, and values.
Using CSS	Inline, internal, and external methods.
Background Properties	Controls colors and images.
Text Manipulation	Controls text appearance and alignment.
Fonts	Defines font style, size, and weight.
Borders & Boxes	Uses CSS box model for layouts.
Margins & Padding	Controls element spacing.
Lists	Customizes list appearance.
Positioning	Controls element placement.
CSS2	Added advanced positioning and styling.
CSS3	Provides modern layouts, animations, and responsive design.

Unit-3

JavaScript – Exam Notes

---

1. Client-Side Scripting with JavaScript --

Definition

JavaScript is a high-level, interpreted programming language used to create dynamic and interactive web pages. It is mainly used for **client-side scripting**, where scripts are executed directly in the user's web browser without sending every request to the server.

Key Points

- JavaScript runs inside web browsers such as Chrome, Firefox, and Edge.
- It works with HTML and CSS to create interactive websites.

- It improves user experience by providing instant responses.
- It can modify HTML content and CSS styles dynamically.
- It is used for form validation, animations, menus, and interactive effects.
- JavaScript code can be embedded inside HTML using the `<script>` tag.

Advantages

- Faster execution because code runs on the client side.
- Reduces server workload.
- Provides interactive web applications.
- Supports event-driven programming.
- Works on multiple platforms.

Applications

- Online forms.
- Web games.
- Image sliders.
- Real-time updates.
- Interactive dashboards.

Exam Definition:
Client-side JavaScript is a scripting technology that executes code in the user's browser to create dynamic and interactive web pages.

---

2. JavaScript Variables --

Definition

A variable in JavaScript is a named storage location used to store data values. Variables allow programs to store, modify, and retrieve information during execution.

Declaring Variables

JavaScript provides three keywords:

1. var

- Older method of variable declaration.
- Function-scoped variable.

Example:

```
var name = "John";
```

2. let

- Introduced in ES6.
- Block-scoped variable.
- Value can be changed.

Example:

```
let age = 20;
```

3. const

- Used for constant values.
- Value cannot be reassigned.

Example:

```
const pi = 3.14;
```

Data Types

- Number
- String
- Boolean
- Object
- Array
- Undefined
- Null

Key Points

- Variables store data temporarily.
- Variable names are case-sensitive.
- Names should be meaningful.
- JavaScript is dynamically typed.

Exam Definition:
Variables are containers used in JavaScript programs to store and manipulate data values during execution.

---

3. JavaScript Functions --

Definition

A function is a reusable block of JavaScript code designed to perform a specific task. Functions help reduce code repetition and improve program organization.

Syntax

```
function functionName(){
   // code
}
```

Types of Functions

1. Built-in Functions

Provided by JavaScript.

Examples:

- alert()
- prompt()
- parseInt()

2. User-defined Functions

Created by programmers according to requirements.

Example:

```
function add(a,b){
 return a+b;
}
```

Function Components

- Function name.
- Parameters.
- Function body.
- Return value.

Advantages

- Code reusability.
- Easier debugging.
- Better program structure.
- Reduces complexity.

Types of Function Execution

- Function call.
- Event-based execution.
- Immediate execution.

Exam Definition:
A JavaScript function is a reusable set of statements that performs a specific operation when called.

---

4. JavaScript Conditions --

Definition

Conditional statements in JavaScript are used to make decisions in programs. They execute different blocks of code based on whether a condition is true or false.

Types of Conditional Statements

1. if Statement

Executes code when a condition is true.

Example:

```
if(age>=18){
 console.log("Adult");
}
```

2. if...else Statement

Executes one block if true and another if false.

3. else if Statement

Used when multiple conditions are checked.

4. switch Statement

Selects one option from multiple choices.

Example:

```
switch(day){
case 1:
console.log("Monday");
}
```

Operators Used

- Comparison operators: `==`, `===`, `>`, `<`
- Logical operators: `&&`, `||`, `!`

Importance

- Controls program flow.
- Helps decision-making.
- Creates dynamic applications.

Exam Definition:
Conditional statements in JavaScript control the execution of code based on specified conditions.

---

5. JavaScript Loops and Repetition --

Definition

Loops are programming structures used to execute a block of code repeatedly until a specific condition becomes false. They reduce repeated code and improve efficiency.

Types of Loops

1. for Loop

Used when the number of repetitions is known.

Example:

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

2. while Loop

Executes as long as a condition remains true.

3. do...while Loop

Executes code at least once before checking the condition.

4. for...in Loop

Used to iterate through object properties.

5. for...of Loop

Used to iterate through iterable objects like arrays.

Loop Control Statements

- **break:** Stops loop execution.
- **continue:** Skips current iteration.

Advantages

- Reduces code duplication.
- Improves efficiency.
- Handles repetitive tasks easily.

Exam Definition:
Loops are JavaScript control structures used to repeat a set of instructions multiple times based on a condition.

---

6. JavaScript Pop-up Boxes --

Definition

Popup boxes are JavaScript dialog boxes used to display messages, collect user input, or request confirmation. They provide interaction between users and web applications.

Types of Popup Boxes

1. Alert Box

Displays information or warning messages.

Syntax:

```
alert("Welcome");
```

Example Uses:

- Error messages.
- Notifications.

---

2. Confirm Box

Used to get user confirmation.

Syntax:

```
confirm("Are you sure?");
```

Returns:

- OK → true
- Cancel → false

---

3. Prompt Box

Used to take input from users.

Syntax:

```
prompt("Enter name");
```

Advantages

- Simple user interaction.
- Provides instant feedback.
- Useful for collecting small inputs.

Applications

- Login messages.
- Confirming actions.
- User data collection.

Exam Definition:
JavaScript popup boxes are dialog windows used to display messages and collect user responses.

---

7. Advanced JavaScript: JavaScript and Objects --

Definition

JavaScript is an object-oriented programming language where objects are used to store data and functions together. Objects represent real-world entities and contain properties and methods.

Object Structure

An object contains:

1. Properties

- Store information about an object.

Example:

```
student.name="John";
```

2. Methods

- Functions associated with an object.

Example:

```
student.display();
```

Creating Objects

Object Literal Method:

```
let person={
name:"John",
age:25
};
```

Key Points

- Objects organize related data.
- Properties describe objects.
- Methods define object behavior.
- Supports inheritance and encapsulation.
- Used extensively in modern web development.

Examples of Objects

- User objects.
- Product objects.
- Date objects.
- Array objects.

Exam Definition:
JavaScript objects are collections of properties and methods that represent data and behavior in programs.

---

8. JavaScript Own Objects (Built-in Objects) --

Definition

JavaScript provides several predefined objects called built-in objects. These objects contain ready-made properties and methods that help programmers perform common tasks.

Important JavaScript Built-in Objects

1. String Object

Used for handling text.

Methods:

- toUpperCase()
- toLowerCase()
- charAt()

2. Array Object

Stores multiple values.

Methods:

- push()
- pop()
- sort()

3. Math Object

Performs mathematical operations.

Methods:

- random()
- sqrt()
- round()

4. Date Object

Works with date and time.

Methods:

- getDate()
- getTime()

5. Number Object

Handles numerical values.

6. RegExp Object

Used for pattern matching.

Advantages

- Saves programming time.
- Provides reusable functions.
- Simplifies complex operations.
- Improves code efficiency.

Exam Definition:
JavaScript built-in objects are predefined objects that provide properties and methods for performing common programming tasks.

9. JavaScript Object-Oriented Features --

Definition

JavaScript supports object-oriented programming (OOP) concepts that help create organized and reusable code. Objects combine data and functions into a single structure.

Main OOP Concepts

1. Encapsulation

- Combines data and methods inside objects.
- Protects data from unauthorized access.

2. Inheritance

- Allows one object to use properties and methods of another object.

3. Polymorphism

- Allows the same method to perform different actions.

4. Abstraction

- Hides unnecessary details and shows only required information.

Benefits

- Code reusability.
- Easier maintenance.
- Better organization.
- Improved scalability.

Applications

- Web applications.
- Games.
- User interface development.
- Complex software systems.

Exam Definition:
JavaScript object-oriented programming uses objects, inheritance, and reusable methods to create structured and maintainable applications.

Quick Revision Table

Topic	Key Idea
Client-Side JavaScript	Runs in browser to create interactive webpages.
Variables	Store and manage data values.
Functions	Reusable blocks of code for specific tasks.
Conditions	Control program execution based on decisions.
Loops	Repeat code execution efficiently.
Popup Boxes	Provide alerts and user interaction.
Objects	Store data and functions together.
Built-in Objects	Predefined JavaScript objects with useful methods.
OOP Features	Encapsulation, inheritance, polymorphism, abstraction.

Important Exam Points

- JavaScript is a **client-side scripting language** used with HTML and CSS.
- Variables are declared using **var, let, and const**.
- Functions improve **code reuse and organization**.
- Conditions include **if, if-else, switch**.
- Loops include **for, while, do-while**.
- Popup boxes include **alert(), confirm(), and prompt()**.
- JavaScript objects contain **properties and methods**.
- Built-in objects include **String, Array, Math, Date, and RegExp**.

DOM, DHTML, Forms, Events and Browser Environment – Exam Notes

---

1. The DOM and Web Browser Environments --

Definition

DOM (Document Object Model) is a programming interface that represents an HTML or XML document as a tree structure of objects. It allows JavaScript to access, modify, add, or delete webpage elements dynamically.

DOM Structure

A webpage is represented as a tree containing:

- **Document Node:** Represents the entire webpage.
- **Element Nodes:** Represent HTML tags like `<p>`, `<div>`, `<h1>`.
- **Attribute Nodes:** Represent attributes like id, class, and href.
- **Text Nodes:** Represent text content inside elements.

Web Browser Environment

A web browser provides an environment where HTML, CSS, and JavaScript execute together.

Browser Components

- **Rendering Engine:** Displays HTML and CSS.
- **JavaScript Engine:** Executes JavaScript code.
- **DOM Interface:** Allows scripts to interact with webpage content.
- **Event System:** Handles user actions.

Importance

- Creates dynamic webpages.
- Allows real-time content changes.
- Provides interaction between users and websites.

Exam Definition:
DOM is a programming model that represents webpage elements as objects, allowing JavaScript to manipulate web content dynamically.

---

2. Manipulation Using DOM --

Definition

DOM manipulation is the process of using JavaScript to change the structure, content, and style of a webpage after it has loaded in the browser.

Common DOM Methods

1. Selecting Elements

- `getElementById()`
- `getElementsByClassName()`
- `querySelector()`
- `querySelectorAll()`

Example:

```
document.getElementById("title");
```

2. Changing Content

- `innerHTML` – Changes HTML content.
- `textContent` – Changes text.

3. Changing Styles

Example:

```
element.style.color="red";
```

4. Creating and Removing Elements

- `createElement()`
- `appendChild()`
- `removeChild()`

5. Changing Attributes

- `setAttribute()`
- `getAttribute()`

Advantages

- Creates interactive websites.
- Updates content without reloading pages.
- Improves user experience.
- Supports dynamic applications.

Applications

- Changing images.
- Updating menus.
- Creating animations.
- Modifying webpage content.

Exam Definition:
DOM manipulation is the technique of using JavaScript to dynamically modify HTML elements, attributes, and styles.

---

3. Forms and Validations in JavaScript --

Definition

HTML forms are used to collect information from users. JavaScript validation checks whether the entered data is correct before sending it to the server.

Form Elements

- Text fields.
- Password fields.
- Radio buttons.
- Checkboxes.
- Drop-down lists.
- Submit buttons.

Types of Validation

1. Client-Side Validation

- Performed in the browser using JavaScript.
- Provides quick feedback.
- Reduces unnecessary server requests.

2. Server-Side Validation

- Performed on the server.
- Provides better security.

Common Validation Checks

- Required fields.
- Correct email format.
- Password length.
- Numeric values.
- Valid input ranges.

Example:

```
if(email==""){
alert("Enter email");
}
```

Advantages

- Improves data accuracy.
- Enhances user experience.
- Prevents incorrect submissions.
- Saves server processing time.

Exam Definition:
JavaScript form validation is the process of checking user input before submission to ensure data correctness and completeness.

---

4. DHTML: Combining HTML, CSS and JavaScript --

Definition

DHTML (Dynamic HyperText Markup Language) is a combination of HTML, CSS, JavaScript, and DOM used to create interactive and dynamic webpages. DHTML allows webpage content and styles to change without loading a new page.

Components of DHTML

1. HTML

- Provides webpage structure.
- Defines elements and content.

2. CSS

- Controls appearance.
- Defines colors, layouts, and styles.

3. JavaScript

- Provides functionality.
- Handles user interaction and dynamic changes.

4. DOM

- Connects HTML elements with JavaScript.
- Allows modification of webpage elements.

Features

- Dynamic content updates.
- Interactive menus.
- Animations.
- Real-time style changes.
- Improved user experience.

Advantages

- Faster webpage interaction.
- Reduced server communication.
- Creates attractive interfaces.
- Supports modern web applications.

Example Applications

- Dropdown menus.
- Image sliders.
- Interactive forms.
- Online games.

Exam Definition:
DHTML is a web development approach that combines HTML, CSS, JavaScript, and DOM to create dynamic and interactive webpages.

---

5. Events in JavaScript --

Definition

An event is an action or occurrence that happens in a webpage and can be detected and handled by JavaScript. Events allow webpages to respond to user actions.

Types of Events

1. Mouse Events

Actions performed using a mouse:

- onclick
- ondblclick
- onmouseover
- onmouseout

2. Keyboard Events

Actions performed using keyboard:

- onkeydown
- onkeyup
- onkeypress

3. Form Events

Related to forms:

- onsubmit
- onchange
- onfocus
- onblur

4. Window Events

Related to browser window:

- onload
- onresize
- onscroll

Event Handling

JavaScript uses event handlers to execute code when an event occurs.

Example:

```
button.onclick=function(){
alert("Clicked");
}
```

Importance

- Provides user interaction.
- Creates dynamic webpages.
- Improves usability.
- Enables real-time responses.

Exam Definition:
JavaScript events are actions performed by users or browsers that trigger specific code execution.

---

6. Buttons in JavaScript --

Definition

Buttons are interactive HTML elements that allow users to perform actions on webpages. JavaScript can control button behavior using events and event handlers.

Creating Buttons

HTML:

```
<button>Click Me</button>
```

Button Events

- onclick – Executes when button is clicked.
- onmouseover – Executes when mouse moves over button.
- onmousedown – Executes when mouse button is pressed.
- onmouseup – Executes when mouse button is released.

JavaScript Button Operations

Buttons can be used to:

- Display messages.
- Submit forms.
- Change webpage content.
- Start animations.
- Perform calculations.

Example:

```
function showMessage(){
alert("Hello User");
}
```

Advantages

- Improves user interaction.
- Provides easy control.

- Supports dynamic actions.
- Enhances webpage functionality.

Applications

- Login buttons.
- Submit buttons.
- Navigation controls.
- Interactive tools.

Exam Definition:
JavaScript buttons are interactive webpage elements that execute specific actions when users perform events such as clicks.

---

7. DOM Events and Event Handling --

Definition

DOM events allow JavaScript programs to respond to changes and user interactions within webpage elements. Event handling connects an event with a specific action.

Event Handling Methods

1. Inline Event Handling

Event code is written inside HTML.

Example:

```
<button onclick="alert('Hi')">
Click
</button>
```

2. Event Handler Property

Events are assigned using JavaScript.

Example:

```
button.onclick=function(){};
```

3. addEventListener()

Modern method for attaching events.

Example:

```
button.addEventListener("click",function(){});
```

Event Flow

- **Capturing Phase:** Event moves from document to target element.
- **Target Phase:** Event reaches the selected element.
- **Bubbling Phase:** Event moves back upward.

Benefits

- Better control of user actions.
- Supports interactive applications.
- Separates HTML and JavaScript code.

Exam Definition:
DOM event handling is the process of responding to user actions by executing JavaScript functions connected to webpage events.

Quick Revision Table

Topic	Key Idea
DOM	Represents webpage as objects for JavaScript manipulation.
Browser Environment	Provides execution environment for HTML, CSS, and JavaScript.
DOM Manipulation	Changes webpage content, structure, and style dynamically.
Forms Validation	Checks user input before submission.
DHTML	Combination of HTML, CSS, JavaScript, and DOM for dynamic pages.
Events	Actions that trigger JavaScript code execution.
Buttons	Interactive elements controlled using JavaScript events.

Important Exam Points

- DOM represents HTML documents in a **tree structure**.
- JavaScript uses DOM methods like **getElementById()** and **querySelector()**.
- Form validation improves **data accuracy and user experience**.
- DHTML combines **HTML + CSS + JavaScript + DOM**.
- Events include **mouse, keyboard, form, and window events**.

- Buttons use event handlers such as **onclick** to perform actions.
- Modern event handling uses **addEventListener()**.

XML (Extensible Markup Language) – Exam Notes

---

1. Introduction to XML --

Definition

XML (Extensible Markup Language) is a markup language used to store, organize, and transport data in a structured format. It was developed by the **World Wide Web Consortium (W3C)** in 1998. Unlike HTML, which focuses on displaying data, XML focuses on describing and storing data.

Key Points

- XML uses user-defined tags.
- It is platform-independent and human-readable.
- XML separates data from presentation.
- It supports data exchange between different applications.
- XML documents follow a tree-like structure.
- It is case-sensitive.
- Every XML document must have a single root element.

Features

- Simple and flexible.
- Supports Unicode characters.
- Self-descriptive data format.
- Easy data sharing between systems.
- Extensible because users can create their own tags.

Applications

- Web services.
- Data storage.
- Configuration files.
- Database communication.
- Data exchange between organizations.

Exam Definition:
XML is a text-based markup language used to store and exchange structured data between different applications and systems.

---

2. Uses of XML --

Definition

XML is widely used for representing and exchanging structured information between different software applications. It provides a common format for communication because it is independent of hardware, operating systems, and programming languages.

Major Uses

1. Data Exchange

- XML allows different systems to exchange data easily.
- Used in web services and APIs.

2. Data Storage

- Stores structured information in files.
- Used for configuration and document storage.

3. Web Applications

- Transfers data between client and server.
- Supports AJAX applications.

4. Database Communication

- Converts database information into a standard format.

5. Configuration Files

- Used by software applications to store settings.

6. Document Management

- Used for electronic documents and publishing systems.

Advantages

- Platform independent.
- Easy data sharing.
- Supports complex data structures.
- Human and machine readable.

Exam Definition:
XML is used for data storage, data exchange, configuration management, and communication between different applications.

---

3. Simple XML Document --

Definition

A simple XML document contains data organized using user-defined tags. XML tags describe the meaning of data rather than controlling its appearance.

Example:

```
<?xml version="1.0"?>

<student>
    <name>John</name>
    <rollno>101</rollno>
    <course>Computer Science</course>
</student>
```

Components of XML Document

1. XML Declaration

Defines XML version and encoding.

Example:

```
<?xml version="1.0"?>
```

2. Root Element

- The main element containing all other elements.
- Every XML document must have one root element.

3. Child Elements

- Elements inside the root element.

- Store actual data.

Rules for XML Documents

- Tags must be properly closed.
- Tags are case-sensitive.
- Elements must be properly nested.
- Attribute values must be enclosed in quotes.

Importance

- Provides structured data representation.
- Allows easy data processing.
- Supports data sharing.

Exam Definition:
A simple XML document is a structured file containing user-defined tags that represent and organize data.

---

4. XML Key Components --

Definition

XML documents consist of several components that define their structure and meaning. These components help organize and validate XML data.

Main Components

1. XML Declaration

- Specifies XML version and encoding.
- Appears at the beginning of a document.

2. Elements

- Basic building blocks of XML.
- Represent data using tags.

Example:

```
<name>David</name>
```

3. Attributes

- Provide additional information about elements.

Example:

```
<student id="101">
```

4. Text Content

- Actual data stored inside elements.

5. Comments

- Provide explanations within XML documents.

Example:

```
<!-- Student information -->
```

6. Entity References

- Represent special characters.

Example:

- `<` for <

7. Processing Instructions

- Provide instructions to applications processing XML.

Importance

These components help create structured, readable, and meaningful XML documents.

Exam Definition:
XML key components include declarations, elements, attributes, text, comments, and entities used to structure XML data.

---

5. DTD (Document Type Definition) --

Definition

DTD is a set of rules that defines the structure and allowed elements of an XML document. It is used to validate whether an XML document follows a specific format.

Types of DTD

1. Internal DTD

- Defined inside the XML document.

Example:

```
<!DOCTYPE student [
<!ELEMENT student (name)>
]>
```

2. External DTD

- Stored in a separate file.
- Can be reused by multiple XML documents.

Functions of DTD

- Defines valid elements.
- Defines attributes.
- Specifies element relationships.
- Ensures data consistency.

Advantages

- Provides document validation.
- Maintains data structure.
- Improves data reliability.

Limitations

- Limited data type support.
- Syntax is less flexible than XML Schema.

Exam Definition:
DTD is a method used to define and validate the structure and rules of an XML document.

---

6. XML Schema --

Definition

XML Schema, also called **XSD (XML Schema Definition)**, is a more powerful alternative to DTD used for defining the structure and validating XML documents.

Key Features

- Written using XML syntax.
- Supports different data types.
- Defines complex data structures.
- Supports namespaces.
- Provides better validation.

XML Schema Defines

- Elements.
- Attributes.
- Data types.
- Number of occurrences.
- Relationships between elements.

Advantages Over DTD

DTD	XML Schema
Limited data types	Supports many data types
Not written in XML	Written in XML
Less flexible	More powerful

Example Data Types

- String.
- Integer.
- Date.
- Boolean.

Importance

- Ensures accurate data.
- Improves XML document reliability.
- Used in enterprise applications.

Exam Definition:
XML Schema is an XML-based language used to define and validate the structure, data types, and constraints of XML documents.

7. Using XML with Applications --

Definition

XML is used with various applications to store, exchange, and process data. It acts as a common communication format between different systems.

Applications of XML

1. Web Services

- XML is used for communication between applications.
- Example: SOAP web services.

2. Database Applications

- Transfers data between databases and applications.

3. Mobile Applications

- Stores configuration and application data.

4. Office Applications

- Used in document formats such as DOCX and XLSX.

5. E-Commerce

- Exchanges product and transaction information.

6. Configuration Management

- Stores software settings.

Advantages

- Platform independent.
- Easy data exchange.
- Supports complex information.
- Compatible with multiple programming languages.

Example

A banking application can use XML to exchange customer information securely between different systems.

Exam Definition:
XML is used with applications as a standard format for storing, sharing, and transferring structured data.

---

8. Transforming XML using XSL --

Definition

XSL (Extensible Stylesheet Language) is a stylesheet language used to control the presentation and transformation of XML documents. It separates XML data from its display format.

Components of XSL

1. XSLT

- Performs transformation of XML documents into other formats.

2. XPath

- Selects specific data from XML documents.

3. XSL-FO

- Formats XML documents for printing.

Uses of XSL

- Converts XML into HTML.
- Creates different views of the same XML data.
- Formats XML documents.

Advantages

- Separates data and presentation.
- Allows flexible output formats.
- Supports reusable stylesheets.
- Improves data presentation.

Example

An XML file containing student records can be transformed into an HTML table using XSL.

Exam Definition:
XSL is a stylesheet technology used to transform and format XML documents for different output presentations.

---

9. XSLT (Extensible Stylesheet Language Transformations) --

Definition

XSLT is a language used to transform XML documents into other formats such as HTML, XML, or plain text. It uses templates and rules to process XML data.

Key Components

1. Template Rules

- Define how XML elements should be transformed.

2. XPath Expressions

- Select XML elements for processing.

3. Output Method

- Defines output format.

XSLT Process

1. XML document is loaded.
2. XSLT processor reads stylesheet rules.
3. XPath selects required elements.
4. Output document is generated.

Advantages

- Converts XML into user-friendly formats.
- Separates data from presentation.
- Supports reusable transformations.
- Works across platforms.

Applications

- XML to HTML conversion.
- Report generation.
- Data integration.
- Web publishing.

Exam Definition:
XSLT is an XML transformation language used to convert XML documents into different formats using templates and XPath.

Quick Revision Table

Topic	Key Idea
XML Introduction	Language for storing and exchanging structured data.
Uses of XML	Data exchange, storage, configuration, and web services.
Simple XML	Document containing user-defined tags and structured data.
XML Components	Declaration, elements, attributes, comments, entities.
DTD	Defines XML document structure and validation rules.
XML Schema	Advanced XML validation using data types and constraints.
XML Applications	Used in web services, databases, and data communication.
XSL	Stylesheet language for XML presentation.
XSLT	Converts XML documents into HTML, XML, or text formats.

Important Exam Points

- XML is **extensible, platform-independent, and self-descriptive**.
- XML stores data, while HTML displays data.
- DTD and XML Schema are used for **XML validation**.
- XML Schema supports **data types and namespaces**.
- XSL separates **XML data from presentation**.
- XSLT transforms XML documents using **templates and XPath**.
- XML is widely used in **web services, databases, and application communication**.

Unit-4

PHP – Exam Notes

---

1. Introduction and Basic Syntax of PHP --

Definition

PHP (Hypertext Preprocessor) is a server-side scripting language used for developing dynamic and interactive websites. It was created by **Rasmus Lerdorf in 1994**. PHP code runs on the web server and generates HTML output that is sent to the user's browser.

Key Points

- PHP is an open-source scripting language.
- PHP files use the extension **.php**.
- It can be embedded inside HTML code.
- PHP supports databases such as MySQL, PostgreSQL, and Oracle.
- It works on different operating systems.
- PHP is mainly used for web application development.

Basic Syntax

PHP code is written inside:

```
<?php
echo "Hello World";
?>
```

Features

- Easy to learn.
- Fast execution.
- Supports object-oriented programming.
- Provides database connectivity.
- Platform independent.

Applications

- Content management systems.
- E-commerce websites.
- Login systems.
- Web portals.

Exam Definition:
PHP is a server-side scripting language used to create dynamic web pages and web applications.

---

2. Decision Making in PHP --

Definition

Decision-making statements in PHP allow programs to execute different blocks of code based on conditions. They control the flow of program execution.

Types of Decision Statements

1. if Statement

Executes code when a condition is true.

Example:

```
<?php
$age=20;

if($age>=18)
{
echo "Eligible";
}
?>
```

2. if...else Statement

Executes one block if the condition is true and another if false.

3. if...elseif...else

Used when multiple conditions are checked.

4. switch Statement

Used to select one option from multiple choices.

Example:

```
switch($day)
{
```

```
case 1:
echo "Monday";
break;
}
```

Operators Used

- Comparison operators: `==`, `!=`, `>`, `<`
- Logical operators: `&&`, `||`, `!`

Importance

- Controls program execution.
- Helps create dynamic applications.
- Allows logical decision-making.

Exam Definition:
Decision statements in PHP control program flow by executing code according to specified conditions.

---

3. Looping in PHP --

Definition

Loops are used to execute a block of code repeatedly until a given condition becomes false. They reduce repeated programming statements.

Types of PHP Loops

1. for Loop

Used when the number of repetitions is known.

Example:

```
for($i=1;$i<=5;$i++)
{
echo $i;
}
```

2. while Loop

Executes while a condition remains true.

Example:

```
while($x<=10)
{
echo $x;
$x++;
}
```

3. do...while Loop

Executes the code at least once before checking the condition.

4. foreach Loop

Used to access elements of arrays.

Example:

```
foreach($name as $value)
{
echo $value;
}
```

Loop Control Statements

- **break:** Terminates the loop.
- **continue:** Skips current iteration.

Advantages

- Reduces code repetition.
- Improves efficiency.
- Handles repeated tasks easily.

Exam Definition:
Loops in PHP are control structures used to execute statements repeatedly based on conditions.

---

4. PHP and HTML --

Definition

PHP and HTML are commonly used together for developing dynamic websites. HTML creates the structure of webpages, while PHP adds programming logic and dynamic content.

Working

- HTML provides the user interface.
- PHP processes data on the server.
- PHP generates HTML output dynamically.
- Browser displays the generated webpage.

Example:

```
<html>
<body>

<?php
echo "Welcome to PHP";
?>

</body>
</html>
```

Uses of PHP with HTML

- Creating login pages.
- Processing forms.
- Displaying database records.
- Generating dynamic content.
- Creating customized webpages.

Advantages

- Interactive websites.
- Better user experience.
- Database connectivity.
- Dynamic page generation.

Difference

HTML	PHP
Client-side language	Server-side language
Creates webpage structure	Adds programming logic

Exam Definition:
PHP works with HTML to create dynamic webpages by generating HTML content from server-side scripts.

---

5. Arrays in PHP --

Definition

An array is a data structure used to store multiple values in a single variable. Arrays help organize and manage related data efficiently.

Types of Arrays in PHP

1. Indexed Array

Stores values using numeric indexes.

Example:

```
$colors=array("Red","Blue","Green");
```

2. Associative Array

Stores values using named keys.

Example:

```
$age=array("John"=>20);
```

3. Multidimensional Array

Contains one or more arrays inside another array.

Common Array Functions

- count() – Counts elements.
- sort() – Sorts array.
- array_push() – Adds elements.
- array_pop() – Removes elements.

Advantages

- Stores multiple values.
- Easy data management.
- Reduces code complexity.
- Supports loops.

Applications

- Student records.
- Product lists.
- Database results.

Exam Definition:
Arrays in PHP are variables that store multiple values under a single name.

---

6. Functions in PHP --

Definition

A function is a reusable block of code designed to perform a specific task. Functions improve program organization and reduce repetition.

Syntax:

```
function functionName()
{
 //code
}
```

Types of Functions

1. Built-in Functions

Provided by PHP.

Examples:

- strlen()
- date()
- count()

2. User-defined Functions

Created by programmers.

Example:

```
function add($a,$b)
{
```

```
return $a+$b;
}
```

Function Components

- Function name.
- Parameters.
- Function body.
- Return value.

Advantages

- Code reusability.
- Easier debugging.
- Better program structure.
- Reduces development time.

Applications

- Data processing.
- Calculations.
- Database operations.

Exam Definition:
A PHP function is a reusable set of statements that performs a specific operation when called.

---

7. Browser Control and Detection in PHP --

Definition

Browser detection is the process of identifying information about the user's web browser and operating system. PHP can control webpage behavior according to browser details.

Browser Information

PHP uses the `$_SERVER` variable to access browser information.

Example:

```
echo $_SERVER['HTTP_USER_AGENT'];
```

Uses

- Detect browser type.
- Provide browser-specific content.
- Improve compatibility.
- Track user environment.

Browser Control Techniques

- Redirect users to different pages.
- Display different layouts.
- Enable or disable features.

Example Applications

- Mobile browser detection.
- Providing optimized pages.
- Browser compatibility handling.

Advantages

- Better user experience.
- Improved website performance.
- Supports different devices.

Exam Definition:
Browser detection in PHP identifies user browser information and allows websites to provide customized responses.

---

8. Strings in PHP --

Definition

A string is a sequence of characters used to store text information in PHP. Strings can contain letters, numbers, and special characters.

Creating Strings

Using single quotes:

```
$name='PHP';
```

Using double quotes:

```
$name="PHP";
```

Common String Functions

- **strlen()** – Finds string length.
- **strtoupper()** – Converts to uppercase.
- **strtolower()** – Converts to lowercase.
- **strpos()** – Finds position of text.
- **substr()** – Extracts part of string.
- **str_replace()** – Replaces text.

String Operations

- Concatenation using dot (.).
- Comparison.
- Searching.
- Formatting.

Applications

- User names.
- Messages.
- Data processing.
- Text manipulation.

Exam Definition:
Strings in PHP are collections of characters used for storing and processing textual data.

---

9. Form Processing in PHP --

Definition

Form processing is the method of collecting user input from HTML forms and processing it using PHP scripts. It is commonly used for login, registration, and feedback systems.

HTML Form Methods

1. GET Method

- Sends data through URL.
- Suitable for non-sensitive data.

2. POST Method

- Sends data securely in request body.
- Used for passwords and large data.

PHP Form Variables

- `$_GET`
- `$_POST`

Example:

```
$name=$_POST["name"];
echo $name;
```

Processing Steps

1. Create HTML form.
2. Submit user data.
3. Receive data in PHP.
4. Validate input.
5. Store/process information.

Advantages

- Collects user information.
- Provides interaction.
- Supports database storage.
- Enables dynamic websites.

Exam Definition:
PHP form processing handles user input received from HTML forms and performs required operations.

---

10. Files in PHP --

Definition

PHP provides file handling features to create, read, write, and manage files stored on a server. File handling is useful for storing information permanently.

File Operations

1. Opening Files

Function:

`fopen()`

2. Reading Files

Functions:

- fread()
- fgets()

3. Writing Files

Function:

- fwrite()

4. Closing Files

Function:

- fclose()

File Modes

- r – Read only.
- w – Write only.
- a – Append data.
- r+ – Read and write.

Applications

- Log files.
- Configuration files.
- Data storage.
- Reports.

Advantages

- Permanent data storage.
- Easy data management.
- Supports server-side processing.

Exam Definition:
PHP file handling allows programs to create, read, write, and manage files on a server.

11. Cookies in PHP --

Definition

Cookies are small text files stored on the user's computer by a website. They store user information and help maintain user preferences between visits.

Creating Cookies

Function:

```
setcookie("user","John");
```

Uses of Cookies

- Remember login information.
- Store user preferences.
- Track website usage.
- Personalize content.

Features

- Stored on client computer.
- Have expiration time.
- Sent with HTTP requests.
- Limited storage size.

Advantages

- Improves user experience.
- Reduces repeated input.
- Supports personalization.

Limitations

- Can be disabled by users.
- Security concerns.

Exam Definition:
Cookies are small data files stored on the user's device to maintain information between web sessions.

12. Sessions in PHP --

Definition

A session is a server-side mechanism used to store user information temporarily while a user interacts with a website. Sessions maintain data across multiple pages.

Starting a Session

```
session_start();
```

Storing Data

```
$_SESSION["user"]="John";
```

Uses

- User authentication.
- Shopping carts.
- Maintaining login status.
- Storing temporary information.

Session Process

1. User visits website.
2. Server creates session.
3. Session ID is stored.
4. Data is maintained until session ends.

Advantages

- More secure than cookies.
- Stores larger data.
- Server-side storage.
- Suitable for login systems.

Difference

Cookies	Sessions
Stored on client	Stored on server
Less secure	More secure
Limited size	Larger storage

Exam Definition:
A PHP session stores user information on the server to maintain user state across multiple webpage requests.

Quick Revision Table

Topic	Key Idea
PHP Basics	Server-side scripting language for dynamic websites.
Decision Making	Controls program flow using conditions.
Looping	Repeats code execution.
PHP with HTML	Combines server logic with webpage structure.
Arrays	Store multiple values in one variable.
Functions	Reusable blocks of code.
Browser Detection	Identifies user browser information.
Strings	Used for text processing.
Form Processing	Handles user input from forms.
Files	Provides server file management.
Cookies	Client-side storage of user information.
Sessions	Server-side storage for maintaining user state.

Important Exam Points:

- PHP is a **server-side scripting language**.
- PHP code is enclosed within `<?php ?>`.
- Decision statements include **if, if-else, switch**.
- Loops include **for, while, do-while, foreach**.
- Arrays store multiple values in a single variable.
- Forms use **GET and POST** methods.
- Cookies store data on the client side.
- Sessions store user data on the server side.

PHP and MySQL – Exam Notes

---

1. Introduction to PHP and MySQL --

Definition

PHP and MySQL are widely used together for developing dynamic and database-driven websites. PHP is a server-side scripting language used to process user requests, while MySQL is a relational database management system (RDBMS) used to store and manage data.

Key Points

- PHP communicates with MySQL using database connection functions.
- MySQL stores information in tables consisting of rows and columns.
- PHP retrieves, inserts, updates, and deletes data from MySQL databases.
- Used for web applications such as e-commerce, content management systems, and user login systems.

Advantages

- Open-source technologies.
- Easy integration.
- Fast performance.
- Secure data management.
- Supports large-scale applications.

Working Process

1. User sends a request through a webpage.
2. PHP processes the request.
3. PHP communicates with MySQL database.
4. Database returns required information.
5. PHP displays the result on the webpage.

Exam Definition:
PHP and MySQL together provide a platform for creating dynamic websites that store and process data efficiently.

---

2. Basic MySQL Commands with PHP Examples --

Definition

MySQL commands are SQL statements used to create, manage, and manipulate databases. PHP uses these commands to interact with MySQL.

Common SQL Commands

1. CREATE

Creates a database or table.

```
CREATE DATABASE college;
```

2. INSERT

Adds records into tables.

```
INSERT INTO student VALUES(1,'John');
```

3. SELECT

Retrieves data.

```
SELECT * FROM student;
```

4. UPDATE

Modifies existing data.

```
UPDATE student SET name='David';
```

5. DELETE

Removes data.

```
DELETE FROM student WHERE id=1;
```

PHP Example

```
$sql="SELECT * FROM student";
$result=mysqli_query($conn,$sql);
```

Importance

- Manages database operations.
- Allows dynamic website development.
- Supports data storage and retrieval.

Exam Definition:
MySQL commands are SQL statements executed through PHP to perform database operations such as creating, inserting, updating, and deleting data.

---

3. Connecting PHP to MySQL Server --

Definition

Connecting PHP to MySQL server allows PHP applications to communicate with databases and perform database operations.

Connection Methods

PHP provides:

- **MySQLi Extension**
- **PDO (PHP Data Objects)**

MySQLi Connection Example

```
$conn=mysqli_connect(
"localhost",
"root",
"",
"college"
);

if(!$conn)
{
die("Connection failed");
}
```

Connection Parameters

- **Server Name:** Location of MySQL server.
- **Username:** Database user name.
- **Password:** User password.
- **Database Name:** Database to connect.

Steps

1. Start MySQL server.
2. Provide connection details.
3. Establish connection.
4. Execute SQL queries.
5. Close connection.

Importance

- Enables database access.

- Supports dynamic applications.
- Allows storing and retrieving information.

Exam Definition:
PHP-MySQL connection is the process of establishing communication between PHP scripts and MySQL databases.

---

4. Creating a Database in MySQL --

Definition

A database is an organized collection of data stored electronically. MySQL allows users to create databases using SQL commands.

SQL Command

```
CREATE DATABASE database_name;
```

Example:

```
CREATE DATABASE school;
```

PHP Example

```
$sql="CREATE DATABASE school";
mysqli_query($conn,$sql);
```

Steps

1. Connect PHP to MySQL server.
2. Execute CREATE DATABASE command.
3. Verify database creation.
4. Select database for operations.

Advantages

- Organizes data efficiently.
- Supports multiple tables.
- Provides secure data storage.
- Allows easy data retrieval.

Applications

- Student management systems.
- Banking systems.
- Online shopping websites.

Exam Definition:
Creating a database means generating a new storage area in MySQL to store and manage related data.

---

5. Selecting a Database --

Definition

Selecting a database means choosing a specific database on which SQL operations will be performed.

SQL Command

```
USE database_name;
```

Example:

```
USE college;
```

PHP Method

```
mysqli_select_db($conn,"college");
```

Importance

- Identifies the working database.
- Allows table creation and modification.
- Ensures queries run on the correct database.

Steps

1. Connect to MySQL server.
2. Create database.
3. Select required database.
4. Perform operations.

Example Applications

- Selecting customer database.

- Accessing student records.
- Managing product information.

Exam Definition:
Selecting a database specifies the database on which further SQL operations will be executed.

---

6. Listing Databases and Tables --

Definition

MySQL provides commands to display available databases and tables stored on a server.

Listing Databases

Command:

```
SHOW DATABASES;
```

Purpose

- Displays all databases available on the server.
- Helps administrators manage databases.

Listing Tables

First select database:

```
USE college;
```

Command:

```
SHOW TABLES;
```

Purpose

- Displays tables inside selected database.
- Helps identify available data structures.

PHP Example

```
$result=mysqli_query($conn,"SHOW TABLES");
```

Importance

- Database management.
- Troubleshooting.
- Checking available resources.

Exam Definition:
Listing databases and tables displays the available databases and table structures stored in MySQL.

---

7. Creating a Table in MySQL --

Definition

A table is a database structure used to store data in rows and columns. Each table contains fields and records.

SQL Command

```
CREATE TABLE student(
id INT,
name VARCHAR(50),
age INT
);
```

Table Components

- **Column:** Represents a data field.
- **Row:** Represents a record.
- **Primary Key:** Unique identifier for records.

PHP Example

```
$sql="CREATE TABLE student(id INT,name VARCHAR(50))";
mysqli_query($conn,$sql);
```

Advantages

- Organizes data.
- Supports searching and sorting.
- Provides structured storage.
- Improves data management.

Exam Definition:
Creating a table means defining a structured format of rows and columns to store data in a database.

---

8. Inserting Data into Tables --

Definition

Insertion is the process of adding new records into a MySQL table.

SQL Syntax

```
INSERT INTO table_name
VALUES(value1,value2);
```

Example:

```
INSERT INTO student
VALUES(1,'John');
```

PHP Example

```
$sql="INSERT INTO student VALUES(1,'John')";
mysqli_query($conn,$sql);
```

Steps

1. Connect database.
2. Select table.
3. Write INSERT query.
4. Execute query.
5. Verify inserted record.

Importance

- Stores user information.
- Adds new records.
- Maintains database information.

Applications

- Registration systems.
- Employee databases.
- Product records.

Exam Definition:
Insertion is a database operation used to add new records into a MySQL table.

---

9. Altering Tables in MySQL --

Definition

ALTER command is used to modify the structure of an existing table. It can add, remove, or change columns.

Operations

Add Column

```
ALTER TABLE student
ADD email VARCHAR(50);
```

Modify Column

```
ALTER TABLE student
MODIFY name VARCHAR(100);
```

Delete Column

```
ALTER TABLE student
DROP COLUMN age;
```

Uses

- Updating table structure.
- Adding new information.
- Removing unnecessary fields.

Advantages

- Allows database modification without recreating tables.
- Saves time.
- Maintains existing data.

Exam Definition:
ALTER TABLE command modifies the structure of an existing database table.

---

10. Queries in PHP and MySQL --

Definition

A query is an SQL instruction used to communicate with a database. PHP executes MySQL queries to perform database operations.

Types of Queries

1. SELECT Query

Retrieves data.

```
SELECT * FROM student;
```

2. INSERT Query

Adds records.

3. UPDATE Query

Changes existing records.

4. DELETE Query

Removes records.

PHP Execution

```
mysqli_query($conn,$sql);
```

Query Functions

- mysqli_query()
- mysqli_fetch_array()
- mysqli_num_rows()

Importance

- Retrieves information.
- Updates databases.
- Controls stored data.
- Supports dynamic websites.

Exam Definition:
Queries are SQL statements executed through PHP to perform operations on MySQL databases.

---

11. Deleting Database, Data, and Tables --

Definition

Deletion commands are used to remove databases, tables, or records from MySQL.

Delete Database

Command:

```
DROP DATABASE database_name;
```

Removes complete database.

Delete Table

Command:

```
DROP TABLE table_name;
```

Removes table structure and data.

Delete Data

Command:

```
DELETE FROM student WHERE id=1;
```

Removes selected records.

Difference

Command	Purpose
DROP	Removes database/table completely

Command	Purpose
DELETE	Removes selected records

Importance

- Removes unnecessary information.
- Maintains database organization.
- Helps database management.

Exam Definition:
Deletion commands remove databases, tables, or records from MySQL according to requirements.

---

12. PHPMyAdmin --

Definition

PHPMyAdmin is a free, open-source web-based application used to manage MySQL databases through a graphical user interface (GUI). It simplifies database administration without requiring command-line knowledge.

Features

- Create and delete databases.
- Create and modify tables.
- Insert and edit records.
- Run SQL queries.
- Import and export databases.
- Manage users and permissions.

Advantages

- Easy database management.
- User-friendly interface.
- Reduces manual command writing.
- Supports database backup.

Uses

- Web development.
- Database administration.
- Testing applications.

- Managing MySQL servers.

Example

PHPMyAdmin is commonly included with XAMPP and WAMP server packages for local PHP development.

Exam Definition:
PHPMyAdmin is a web-based tool used to manage MySQL databases through an easy graphical interface.

---

13. Database Bugs and Troubleshooting --

Definition

Database bugs are errors or problems that occur during database design, connection, or query execution. They can affect application performance and data accuracy.

Common Database Bugs

1. Connection Errors

- Incorrect username/password.
- Server not running.

2. SQL Syntax Errors

- Incorrect SQL commands.
- Missing keywords.

3. Data Errors

- Duplicate records.
- Incorrect data types.

4. Security Issues

- SQL injection attacks.
- Weak passwords.

Troubleshooting Methods

- Check error messages.
- Test database connection.
- Verify SQL queries.
- Check table structure.
- Validate user input.
- Use backups.

Prevention

- Use prepared statements.
- Apply proper validation.
- Maintain regular backups.

Exam Definition:
Database bugs are errors in database operations that affect data storage, retrieval, and application performance.

Quick Revision Table

Topic	Key Idea
PHP and MySQL	Used together for dynamic database-driven websites.
Basic Commands	SQL commands manage database operations.
Connection	Establishes communication between PHP and MySQL.
Create Database	Creates storage area for data.
Select Database	Chooses database for operations.
List Database/Tables	Displays available databases and tables.
Create Table	Defines data structure using rows and columns.
Insert Data	Adds new records.
Alter Table	Modifies table structure.
Queries	Perform database operations.
Delete Operations	Remove databases, tables, or records.
PHPMyAdmin	GUI tool for MySQL management.
Database Bugs	Errors affecting database operations.

Important Exam Points:

- PHP communicates with MySQL using **MySQLi or PDO**.
- SQL commands include **CREATE, SELECT, INSERT, UPDATE, DELETE, ALTER, DROP**.
- Databases contain tables; tables contain rows and columns.

- PHPMyAdmin provides graphical database management.
- Prepared statements help prevent **SQL injection attacks**.