



Web Design and Development

Student Module

Grade 11



FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA
MINISTRY OF EDUCATION

Web Design and Development

Student Module

Grade 11

Writers:

Bruk T/Berhan Keweti (MSc)

Dereze Teshome Demisie (MSc)

Misikr Selemon Tsige (MSc)

Evaluators:

Abebe Basazinew Bekele (MSc)

Bazezew Tsehay Bekele (MSc)

Mekonnen Mulugeta Moges (MSc)

© Federal Democratic Republic of Ethiopia, Ministry of Education 2024

All rights reserved. No part of this Module may be reproduced, stored in a retrieval system or transmitted in any form or by any means including electronic, mechanical, magnetic, photocopying, recording or otherwise, without the prior written permission of the Ministry of Education or licensing in accordance with the Federal Democratic Republic of Ethiopia as expressed in the Federal Negarit Gazeta, Proclamation No. 410/2004 - Copyright and Neighboring Rights Protection.

First Edition 2024

Every effort has been made to trace the copyright holders and the Ministry of Education apologizes in advance for any unintentional omissions. If you are the owner of copyrighted material not cited or improperly cited, please contact the Ministry of Education, Head Office, Arat Kilo, (P.O. Box 1367), www.moe.gov.et, Addis Ababa Ethiopia.

ISBN: xxx-xxxxxx-x-xxx-x

Acknowledgement

The Federal Democratic Republic of Ethiopia, Ministry of Education expresses its gratitude to the Federal Technical and Vocational Training Institute, Manufacturing Industry Development Institute Kaizen Excellence Center, Misrak Polytechnic College, General Wingate Polytechnic College, Entoto Polytechnic College and Dagmawi Menelik Secondary School for professional support in the development of this Module. The Ministry of Education is grateful for the time and unwavering commitment of expertise provided by the writers and evaluators of the aforementioned institutions.

The Ministry of Education also acknowledges those who played key roles. Negesse Teferedegn Tilaye and Ashenafi Nigusse W/hana are acknowledged for their writing of the initial generic Professional Ethics Module, while Esubalew Wondim Zegeye and Gebeyaw Tesfaye Ayehu are recognized for their evaluation of this Professional Ethics Module. In addition, Kindu Oumer Ali and Tujuba Ragasa Garedew are appreciated for their writing of the initial generic Basic Kaizen Module, while Desalegne Berhanu Addisu and Girma Hailu Hailemicheal are recognized for their evaluation of this Basic Kaizen Module.

LIST OF MODULES

| Module | Module Title | Page |
|---------------|---------------------------------------|-------------|
| Module I | Operate Personal Computer | 1 |
| Module II | Access and Analyze Online Information | 77 |
| Module III | Develop Professional Ethics | 261 |
| Module IV | Build Web Pages Using HTML5 | 313 |
| Module V | Integrate Cascading Style Sheets | 449 |
| Module VI | Apply interactivity to a website. | 572 |
| Module VII | Basic Kaizen | 664 |

Introduction to the Student Module

Web Design and Development learning module! In today's digital age, the internet plays a significant role in our lives, providing us with endless opportunities to connect, learn, and share information. Behind every website you visit lies a team of skilled web designers and developers who work tirelessly to create engaging and user-friendly online experiences.

This learning module is designed to introduce Grade 11 students within 315 of total period to the exciting world of web design and development. Throughout this module, you will explore various concepts and skills essential for building and managing websites effectively. Whether you're interested in pursuing a career in web development or simply want to enhance your digital skills, this module will provide you with a solid foundation.

By the end of this learning module, you will have gained a comprehensive understanding of web design and development. You will possess the skills necessary to create visually appealing, user-friendly websites and manage their content effectively. Whether you aspire to become a professional web developer or simply want to enhance your digital literacy, this module will provide you with a solid foundation to embark on your web design and development journey.

So, let's get started and explore the exciting world of web design and development together!

This Web Design and Development learning modules for grade 11 embraces the following five (5) modules:

1. Operating a Personal Computer:

- Connect Hardware and Network Peripherals
- Install operating system and application Software
- Navigate and use Operating System

2. Access and analyze online information

- Describe the Internet and the world wide web
- Use and configure Web Browsers
- Search effectively for online information and critically evaluate web content.
- Enhance online communication and collaboration
- Create a safe online environment and Well-Being

3. Building Web Pages with HTML:

- Identify markup languages
- Setup the HTML development environment and Create HTML pages

- Create web pages and insert contents using HTML
4. Build Web Pages with CSS:
- Apply basic CSS Properties
 - Create layout and navigation of a web page
5. Adding Interactivity to Websites:
- Identify scripting language
 - Apply interactivity using JavaScript
 - Validate User Input

The modules are designed to provide a structured and progressive learning experience, building upon the knowledge and skills gained in each preceding module. By following this organized sequence, Grade 11 students will develop a comprehensive skill of web design and development, gradually acquiring the necessary tools to create visually appealing, user-friendly, and dynamic websites.

Module I

Operate Personal Computer

Contents

| No | Contents | page |
|----------|--|------|
| | Module Description | 1 |
| 1 | UNIT 1: Web Careers Pathways | 2 |
| | 1.1 Unit Overview | 2 |
| | 1.2 Web Design and Development career Paths | 4 |
| | Unit Summary | 10 |
| | Unit Review Questions | 11 |
| 2 | Unit 2: Hardware and Network Peripherals | 12 |
| | 2.1 Introduction to Computer | 12 |
| | 2.2 Organization of computer systems | 24 |
| | 2.3 Safe Work Conditions and Procedures | 27 |
| | 2.4 Environment Protection | 35 |
| | Unit Summary | 36 |
| | Unit Review Questions | 37 |
| 3 | Unit 3: Operating System And Application Software | 40 |
| | 3.1 Operating Systems and Installation | 40 |
| | 3.2 Application Software Installation | 49 |
| | 3.3 Installing Device Drivers | 51 |
| | Unit Summary | 56 |
| | Unit Review Questions | 56 |
| 4 | Unit 4: Operating System Environment | 59 |
| | 4.1 Navigate and use the Operating System environment | 59 |
| | 4.2 Desktop Environment | 60 |
| | 4.3 Manage Files and Folders | 61 |
| | 4.4 Uninstalling and Removing Applications | 64 |
| | 4.5 Create and Manage User Accounts | 64 |
| | 4.6 Essential keyboard shortcuts | 65 |
| | 4.7 Use Windows system tools and applications | 66 |
| | Unit Summary | 68 |
| | Unit Review Questions | 68 |
| | References | 76 |

Module Description

This Module defines the competence required to operate a personal computer, including starting the PC, logging in, using and understanding desktop icons and their links to underlying programs, navigating a directory structure, saving work, printing, operate word processing and closing down the PC. This will deliver a total of 100 periods: 24 periods for theory and 76 for practical training.

This module will also assist you to attain the following learning outcomes:

- Identify web careers pathways
- Connect Hardware and Network Peripherals
- Install operating system and application Software
- Navigate and use Operating System

Module Instruction:

Learning Instructions: How to use this Module

For effective use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what are being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

UNIT 1

WEB CAREERS PATHWAYS

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 1.1. Introduction Information Technology as a career
- 1.2. Introduction to Web Design and Development career Paths

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Identify web careers pathways

Key Terms: *web, OS, Application software, Navigate Operating system, Back end, Front End and Full stack development.*

1.1. Unit Overview

Information and Communication Technology (ICT) is a broad term that encompasses the use of technology to manage and process information. It refers to the convergence of computing, telecommunications, and information systems. ICT plays a crucial role in various industries, organizations, and sectors, transforming the way we communicate, work, and access information.

As a career field, Information Technology offers diverse opportunities for individuals interested in working with computers, networks, software, and data. Professionals in this field use their technical skills and knowledge to design, develop, implement, and manage technology solutions to meet the needs of businesses and individuals.

Here are some key aspects of IT as a career:

Job Roles: IT offers a wide range of job roles, including but not limited to:

- IT Support Specialist: Provide technical assistance and support to users.
- Network Administrator: Design, implement, and maintain computer networks.
- System Administrator: Manage and maintain computer systems and servers.
- Software Developer: Design and develop software applications.
- Data Analyst: Analyze and interpret data to derive insights.
- Cyber security Analyst: Protect computer systems and networks from security threats.
- IT Project Manager: Plan, execute, and manage technology projects.

Skill Set: A career in IT requires a combination of technical skills and knowledge. Some essential skills include:

- Proficiency in programming languages and software development.
- Knowledge of computer networks, hardware, and operating systems.
- Understanding of databases and data management.
- Problem-solving and analytical thinking.
- Strong communication and teamwork skills.
- Continuous learning and adaptability to keep up with evolving technologies.

Industry Demand: With the increasing reliance on technology, there is a growing demand for skilled IT professionals. Almost every industry, including healthcare, finance, education, entertainment, and manufacturing, relies on IT to streamline operations, improve efficiency, and deliver innovative solutions. This demand creates numerous job opportunities for IT professionals.

Career Growth: IT offers excellent career growth prospects. Professionals can advance their careers by gaining experience, acquiring certifications, and continuously updating their skills. There are opportunities to move into management roles, specialize in specific areas like cyber security or data science, or even start their own businesses as technology consultants.

Continuous Learning: The field of IT is constantly evolving, with new technologies and innovations emerging regularly. As an IT professional, continuous learning is essential to stay updated with the latest trends, tools, and techniques. This can be done through self-study, attending workshops, obtaining certifications, or pursuing higher education.

Impact and Innovation: IT has a significant impact on society, enabling advancements in various fields. It has revolutionized communication, improved access to information, transformed business processes, and facilitated global connectivity. IT professionals have the opportunity to contribute to meaningful projects, drive innovation, and make a positive difference in the world.

Overall, a career in IT offers a dynamic and challenging environment for individuals passionate about technology. It provides opportunities to work on exciting projects, solve complex problems, and contribute to the digital transformation of organizations and society as a whole.

1.2. **Web Design and Development career Paths**

Web development is a thriving field with a wide range of career opportunities. Here are some popular web career pathways:

1. Web Designer:

- Create visually appealing and user-friendly website layouts.
- Use graphic design tools and principles to design website elements.
- Work closely with clients to understand their design requirements.

2. Front-End Developer:

- Implement the visual and interactive elements of a website.
- Use HTML, CSS, and JavaScript to create engaging user interfaces.
- Ensure cross-browser compatibility and responsive design.

3. Back-End Developer:

- Develop the server-side logic and functionality of websites or web applications.
- Utilize programming languages such as Python, Ruby, PHP, or Java.
- Design and manage databases, handle server operations, and integrate APIs.

4. Full-Stack Developer:

- Possess knowledge and skills in both front-end and back-end development.
- Work on all aspects of web development, from designing interfaces to server-side programming.
- Handle the entire web development process and provide end-to-end solutions.

5. User Experience (UX) Designer:

- Focus on creating intuitive and user-friendly website experiences.
- Conduct user research and usability testing to optimize website usability.
- Use wire framing and prototyping tools to design seamless user journeys.

6. Content manager:

- Manage and update website content, ensuring its accuracy and relevance.
- Coordinate with content creators and stakeholders to publish engaging content.
- Implement content management systems and SEO best practices.

These are just a few examples of the many career paths available in web development like “web developer”, “website administrator”, “website Administrator” and etc. Each career path offers unique opportunities for growth and specialization. It's essential to explore your interests, skills, and goals to find the right fit for your web career.

1.2.1. Front-end development

Dear learners! What is Front-end development? Can you try to define the term based the introduction given above? Hoping that you have defined it, let's see together some of the most common definitions of web development as given by different authors.

Front-end development refers to the practice of building the user-facing aspects of websites or web applications. Front-end developers work on the client side of web development, focusing on creating visually appealing and interactive interfaces that users directly interact with.

Key aspects of front-end development include:

1. **HTML (Hypertext Markup Language):** Front-end developers use HTML to structure the content of web pages. HTML defines the elements and their relationships, such as headings, paragraphs, images, links, forms, and more.
2. **CSS (Cascading Style Sheets):** CSS is used to style and format the HTML elements. Front-end developers use CSS to control the layout, colors, fonts, spacing, and other visual aspects of a web page, ensuring a consistent and appealing design.
3. **JavaScript:** JavaScript is a programming language that adds interactivity and dynamic behavior to web pages. Front-end developers use JavaScript to create interactive elements, handle user events, and manipulate the content of a web page in response to user actions.

4. **Responsive Design:** Front-end developers ensure that websites are optimized for various devices and screen sizes. They use responsive design techniques to create layouts that adapt and resize automatically, providing a seamless experience across desktops, tablets, and mobile devices.
5. **Frameworks and Libraries:** Front-end developers often leverage frameworks and libraries to streamline development and enhance productivity. Popular front-end frameworks include React, Angular, and Vue.js, while libraries like jQuery provide pre-built functions to simplify common tasks.
6. **Cross-Browser Compatibility:** Front-end developers need to ensure that websites function properly across different web browsers (such as Chrome, Firefox, Safari, and Edge) and versions. They test and optimize the code to maintain consistent functionality and appearance.
7. **Web Performance Optimization:** Front-end developers optimize websites for fast loading times and optimal performance. Techniques include minifying and compressing files, optimizing images, and reducing HTTP requests to improve the overall user experience.
8. **Accessibility:** Front-end developers strive to create websites that are accessible to users with disabilities. They follow accessibility guidelines, use semantic markup, provide alternative text for images, and ensure keyboard navigation is available.

Front-end development requires a combination of technical skills, creativity, and attention to detail. Front-end developers collaborate closely with designers, back-end developers, and other team members to ensure a seamless integration of design and functionality. They play a crucial role in delivering a visually engaging and user-friendly experience to website visitors.



Self-check 1-1:

1. What is the role of HTML in front-end development?
 2. How does CSS contribute to the visual styling of a website?
 3. What is JavaScript, and what is its role in front-end development?
-

1.2.2. Back-end development

Dear learners! What is Back-end development? Can you try to define the term based the introduction given above? Hoping that you have defined it, let's see together some of the most common definitions of web development.

Back-end development refers to the aspect of web development that focuses on the server-side of websites or web applications. Back-end developers work on the behind-the-scenes functionality that powers the website and handles data processing, storage, and communication.

Key aspects of back-end development include:

1. **Server-Side Programming Languages:** Back-end developers work with programming languages like Python, Ruby, PHP, Java, or Node.js to build the logic and functionality of the server-side. These languages allow developers to handle data, interact with databases, and perform complex computations.
2. **Databases:** Back-end developers work with databases to store and manage data. Common databases used in back-end development include MySQL, PostgreSQL, MongoDB, and Oracle. Developers design database schemas, write queries to retrieve and manipulate data, and ensure data integrity.
3. **APIs (Application Programming Interfaces):** Back-end developers create APIs that allow different software systems to communicate with each other. APIs enable the exchange of data and functionality between the front-end and back-end, as well as integration with external services or platforms.
4. **Server Management:** Back-end developers handle server configuration and management. They ensure the proper setup and deployment of web servers, such as Apache or Nginx, and manage hosting environments, scalability, security, and performance optimizations.
5. **Security:** Back-end developers play a crucial role in ensuring the security of web applications. They implement authentication and authorization mechanisms, protect against common vulnerabilities like cross-site scripting (XSS) or SQL injection, and apply encryption and secure communication protocols.
6. **Frameworks and Libraries:** Back-end developers often utilize frameworks and libraries to streamline development and leverage pre-built functionalities. Popular back-end frameworks include Django (Python), Ruby on Rails (Ruby), Laravel (PHP), Spring (Java), and Express.js (Node.js).

7. **Testing and Debugging:** Back-end developers write tests to ensure the reliability and stability of the server-side code. They perform debugging to identify and fix software issues and optimize performance.
8. **Integration of Third-Party Services:** Back-end developers integrate external services, such as payment gateways, email services, or social media APIs, into web applications to extend functionality and enhance user experience.

Back-end development focuses on the server-side infrastructure and functionality that makes a website or web application work. Back-end developers collaborate closely with front-end developers, designers, and other team members to ensure a seamless integration of user interfaces with the underlying server logic. They are responsible for creating robust and efficient server-side systems that handle data processing, business logic, and communication with external services.



Self-check 1-2:

-
1. Name some common back-end programming languages.
 2. What is the role of databases in back-end development?
 3. Name some popular back-end frameworks or libraries and their use cases.
-

1.2.3. Full-stack development

Dear learners! *What is Full-stack development? Can you try to define the term based the introduction given above? Hoping that you have defined it, let's see together some of the most common definitions of web development.*

Full-stack development refers to the practice of being proficient in both front-end and back-end development. Full-stack developers have the knowledge and skills to work on all layers of a web application, from the user interface to the server-side logic and database management. They are capable of handling various aspects of web development, making them versatile and valuable in the industry.

Key aspects of full-stack development include:

1. Front-End Development: Full-stack developers are proficient in front-end technologies such as HTML, CSS, and JavaScript. They can create visually appealing and interactive user interfaces, ensuring a seamless user experience.
2. Back-End Development: Full-stack developers have a strong understanding of server-side programming languages like Python, Ruby, PHP, Java, or Node.js. They can develop the logic, functionality, and database management required to handle server-side operations.
3. Databases: Full-stack developers are skilled in working with databases such as MySQL, PostgreSQL, MongoDB, or Oracle. They can design efficient database schemas, write complex queries, and optimize data retrieval and storage.
4. Server Management: Full-stack developers have knowledge of server management, including setting up and configuring web servers, managing hosting environments, and ensuring scalability, security, and performance optimizations.
5. APIs and Integration: Full-stack developers can create and consume APIs to enable communication between different software systems. They can integrate external services or platforms into web applications, expanding functionality and enhancing user experience.
6. Version Control: Full-stack developers are proficient in using version control systems like Git. They can manage code repositories, collaborate with other developers, and ensure code integrity and versioning.
7. Problem Solving and Troubleshooting: Full-stack developers possess strong problem-solving skills and can debug issues across the entire web application stack. They can identify and resolve software bugs, optimize performance, and ensure efficient operation.
8. Project Management: Full-stack developers have a holistic understanding of the web development process and can contribute to project management tasks. They can plan, estimate, and prioritize tasks, collaborate with team members, and ensure the successful delivery of projects.

By combining front-end and back-end skills, full-stack developers can take on a broader range of responsibilities in web development projects. They can work independently or collaborate with specialized front-end and back-end developers, depending on the scale and requirements of the project. Full-stack developers are capable of handling the entire development process, from designing interfaces to implementing server-side logic, and are highly adaptable to different project needs and technologies.



Self-check 1-3:

1. What is full-stack development, and how does it differ from front-end and back-end development?
 2. What are the advantages of being a full-stack developer?
 3. Explain the skills and knowledge required for full-stack development.
-

Unit Summary

In this unit, we explored various career pathways in web development. Some of the pathways include web design, front-end development, back-end development, full-stack development, user experience (UX) design, web content management, web project management, web analytics, SEO specialization, and e-commerce specialization.

Front-end development focuses on building the user-facing aspects of websites, using HTML, CSS, and JavaScript to create visually appealing and interactive interfaces.

Back-end development involves working on the server-side of websites, handling data processing, storage, and communication using programming languages, databases, APIs, and server management.

Full-stack development combines both front-end and back-end development skills, allowing developers to work on all aspects of web development and provide end-to-end solutions.

Each career pathway offers unique opportunities for specialization and growth, and it's important to explore your interests, skills, and goals to choose the right path for your web development career.

Unit Review Questions

Instruction: Give brief answer for the following review questions: These questions should help you review the key concepts and knowledge related to front-end development, back-end development, and full-stack development. Remember to provide detailed answers and examples to reinforce your understanding of each topic.

1. Identify and explain web development career pathways?
2. What is front-end development, and what is its primary focus?
3. Name some key technologies used in front-end development.
4. Explain the concept of responsive design and why it is important in front-end development.
5. How do front-end developers ensure cross-browser compatibility?
6. What is the significance of accessibility in front-end development?
7. Name some popular front-end frameworks or libraries and their use cases.
8. How can front-end development contribute to a seamless user experience?
9. What is back-end development, and what is its primary focus?
10. Explain the concept of APIs and their importance in back-end development.
11. What are some security considerations in back-end development?
12. How do back-end developers handle data storage and retrieval?
13. What is the role of testing and debugging in back-end development?
14. How do full-stack developers integrate front-end and back-end components?
15. What are some common challenges faced by full-stack developers?
16. How does version control contribute to full-stack development workflow?
17. Describe the role of full-stack developers in project management.
18. What are some popular tools or technologies used in full-stack development?
19. How does being a full-stack developer provide versatility and flexibility in the industry?

UNIT 2

HARDWARE AND NETWORK PERIPHERALS

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 2.1. *Introduction to Computer*
- 2.2. *Organization of computer systems*
- 2.3. *Safe Work Conditions and Procedures*
- 2.4. *Equipment Protection*
- 2.5. *Environment Protection*

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ *Connect hardware and network peripherals to the system /system unit.*

Key Terms: *Computer, Organization of computer, Procedures, Equipment Protection*

2.1. Introduction to Computer

Computers have become an integral part of our daily lives, revolutionizing the way we work, communicate, and access information. A computer is a powerful electronic device that can perform various tasks, process data, and execute instructions. It has transformed industries, accelerated scientific advancements, and enhanced our capabilities in numerous fields.

A peripheral is a piece of computer hardware that is added to a computer in order to expand its abilities. The term peripheral is used to describe those devices that are optional in nature, as opposed to hardware that is either demanded or always required in principle. There are all different kinds of peripherals you can add to your computer. The main distinction among peripherals is the way they are connected to your computer. They can be connected internally or externally.

Buses

A bus is a subsystem that transfers data between computer components inside a computer or between computers. Unlike a point-to-point connection, a bus can logically connect several peripherals over the same set of wires. Each bus defines its set of connectors to physically plug devices, cards or cables together. There are two types of buses: internal and external. Internal buses are connections to various internal components. External buses are connections to various external components. There are different kinds of slots that internal and external devices can connect to.

Internal

Types of Slots

There are many different kinds of internal buses, but only a handful of popular ones. Different computers come with different kinds and number of slots. It is important to know what kind and number of slots you have on your computer before you go out and buy a card that matches up to a slot you don't have.

PCI

PCI (Peripheral Component Interconnect) is common in modern PCs. This kind of bus is being succeeded by PCI Express. Typical PCI cards used in PCs include: network cards, sound cards, modems, extra ports such as USB or serial, TV tuner cards and disk controllers. Video cards have outgrown the capabilities of PCI because of their higher bandwidth requirements.



Fig: 2.1.1. PCI Slots

PCI Express

PCI Express was introduced by Intel in 2004. It was designed to replace the general-purpose PCI expansion bus and the AGP graphics card interface. PCI express is not a bus but instead a point-to-point connection of serial links called lanes. PCI Express cards have faster bandwidth than PCI cards which make them more ideal for high-end video cards.

AGP

AGP (Accelerated Graphics Port) is a high-speed point-to-point channel for attaching a graphics card to a computer's motherboard, primarily to assist in the acceleration of 3D computer graphics. AGP has been replaced over the past couple years by PCI Express. AGP cards and motherboards are still available to buy, but they are becoming less common.

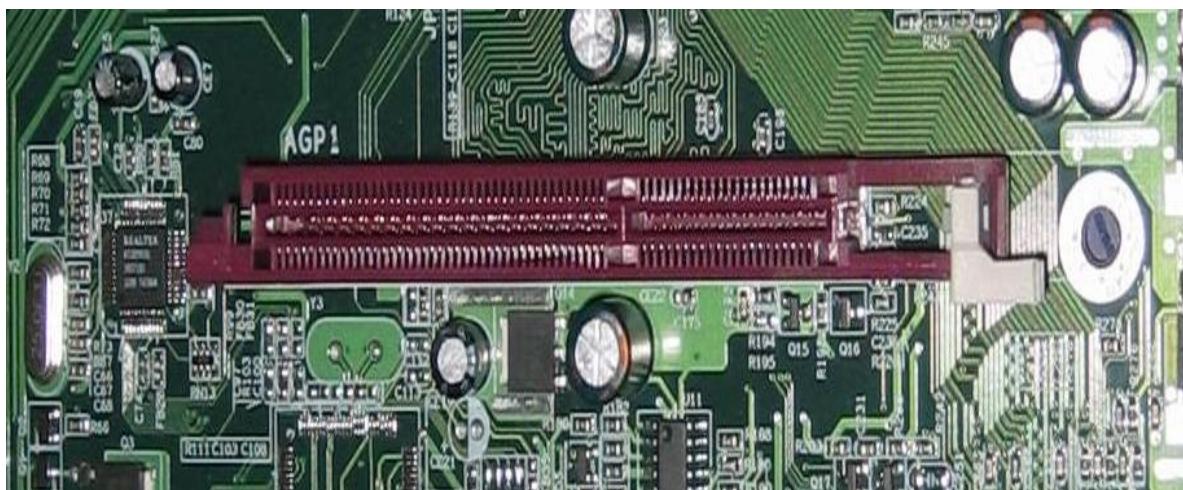


Fig: 2.1.2. AGP Slot

Types of Cards

Video Card

A video card (also known as graphics card) is an expansion card whose function is to generate and output images to a display. Some video cards offer added functions, such as video capture, TV tuner adapter, ability to connect multiple monitors, and others. Most video cards all share similar components. They include a graphics processing unit (GPU) which is a dedicated microprocessor optimized for 3D graphics rendering.

It also includes a video BIOS that contains the basic program that governs the video card's operations and provides the instructions that allow the computer and software to interface with the card. If the video card is integrated in the motherboard, it may use the computer RAM memory. If it is not it will have its own video memory called Video RAM. This kind of memory can range from 128MB to 2GB. A video card also has a RAMDAC (Random Access Memory Digital-to-Analog Converter) which takes responsibility for turning the digital signals produced by the computer processor into an analog signal which can be understood by the computer display. Lastly, they all have outputs such as an HD-15 connector (standard monitor cable), DVI connector, S-Video, composite video or component video.



Fig: 2.1.3. Graphics Card

Sound Card

A sound card is an expansion card that facilitates the input and output of audio signals to/from a computer under control of computer programs. Typical uses for sound cards include providing the audio component for multimedia applications such as music composition, editing video or audio, presentation/education, and entertainment. Many computers have sound capabilities built in,, while others require additional expansion cards to provide for audio capability.

Network Card

A network card is an expansion card that allows computers to communicate over a computer network. It allows users to connect to each other either by using cables or wirelessly. Although other network technologies exist, Ethernet has achieved near-ubiquity for a while now. Every Ethernet network card has a unique 48-bit serial number called a MAC address, which is stored in

ROM carried on the card. You can learn more about networking in the introduction to networking lesson.



Fig: 2.1.3 Network Card

External

Types of Connections

USB

USB (Universal Serial Bus) is a serial bus standard to interface devices. USB was designed to allow many peripherals to be connected using a single standardized interface socket and to improve the plug-and-play capabilities by allowing devices to be connected and disconnected without rebooting the computer. Other convenient features include providing power to low-consumption devices without the need for an external power supply and allowing many devices to be used without requiring manufacturer specific, individual device drivers to be installed. USB is by far the dominating bus for connecting external devices to your computer.

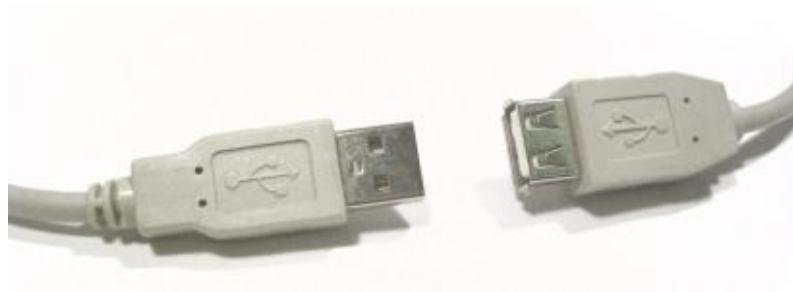


Fig: 2.1.4. USB Connectors

Fire wire

Firmware (technically known as IEEE 1394 and also known as i. LINK for Sony) is a serial bus interface standard for high-speed communications and isochronous real-time data transfer, frequently used in a personal computer? Firmware has replaced Parallel ports in many applications. It has been adopted as the High-Definition Audio-Video Network Alliance (HANA) standard connection interface for A/V (audio/visual) component communication and control. Almost all modern digital camcorders have included this connection.



Fig: 2.1.5. Firewire Cable

PS/2

The PS/2 connector is used for connecting some keyboards and mice to a PC compatible computer system. The keyboard and mouse interfaces are electrically similar with the main difference being that open collector outputs are required on both ends of the keyboard interface to allow bidirectional communication. If a PS/2 mouse is connected to a PS/2 keyboard port, the mouse may not be recognized by the computer depending on configuration.

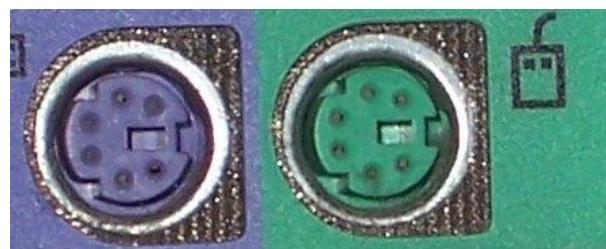


Fig: 2.1.6 PS/2 Ports

Devices

Removable Storage

The same kinds of CD and DVD drives that could come built-in on your computer can also be attached externally. You might only have a CD-ROM drive built-in to your computer but you need a CD writer to burn CDs. You can buy an external CD writer that connects to your USB port and acts the same way as if it was built-in to your computer. The same is true for DVD writers, Blu-ray drives, and floppy drives. Flash drives have become very popular forms of removable storage especially as the price of flash drives decreases and the possible size for them increases. Flash drives are usually USB ones either in the form USB sticks or very small, portable devices. USB flash drives are small, fast, removable, rewritable, and long-lasting. Storage capacities range from 64MB to 32GB or more. A flash drive does not have any mechanically driven parts so as opposed to a hard drive which makes it more durable and smaller usually.



Fig: 2.1.7: USB Flash Drive

Non-removable Storage

Non-removable storage can be a hard drive that is connected externally. External hard drives have become very popular for backups, shared drives among many computers, and simply expanding the amount of hard drive space you have from your internal hard drive. External hard drives come in many shapes and sizes like flash drives do. An external hard drive is usually connected by USB but you can also have a networked hard drive which will connect to your network which allows all computers on that network to access that hard drive.

Input

Input devices are absolutely crucial to computers. The most common input devices are mice and keyboards which barely every computer has. A new popular pointing device that may eventually

replace the mouse is touch screen which you can get on some tablet notebooks. Other popular input devices include microphones, webcams, and fingerprint readers which can also be built in to modern laptops and desktops. A scanner is another popular input device that might be built-in to your printer.



Fig: 2.1.8: Webcam

Output

There are lots of different kinds of output devices that you can get for your computer. The absolute most common external output device is a monitor. Other very popular output devices are printers and speakers. There are lots of different kinds of printers and different sizes of speakers for your computer. Monitors are connected usually through the HD-15 connector on your video card. Printers are usually connected through a USB port. Speakers have their own audio out port built-in to the sound card.



Fig: 2.1.9: Monitor

Practical Work:

1. Dear all students let us go and search different computer in different models from the internet.
 2. Identify all hardware components.
 3. Compare with your computer and Present for the class
-

Note: Please teacher form in group and evaluate the students as per the presentation

2.1.1. Generation Of computers

Web development is a thriving field with a wide range of career opportunities. Here are some popular web career pathways:

Computers have evolved over time through different generations, each marked by significant advancements in technology.

Characteristics and Capabilities of Computer Generations

| Generation of Computer | Size | Circuitry | Other Description |
|----------------------------------|----------------------------|-----------------------------------|--|
| First Generation (1940-1959) | Room Size Mainframe | Vacuum tubes | Limited storage capacity Slow speed Problem of over heating Programming over heating |
| Second Generation (1959-1964) | Closest size Mainframe | Transistors and diodes | Increased storage capacity Reduction in size and heat High Level Programming Language (COBOL, FORTRAN) |
| Third Generation (1965-1970) | Desk-size Mini computer | Instigates semiconductor circuits | More flexible with input/ Output Smaller size and better performance |

| | | | |
|-------------------------------------|-----------------------|-------------------------------------|--|
| Fourth Generation (Since 1970) | Desktop to handheld | Very large scale instigated circuit | Increase Storage Modular design versatility and compatibility Increased uses of microcomputers or (PC's) |
| Fifth Generation Future Computer | Small Card size Micro | VLSI superconductor or circuits | Artificial intelligence's and Robotics |

2.1.2. Characteristics of a Computer System

A computer system possesses several key characteristics that make it a powerful tool for processing information. These include:

- **Speed:** Computers are designed to perform tasks at incredibly high speeds. They can execute millions or even billions of instructions per second, depending on the processor's clock speed. This speed allows computers to perform complex calculations, process large amounts of data, and complete tasks in a fraction of the time it would take a human.
- **Accuracy:** Computers are highly accurate when it comes to executing instructions and performing calculations. They operate based on precise binary logic, ensuring that calculations and operations are carried out without errors. However, it's important to note that accuracy is dependent on the correctness of the input provided and the programming of the instructions.
- **Storage Capacity:** Computer systems have vast storage capacities, allowing them to store and retrieve massive amounts of data quickly. The primary storage component, such as RAM (Random Access Memory), provides temporary storage for data and instructions that are currently being processed. Secondary storage devices, such as hard drives or solid-state drives, offer larger capacities for long-term data storage.
- **Versatility:** Computers are highly versatile machines. They can be programmed to perform various tasks and handle different types of data. Through software applications, computers can process text, images, audio, video, and other forms of data. This versatility allows computers to be used in a wide range of industries and applications, from scientific research to business operations and entertainment.

- **Automation:** One of the key advantages of computers is their ability to automate repetitive tasks. Through programming and the use of software applications, computers can automate processes, reducing the need for manual intervention. This automation increases efficiency, accuracy, and productivity, freeing up human resources to focus on more complex and creative tasks.
- **Security:** The measures and protocols implemented to protect the computer system and its data from unauthorized access, breaches, or malicious activities. Security mechanisms include authentication, encryption, firewalls, and antivirus software.
- **Reliability:** The measure of how consistently and accurately the computer system performs its intended functions. Reliable systems minimize errors, crashes, and data loss, ensuring consistent and predictable operation.
- **Scalability:** The ability of the computer system to accommodate changes in workload or resource demands. Scalability allows the system to handle increased processing requirements, storage needs, or user demands without significant performance degradation.
- **Compatibility:** The ability of the computer system to work efficiently with different hardware, software, or data formats. Compatibility ensures seamless integration and interoperability with various devices, applications, and standards.
- **Upgradeability:** The ability to enhance or expand the capabilities of the computer system by adding or replacing hardware components or upgrading software. Upgradeability allows for improved performance, storage capacity, or functionality.

Note: Characteristics may vary depending on the specific hardware configuration, software capabilities, and programming of the computer system. Nonetheless, these inherent characteristics make computers powerful tools for processing information and performing a wide range of tasks efficiently.

2.1.3. Classification of computers

Computers can be classified into different categories based on their size, purpose, and capabilities. The common classifications include:

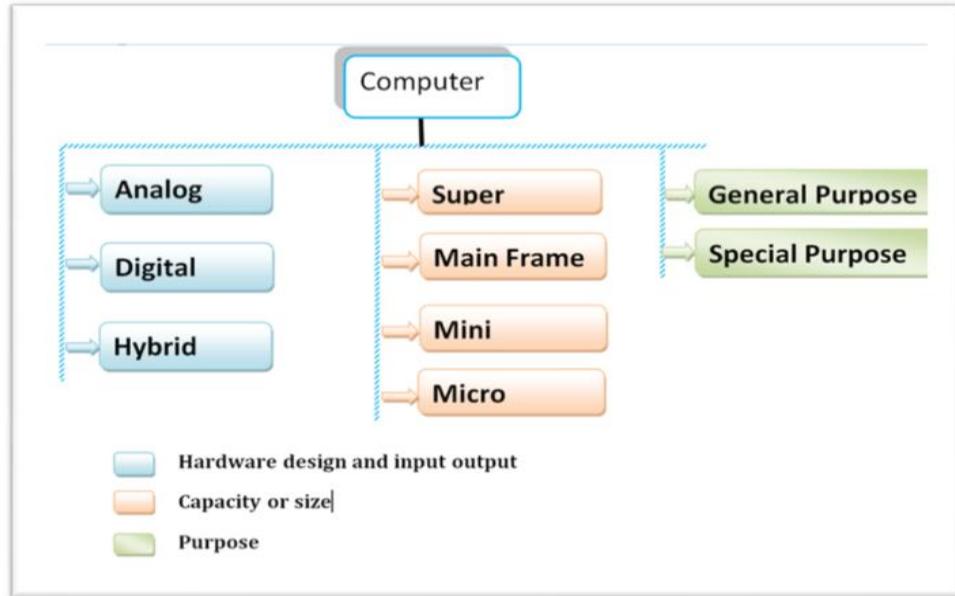


Fig:2.1.3. Classification of Computer

Discussion



Discuss about Classification of Computer which is place on Fig:2.1.3. above and explain for your class room.

2.1.4. Application of computers

Class work: feel free to discuss and write applications area of computers in our state? List all and present for the class room. Please Teacher follow and ask why and how it applicable.

Computers have a wide range of applications in various fields and industries. Here are some common applications of computers:

- *Education.*
- *Communication industry.*
- *Transport industry.*
- *Library services.*
- *Multimedia applications.*
- *Defense/Police (Law enforcement agencies).*
- *Government Institutions.*
- *Health Care*
- *Process control.*
- *Industries/ Banks/Insurance industries*
- *Supermarkets.*
- *Research.*
- *Engineering Design*

Discussion



Discuss how to apply computer in the above disciplines? Choose at least three industries

2.2. Organization of computer systems

The organization of computer systems ensures efficient data flow, storage, and communication, enabling the computer to function effectively and provide the desired functionality to users.

2.2.1. Organization of computer systems

Computer systems are organized into various components that work together to perform tasks and process information. The organization of computer systems typically includes the following components:

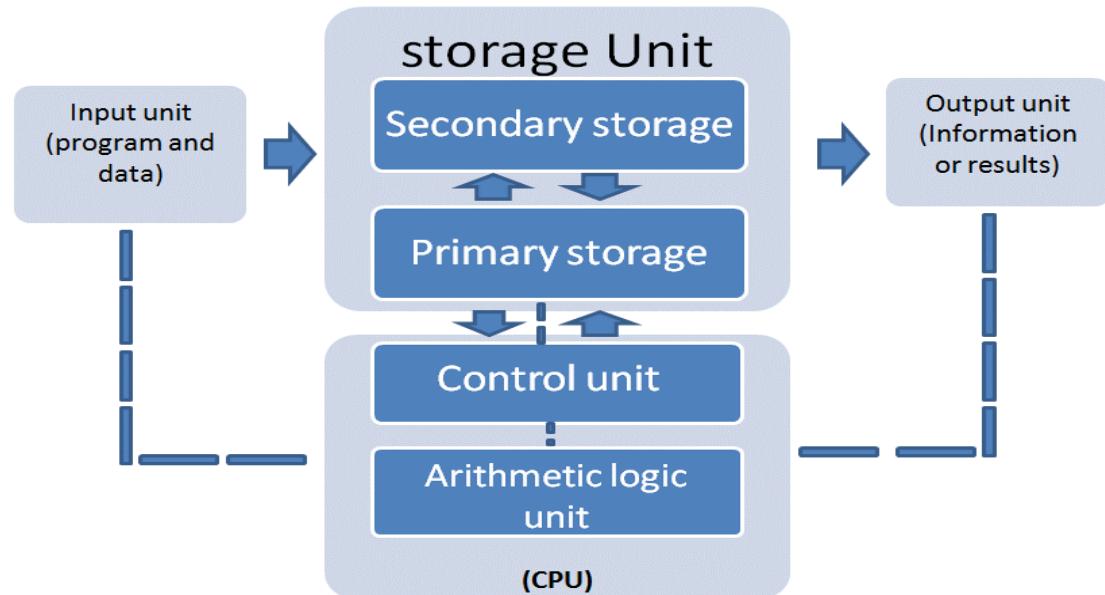


Fig: 2.2.1: Organization of Computer

- Central Processing Unit (CPU):** The CPU is the "brain" of the computer and performs most of the processing. It consists of the arithmetic logic unit (ALU) for performing calculations and the control unit (CU) for managing instructions and data flow.

2. **Memory:** Memory is used to store data and instructions that the CPU needs to access quickly. There are different types of memory, including:
 - Random Access Memory (RAM): RAM is volatile memory that provides temporary storage for data and instructions while the computer is running.
 - Read-Only Memory (ROM): ROM is non-volatile memory that stores permanent instructions, such as the computer's boot-up instructions.
3. **Storage Devices:** Storage devices are used for long-term data storage. Common types of storage devices include hard disk drives (HDDs), solid-state drives (SSDs), and optical drives (CD/DVD).
4. **Input Devices:** Input devices allow users to input data and instructions into the computer. Examples include keyboards, mice, touchscreens, scanners, and microphones.
5. **Output Devices:** Output devices display or present processed data to the user. Examples include monitors, printers, speakers, and projectors.
6. **Motherboard:** The motherboard is the main circuit board that connects and allows communication between all the components of the computer system.
7. **Expansion Cards:** Expansion cards are additional circuit boards that can be added to the motherboard to enhance the computer's capabilities. Examples include graphics cards, sound cards, and network interface cards.
8. **Bus System:** The bus system is a communication pathway that allows data and instructions to be transferred between the different components of the computer system.
9. **Operating System:** The operating system is a software program that manages the computer's resources, controls hardware operations, and provides a user interface. It enables the execution of applications and coordinates the interaction between software and hardware components.
10. **Software:** Software refers to the programs and applications that run on the computer system, including operating systems, productivity software, games, and utility programs.

These components work together to process data, execute instructions, and perform various tasks.

2.2.2. Computer Ports & Cables

Computer ports and cables are essential for connecting various devices and peripherals to a computer system. They facilitate data transfer, power supply, and communication between devices. Here are some commonly used ports and cables:

1. **USB (Universal Serial Bus) Ports:** USB ports are widely used for connecting devices such as keyboards, mice, printers, external hard drives, smartphones, and other peripherals. USB ports come in different versions, including USB 2.0, USB 3.0, USB 3.1, and USB-C, each offering different data transfer speeds and power delivery capabilities.
2. **HDMI (High-Definition Multimedia Interface) Port:** HDMI ports are used to connect computers to external displays, such as monitors, TVs, and projectors. They transmit high-quality audio and video signals, allowing for seamless multimedia playback.
3. **DisplayPort:** DisplayPort is another video interface used for connecting computers to monitors and other display devices. It supports high-resolution video and audio, and can also carry data signals.
4. **Ethernet Port:** Ethernet ports, also known as RJ-45 ports, are used for wired network connections. They enable computers to connect to local area networks (LANs) and access the internet through Ethernet cables.
5. **Audio Ports:** Audio ports, commonly found on computers and laptops, allow for the connection of headphones, speakers, microphones, and other audio devices. They can be 3.5mm audio jacks or specialized ports for specific audio needs.
6. **VGA (Video Graphics Array) Port:** VGA ports are older analog video ports used for connecting computers to displays. While they are less common now, some older monitors and projectors still use VGA connections.
7. **DVI (Digital Visual Interface) Port:** DVI ports are used for digital video connections and can support both analog and digital signals. They are commonly found on older monitors and graphics cards.
8. **Thunderbolt Port:** Thunderbolt ports are high-speed ports that can transmit both data and video signals. They are often found on Apple computers and allow for the connection of various peripherals, such as external hard drives and monitors.
9. **Power Port:** Power ports are used to connect the computer to a power source. They vary depending on the computer model, with some using AC power adapters and others using DC power connectors.

10. Cable Types: Various cables are used to connect devices to the computer ports. Some common cable types include USB cables, HDMI cables, Ethernet cables (e.g., Cat5e, Cat6), audio cables (e.g., 3.5mm audio cables), DisplayPort cables, and VGA cables.

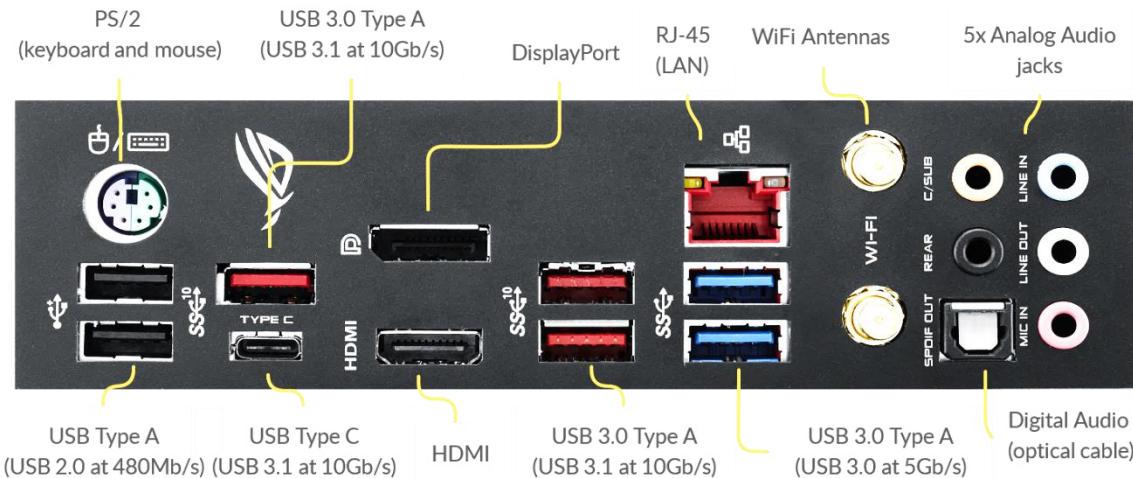


Fig:2.2.2. Computer Ports

It's important to note that the availability of specific ports and cables may vary depending on the computer model and its intended use. Additionally, advancements in technology may introduce new ports and cables to meet evolving connectivity needs.



Self-check 2-1:

1. What are the key components of a computer system?
2. Name and briefly explain the generations of computers.
3. What are the characteristics that define a computer system?

2.3. Safe Work Conditions and Procedures

Operation title: Connect Monitor to the work station with regular procedure

Purpose: To practice and demonstrate the knowledge and skill required in Connecting Monitor/Display to the system Unit

Instruction: Use the given tools and equipment to connect the peripherals. For this operation you have given 20min and demonstrate to your trainer

Tools and requirement:

- Personal Computer
- Peripheral Devices

Precautions: take under consideration any required Safety measures during work

Procedures used to accomplish the task

Step 1: - Find the monitor's connector cable. It should be included with your purchase.

Step 2: - Match the cable with one of the video ports on the back of the monitor. The images below show each of the video ports used today.



Step 3: - Connect the cable to the corresponding ports on both your monitor and computer.

Step 4: - If not already done, connect the flat end of the monitor power cord (left cable end shown below) to the back of the monitor. The power cord may also be built into the back of the monitor.



Step 5:- Connect the other end of the monitor power cord (shown above) into a power outlet, or better, a surge protector.

Step 6: - Turn on the computer and turn on the monitor. The power button for the monitor is often found on the front or bottom of the monitor on the right side.

2.3.1. General Safety Guideline

Safe work conditions and procedures for computer-related work are essential to ensure the well-being and safety of employees who spend a significant amount of time working with computers. Here are some key considerations for creating a safe work environment for computer-related tasks.

Health and Safety Computers

- No drinks food or wet materials near the computers.
- Ensure you log off once you have completed and saved your work.
- Report any damaged equipment or problems to your teacher.
- Clear your area of work materials before you leave the station.

The illustration shows a person with short brown hair, wearing a pink long-sleeved shirt and blue pants, sitting at a black desk. A computer monitor is on the desk, and the person is using a keyboard and mouse. The desk is labeled "Desk about 70cm high". The person's feet are resting on a footrest under the desk. The text "Footrest if required" is next to the footrest. The person's arms are extended over the keyboard, with the text "Wrist support if required" pointing to the wrists. The text "Top of screen at eye level" points to the monitor. The person's forearms are labeled "Forearms roughly horizontal". The chair is black with a mesh back and wheels, labeled "Chair with adjustable seat height as well as adjustable back height and tilt".

2.3.1. General Safety Guideline

When it comes to general safety guidelines for computer use, here are some key recommendations to ensure a safe and healthy working environment:

1. **Keep the workspace clean and organized:** Maintain a clutter-free workspace to prevent tripping hazards and allow for easy access to equipment. Keep the area around the computer free from liquids and food to avoid spills and damage to the equipment.
2. **Use surge protectors and uninterruptible power supplies (UPS):** Protect your computer and other electronic devices from power surges and outages by using surge protectors and UPS devices. These help prevent damage to the equipment and data loss.
3. **Regularly back up important data:** Implement a regular backup system to ensure that important files and data are securely stored. This protects against data loss in the event of hardware failure, malware attacks, or other unforeseen circumstances.
4. **Use reputable antivirus software:** Install and regularly update antivirus software to protect your computer from malware, viruses, and other security threats. Keep the software definitions up to date and perform regular scans.
5. **Be cautious when opening email attachments or clicking on links:** Exercise caution when opening email attachments or clicking on links, especially from unknown or suspicious sources. These can potentially contain malware or phishing attempts. Verify the sender's identity before downloading or opening any file.
6. **Create strong and unique passwords:** Use strong, complex passwords for all your accounts and avoid using the same password for multiple accounts. Consider using a password manager to securely store and manage your passwords.
7. **Be mindful of ergonomics:** Maintain proper posture and ergonomics while working at the computer. Sit with your back straight, feet flat on the floor, and wrists in a neutral position. Adjust your chair, desk, and monitor height to ensure a comfortable and ergonomic setup.
8. **Take regular breaks:** Take regular breaks from computer work to prevent eye strain, fatigue, and musculoskeletal issues. Follow the 20-20-20 rule: Every 20 minutes, look away from the screen and focus on something 20 feet away for at least 20 seconds.
9. **Adjust display settings:** Adjust the brightness, contrast, and font size on your computer screen to ensure optimal visibility and reduce eye strain. Use screensavers or power-saving settings to prevent screen burn-in.

10. **Secure your computer and accounts:** Use strong passwords or biometric authentication to lock your computer when not in use. Enable automatic updates for your operating system and software to ensure you have the latest security patches.
11. **Be cautious of social engineering attacks:** Be vigilant against social engineering attacks, such as phishing attempts or phone scams. Be skeptical of unsolicited requests for personal or financial information and verify the authenticity of requests before providing any sensitive data.
12. **Report any suspicious activity or security incidents:** If you notice any unusual activity, security breaches, or suspicious behavior on your computer or network, report it to your IT department or system administrator immediately.

By following these general safety guidelines, you can promote a secure and productive computer usage environment while minimizing potential risks and ensuring your own safety and the safety of your data.

2.3.2. Fire Safety Guideline

Fire safety is crucial when it comes to computer use to prevent accidents, protect equipment, and ensure the safety of individuals in the vicinity. Here are some fire safety guidelines for computer users:

1. Electrical safety:

- Use surge protectors and avoid overloading electrical outlets.
- Regularly inspect power cords and plugs for any signs of damage or wear. Replace damaged cords immediately.
- Do not run cables or cords under carpets or rugs, as this can cause overheating.
- Avoid using extension cords for computer equipment whenever possible.

2. Proper ventilation:

- Ensure that computer equipment, including desktop computers and servers, are placed in well-ventilated areas and not obstructed by objects that could restrict airflow.
- Avoid placing items on top of or around computer equipment that could block ventilation and cause overheating.

3. No smoking policy:

- Prohibit smoking in areas where computers and electronic equipment are present to prevent the risk of fire caused by discarded cigarette butts or ashes.

4. Adequate fire protection and suppression:

- Install smoke detectors in the vicinity of computer equipment and ensure they are regularly tested and maintained.
- If feasible, install fire suppression systems, such as automatic sprinklers, in areas with computer servers or other critical equipment.

5. Knowledge of evacuation procedures:

- Familiarize yourself with the fire evacuation procedures of your workplace or building. Know the location of fire exits, assembly points, and fire extinguishers.
- Ensure that pathways to fire exits are clear and unobstructed.

6. Fire extinguisher awareness:

- Know the locations of fire extinguishers in your workspace and understand how to use them correctly.
- If you notice a small fire that can be safely extinguished, use the appropriate fire extinguisher and follow the PASS method: Pull the pin, Aim at the base of the fire, Squeeze the handle, and Sweep from side to side.

7. Reporting hazards:

- Promptly report any fire hazards, malfunctioning electrical equipment, or potential fire risks to your supervisor, manager, or the appropriate authority in your organization.
- If you observe signs of electrical issues, such as sparks, smoke, or unusual smells, disconnect the equipment and immediately notify the responsible personnel.

8. Backup and off-site storage of data:

- Regularly backup important data and ensure the backups are stored off-site or in a secure location. This protects against data loss in the event of a fire or other emergencies.

Remember, fire safety is a shared responsibility. It is important to follow organizational fire safety protocols, participate in fire drills, and stay vigilant to prevent and respond effectively to fire hazards.



Self-check 2-2:

-
1. What are some general safety guidelines when working with computers?
 2. Explain the importance of fire safety in a computer environment.
-

2.3.3. Equipment Protection

Certainly! Here's a breakdown of equipment protection, including ESD protection, types of power fluctuations, and power protection devices:

2.3.3.1.ESD Protection:

ESD (Electrostatic Discharge) is a sudden flow of electricity between two objects caused by a static charge imbalance. It can damage sensitive electronic components. To protect equipment from ESD:

- **Grounding:** Ensure that all equipment, including workstations and personnel, are properly grounded to prevent static buildup.
- **ESD wrist straps:** Use antistatic wrist straps when handling sensitive electronic components to dissipate static charges.
- **ESD mats and flooring:** Install ESD mats on work surfaces and use ESD flooring in areas where static-sensitive equipment is handled.
- **ESD bags and containers:** Store and transport sensitive equipment/components in ESD bags or containers to shield them from static charges.

2.3.3.2.Types of Power Fluctuations:

Power fluctuations can be categorized into the following types:

- **Voltage Sag:** A short-term drop in voltage, typically caused by sudden increases in power demand or faults in the power distribution system.
- **Voltage Surge:** A brief increase in voltage, often caused by lightning strikes, switching operations, or faults in the power grid.

- **Voltage Spike:** A rapid increase in voltage that lasts for a short duration, usually caused by switching off high-powered electrical devices or sudden load changes.
- **Voltage Interruption:** A complete loss of power supply for a brief period due to faults, outages, or tripped circuit breakers.
- **Voltage Transient:** Sudden, brief fluctuations in voltage caused by electrical disturbances, such as lightning, electromagnetic interference (EMI), or switching operations.

2.3.3.3. Power Protection Devices:

To safeguard equipment against power fluctuations and electrical disturbances, various power protection devices can be used:

- **Surge Protectors:** These devices absorb excess voltage spikes and redirect them away from connected equipment. They are commonly used for protecting computers, peripherals, and other sensitive electronics.
- **Uninterruptible Power Supplies (UPS):** UPS systems provide backup power during outages and protect against voltage sags, surges, and interruptions. They typically include surge protection, battery backup, and voltage regulation features.
- **Voltage Regulators:** Voltage regulators stabilize incoming voltage to ensure a steady supply to connected equipment, compensating for voltage sags and surges.
- **Isolation Transformers:** These transformers isolate equipment from the main power supply, protecting against voltage spikes and transient noise.
- **Line Conditioners:** Line conditioners regulate voltage, filter out noise, and provide surge protection to ensure clean and stable power supply to sensitive equipment.
- **Automatic Voltage Regulators (AVR):** AVR devices monitor voltage levels and automatically adjust them to maintain a stable output voltage within a specified range.
- **Power Filters:** Power filters remove noise and interference from the power supply to protect sensitive equipment from electrical disturbances.

Implementing appropriate ESD protection measures and using power protection devices can help safeguard equipment, prevent damage from power fluctuations, and ensure the longevity of electronic devices.



Self-check 2-3:

1. What is ESD protection, and why is it important for computer equipment?
2. List and explain the different types of power fluctuations that can affect computer systems.

2.4. Environment Protection

Proper disposal procedures in a computer perspective refer to the appropriate and environmentally responsible methods of getting rid of computer equipment and electronic waste. This includes recycling programs, engaging authorized e-waste recyclers, securely erasing data, considering donation or resale options, utilizing manufacturer take-back programs, disassembling components for recycling, handling hazardous materials, and complying with relevant regulations. These procedures aim to minimize environmental impact, prevent pollution, and promote sustainable practices in the disposal of computer equipment.

2.4.1. Proper Disposal Procedures in a Computer Perspective

Proper disposal of computer equipment is important to prevent environmental pollution and comply with regulations. Here are some guidelines for environmentally friendly disposal of computer equipment:

1. **Recycling Programs:** Many countries and organizations have recycling programs specifically designed for electronic waste (e-waste). Research and identify local recycling facilities or programs that accept computer equipment.
2. **Authorized E-Waste Recyclers:** Ensure that you engage with authorized e-waste recyclers who follow proper disposal practices. These recyclers are equipped to handle and process electronic waste in an environmentally friendly manner.
3. **Data Security:** Before disposing of any computer equipment, ensure that all data is securely erased to protect sensitive information. Use data wiping tools or software that comply with industry standards for data sanitization.
4. **Donation or Resale:** Consider donating or reselling computer equipment that is still functional and in good condition. Many organizations or individuals may benefit from used computers, extending their lifespan and reducing waste.

5. **Manufacturer Take-Back Programs:** Check if the computer equipment manufacturer offers take-back programs or recycling initiatives. Some manufacturers have programs in place to responsibly collect and recycle their products.
6. **Disassembly and Component Separation:** If you have the necessary knowledge and expertise, you can disassemble computer equipment and separate different components for recycling. This approach allows for the proper recycling of individual components and materials like metals, plastics, and circuit boards.
7. **Hazardous Materials:** Computers and other electronic devices may contain hazardous materials such as lead, mercury, and flame-retardant chemicals. It is essential to handle and dispose of these materials according to local regulations to prevent environmental contamination. Contact local authorities or recycling centers for guidance on handling hazardous materials.
8. **Compliance with Regulations:** Familiarize yourself with local, regional, and national regulations regarding the disposal of electronic waste. Follow the guidelines and requirements for proper disposal to ensure compliance with environmental laws.

Remember, improper disposal of computer equipment can have detrimental effects on the environment and human health. By following proper disposal procedures and recycling practices, you contribute to a more sustainable and responsible approach to managing electronic waste.



Self-check 2-4:

-
1. Why is proper disposal of computer equipment important for the environment?
 2. What are some recommended disposal procedures for computer equipment?
-

Unit Summary

This set of topics covers essential aspects of computer systems, including hardware and network peripherals, safe work conditions, equipment protection, and environment protection. It provides an introduction to computers, their generations, characteristics, classifications, and applications. It also explores the organization of computer systems, including the use of ports and cables. Safety guidelines are outlined, covering general safety and fire safety. Additionally, equipment protection is discussed, highlighting the importance of ESD protection, types of power fluctuations, and power protection devices. Lastly, proper disposal procedures for computer equipment are emphasized to promote environmental responsibility.

Units Review Questions

1. Which of the following is NOT part of the introduction to computers?
 - a) Introduction to computer hardware
 - b) Generation of computers
 - c) Characteristics of computer systems
 - d) Types of computer viruses
2. What is the purpose of computer ports and cables?
 - a) To connect peripherals and devices to a computer
 - b) To protect the computer from power fluctuations
 - c) To dispose of computer equipment properly
 - d) To prevent environmental damage
3. What does ESD protection refer to?
 - a) Protection against power fluctuations
 - b) Protection against environmental damage
 - c) Protection against electrostatic discharge
 - d) Protection against computer viruses
4. What are the types of power fluctuations that can affect computer equipment?
 - a) Fire and water damage
 - b) Electrostatic discharge and power surges
 - c) Virus attacks and malware infections
 - d) Recycling and disposal procedures
5. Which of the following is a guideline for maintaining general safety when working with computers?
 - a) Proper disposal procedures
 - b) Fire safety guidelines
 - c) Using ESD protection measures
 - d) Recycling computer equipment

6. How can you connect hardware peripherals to a computer?
 - a) Using computer ports and cables
 - b) Using fire safety guidelines
 - c) Using proper disposal procedures
 - d) Using ESD protection measures
7. Which of the following is an example of a computer port?
 - a) USB port
 - b) Fire extinguisher port
 - c) Recycling bin port
 - d) ESD protection port
8. What is the purpose of power protection devices?
 - a) To connect computer peripherals
 - b) To protect against power fluctuations
 - c) To dispose of computer equipment properly
 - d) To classify computers based on their generation
9. What is the importance of safe work conditions and procedures?
 - a) To protect computer equipment from ESD
 - b) To prevent environmental damage
 - c) To ensure personal safety when working with computers
 - d) To dispose of computer equipment properly
10. What do proper disposal procedures refer to in the context of environment protection?
 - a) Using ESD protection measures
 - b) Recycling computer equipment responsibly
 - c) Connecting peripherals to computer ports
 - d) Protecting against power fluctuations

Explain

1. Provide examples and describe the different classifications of computers.
2. How are computers used in various applications?
3. What steps should be taken to prevent and handle fire incidents in a computer setup?
4. Describe the purpose and functionality of power protection devices such as surge protectors, UPS systems, and voltage regulators.
5. How can hazardous materials in computer equipment be handled safely during disposal?

Practical Demonstration

Based on general safety Guideline perform the following activities

Task 1: - Identify and Demonstrate Peripheral devices

Task 2:- Connect required Peripheral devices to the system Unit

Note:-Your teacher will evaluate your output either satisfactory or unsatisfactory. If unsatisfactory, your teacher shall advice you on additional work. But if satisfactory, you can proceed to the next topic.

UNIT 3

OPERATING SYSTEM AND APPLICATION SOFTWARE

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 3.1.Operating Systems and Installation
- 3.2.Application Software Installation
- 3.3.Installing Devices Driver

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Install operating system
- ✓ Identify Application software types

Key Terms: *Operating system (OS), Installation, Application software, Devices Driver*

3.1. Operating Systems and Installation

An operating system (OS) is software that manages computer hardware and software resources and provides services for computer programs. It acts as an intermediary between the user and the computer hardware, enabling the execution of applications and coordinating various system components.

3.1.1. Introduction to an operating system

This unit focuses on the installation of operating systems, application software, and device drivers. It begins with an introduction to operating systems, covering their purpose and functions. The types of operating systems are explained, including examples such as Windows, macOS, and Linux. The process of installing an operating system is discussed, highlighting the steps involved.

Next, the unit moves on to application software installation. Different types of application software are explored, such as productivity software, multimedia software, and specialized software. The concept of software licensing is introduced, including proprietary, open-source, and freeware licenses. The importance of accessing minimum and maximum hardware and operating system requirements is emphasized when installing application software. The unit also covers the installation, configuration, and launching of office applications. Additionally, the processes of updating and removing software are explained.

The unit concludes with the topic of installing device drivers. The purpose and use of drivers are described, emphasizing their role in facilitating communication between hardware devices and the operating system. Different methods of driver installation are discussed, such as automatic installation through operating system updates or manual installation using driver installation files. The importance of regularly checking for installed drivers and updates is highlighted to ensure optimal device performance.

Overall, this unit provides a comprehensive overview of the installation processes for operating systems, application software, and device drivers, enabling learners to effectively set up and configure computer systems.

3.1.2. Types of operating systems:

There are different types of operating systems, including:

- Windows: Developed by Microsoft and widely used in personal computers.
- macOS: Developed by Apple and used in Mac computers.
- Linux: An open-source operating system available in various distributions, known for its flexibility and customization options.
- UNIX: A multi-user and multitasking operating system used in servers and enterprise environments.



Self-check 3-2:

1. Dear all students let us go and search different windows operating system types from the internet.
2. Present for the class



Please teachers give assignment for the students and ask to present for the class.

3.1.3. Operating system installation:

Operating system installation involves preparing the installation media (such as a DVD or USB drive) and following a step-by-step installation process.

The installation process typically includes selecting the installation type, partitioning the hard drive, specifying the installation location, and configuring system settings.

Practical work: Operating System Installation steps

The steps for installing an operating system may vary depending on the specific operating system you are installing. However, here is a general outline of the steps involved in installing an operating system:

1. Prepare for installation:

- Ensure that you have a compatible hardware setup for the operating system you are installing.
- Back up any important data from the existing operating system if you are performing a clean installation.

2. Prepare the operating system:

- Download the operating system installation files from the official website or obtain a physical installation disc or USB drive.

3. Create installation media (if necessary):

- If you have downloaded the operating system files, create a bootable USB drive or burn the files to a DVD or CD using appropriate software.

4. Configure the BIOS/UEFI settings:
 - Restart your computer and access the BIOS or UEFI settings by pressing the designated key (usually displayed during the boot process).
 - Configure the boot order to prioritize the installation media (USB or DVD drive) to boot from.
5. Start the installation process:
 - Insert the installation media into the computer and restart it.
 - Follow the on-screen prompts to start the installation process.
6. Select language and region settings:
 - Choose your preferred language and region settings for the operating system.
7. Accept the license terms:
 - Read and accept the license agreement for the operating system.
8. Choose the installation type:
 - For a clean installation, select the option to install the operating system on a new or formatted partition.
 - For an upgrade installation, choose the option to upgrade the existing operating system.
9. Select the installation location:
 - Choose the disk or partition where you want to install the operating system.
10. Customize installation settings (if available):
 - Some operating systems may provide additional customization options, such as selecting specific features or components to install.
11. Wait for the installation to complete:
 - The installation process may take some time, during which the operating system files will be copied and configured on your computer.
12. Set up user accounts and preferences:
 - Follow the prompts to create a user account and set up additional preferences, such as computer name, password, network settings, etc.
13. Complete the installation:
 - Once the installation is finished, the computer may restart.
 - Remove the installation media (USB or DVD) when prompted.

14. Install device drivers and updates:

- After the operating system installation is complete, install necessary device drivers for hardware components, such as graphics cards, sound cards, etc.
- Apply any available operating system updates and security patches.

15. Restore data and settings (if applicable):

- If you performed a clean installation, restore your backed-up data and settings to the newly installed operating system.

16. Restart the computer:

- Restart the computer to ensure that all changes and settings are applied correctly.

These steps provide a general overview of the installation process, but it's important to refer to the specific documentation or instructions provided by the operating system manufacturer for detailed guidance.

Windows 10 Installation steps

1. Create Windows 10 USB with Media Creation Tool

To create an installation media using the Media Creation Tool, connect a USB flash drive of at least 8GB of space, and use these steps:

- A. Open the Windows 10 download page.
- B. Select the "Create installation media (USB flash drive, DVD, or ISO file) for another PC" option.

2. Windows 10 clean install process

After the preparation, you can proceed with the clean installation of Windows 10.

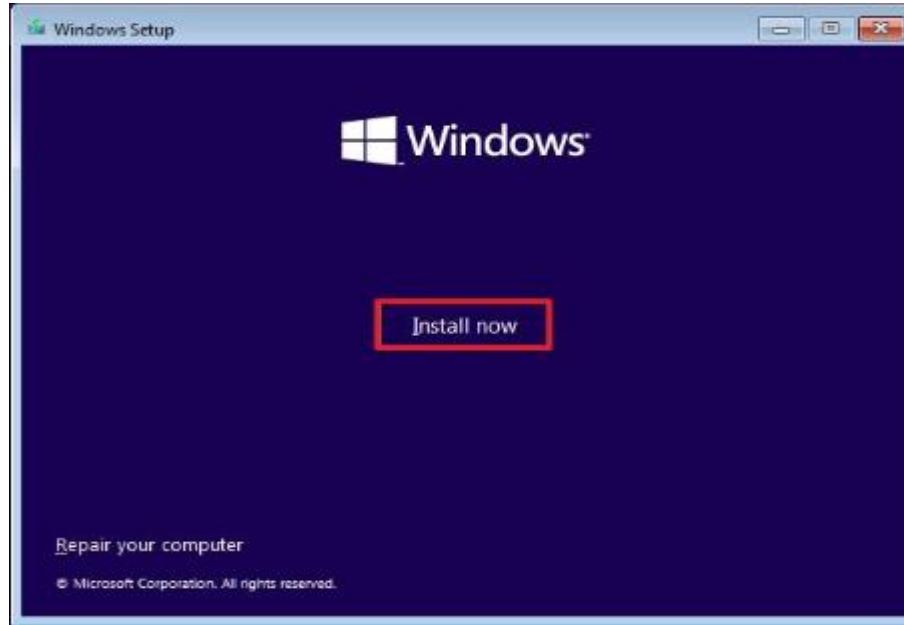
To do a clean installation of Windows 10, use these steps:

1. Start the PC with Windows 10 USB media.
2. On prompt, press any key to boot from the USB flash drive.
3. On the "Windows Setup," click the Next button.

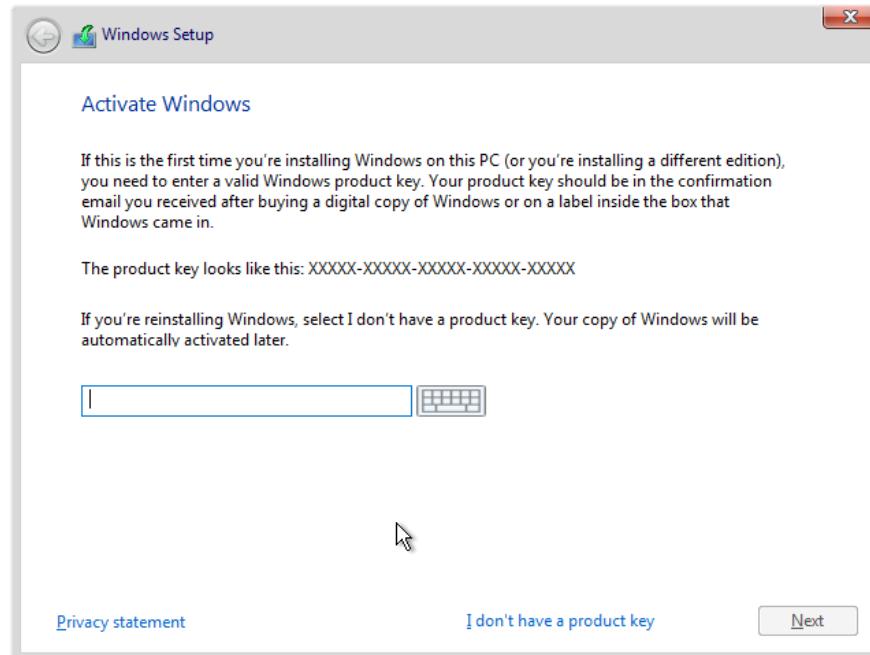
Quick tip: If your language, time and currency, and keyboard are different from the default selections, make sure to select the correct settings.



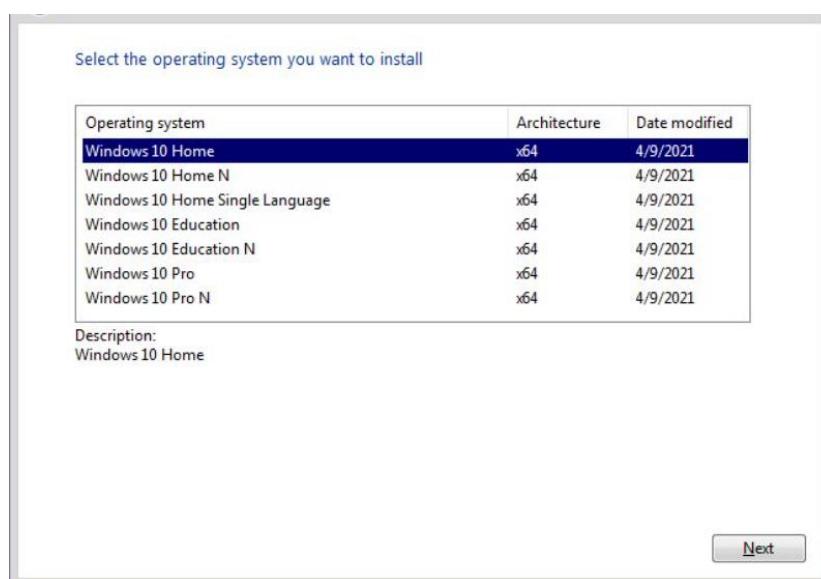
4. Click the **Install now** button



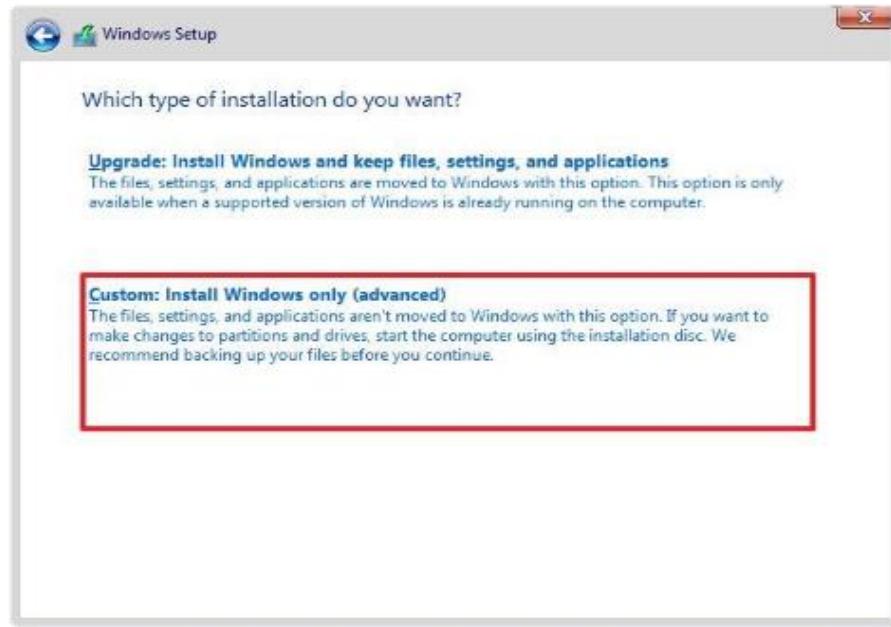
5. On a reinstallation, click the "I don't have a product key" option to continue (assuming your device was already activated). Otherwise, if this is your first time installing Windows 10, enter the product key.



6. Click the Next button.
 7. Select the edition of Windows 10 (If applicable).
- **Quick note:** The selection has to match the edition of your product key. If you don't select the correct edition, Windows 10 won't activate, and you'll have to redo the entire process.

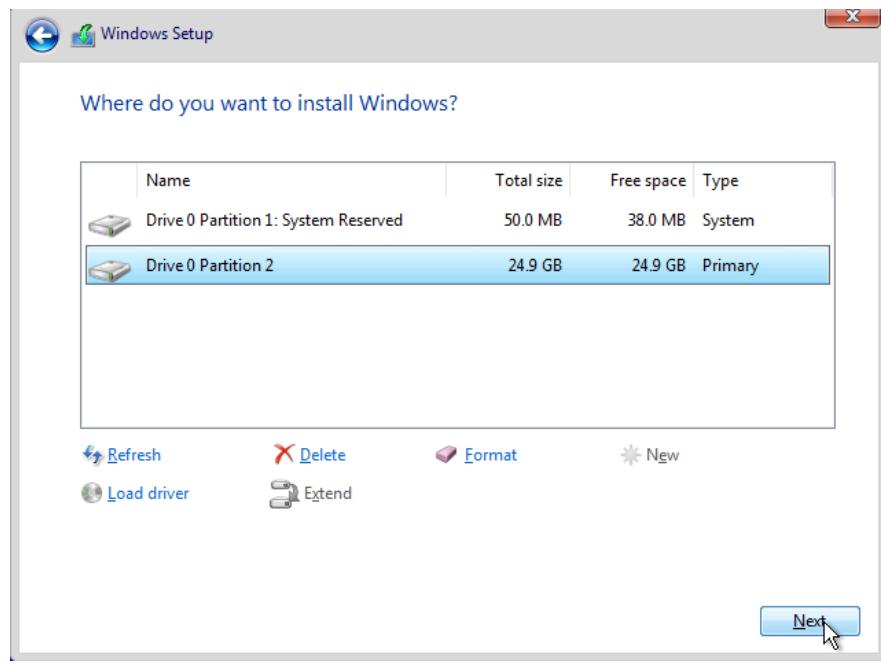


8. Click the Next button.
9. Select the "I accept the license terms" option to continue.
10. Click the Next button.
11. Click the "Custom: Install Windows only (Advanced)" option to continue with a clean installation.

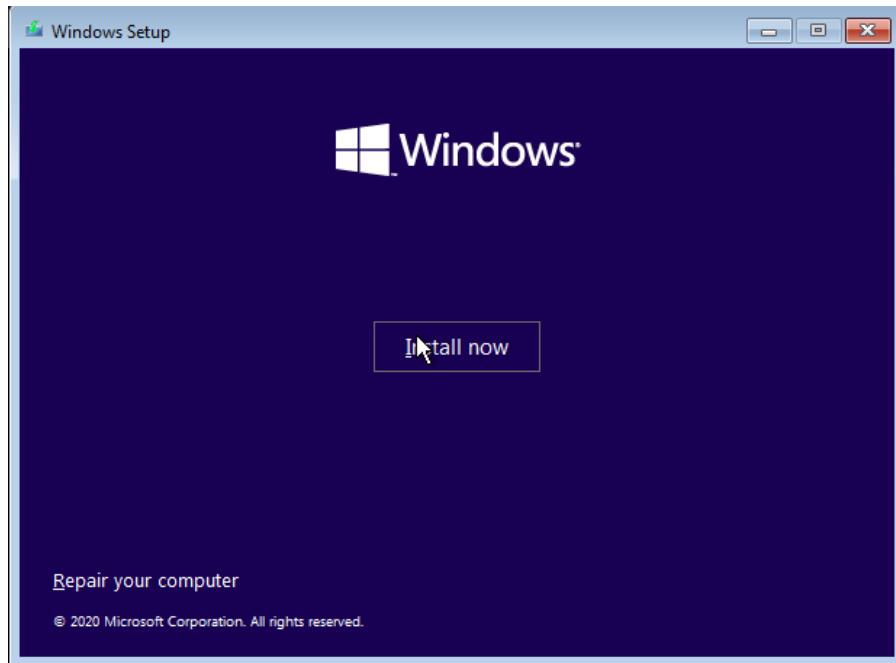


12. Select the partition with the current installation (usually "Drive 0") and click the Delete button.
- **Quick tip:** If "Drive 0" has multiple partitions, you must select and delete each partition to allow the setup to use the entire hard drive for the new clean installation. The setup will create the required partitions automatically during the process. Also, it's not necessary to delete the partitions on secondary drives.

13. Click the Yes button.



14. Select the empty drive (Drive 0 Unallocated Space).



15. Click the Next button.

Once you complete the steps, the setup will install a fresh copy of Windows 10 on the device.

3.2. Application Software Installation

3.2.1. Types of Application software:

Application software refers to programs designed for specific tasks or purposes, such as word processing, spreadsheet management, graphic design, and multimedia editing.

There are several types of application software, each designed to fulfill specific purposes and cater to different user needs. Here are some common types of application software:

Word Processing Software: This type of software allows users to create, edit, and format text documents. Examples include Microsoft Word, Google Docs, and Apple Pages.

Spreadsheet Software: Spreadsheet software enables users to manage and analyze numerical data in the form of tables. It provides functions for calculations, data visualization, and data manipulation. Examples include Microsoft Excel, Google Sheets, and Apple Numbers.

Presentation Software: Presentation software is used to create visually appealing slideshows or presentations. Users can add text, images, charts, and multimedia elements to convey information effectively. Examples include Microsoft PowerPoint, Google Slides, and Apple Keynote.

Database Software: Database software helps users store, manage, and organize large amounts of structured data efficiently. It allows for data entry, querying, and generating reports. Examples include Microsoft Access, MySQL, and Oracle Database.

Graphics and Design Software: Graphics and design software are used to create and edit visual content, such as images, illustrations, and layouts. They provide tools for image editing, vector graphics, and page layout. Examples include Adobe Photoshop, Adobe Illustrator, and CorelDRAW.

Web Browsers: Web browsers enable users to access and navigate websites on the internet. They provide a graphical interface for browsing web pages, displaying text, images, videos, and interactive content. Examples include Google Chrome, Mozilla Firefox, and Microsoft Edge.

Communication Software: Communication software facilitates communication between individuals or groups. It includes email clients, instant messaging applications, video conferencing tools, and collaboration platforms. Examples include Microsoft Outlook, Gmail, Skype, and Slack.

Multimedia Software: Multimedia software allows users to create, edit, and play various types of media content, such as audio, video, and images. It includes media players, video editors, audio editing software, and graphic editing tools. Examples include Adobe Premiere Pro, Windows Media Player, and Audacity.

Utility Software: Utility software provides essential tools for system maintenance, optimization, and security. It includes antivirus software, disk cleaners, file compression tools, and backup software. Examples include Norton Antivirus, CCleaner, and WinRAR.

Educational Software: Educational software is designed to support teaching and learning processes. It includes interactive learning programs, language learning software, educational games, and simulation tools. Examples include Rosetta stone, Moodle, and Scratch.

These are just a few examples of the diverse range of application software available. The specific applications and their functionalities may vary depending on the operating system and the specific needs of the users.

3.2.2. Types of Software licensing:

Software licensing determines the terms and conditions under which software can be used. Common licensing types include proprietary licenses (commercial software), open-source licenses (freely available source code), and freeware licenses (free to use without restrictions).

3.2.3. Accessing minimum and maximum hardware and operating system requirements:

Before installing application software, it is important to check the minimum and maximum hardware requirements specified by the software vendor.

These requirements include processor speed, RAM, storage space, and operating system version compatibility.

Example: Confirm Windows 10 requirements

If the device already has Windows 10, the following major update will likely meet the minimum hardware requirements. However, if you're about to upgrade a computer running Windows 7 or Windows 8.1, you want to check that the device meets the minimum requirements.

Here are the minimum requirements to install Windows 10 versions 21H2, 22H2, and higher:

- **Processor:** 1GHz or System on a Chip (SoC).
- **Memory:** 1GB for 32-bit or 2GB for 64-bit.
- **Storage:** 32GB for 64-bit or 32-bit.
- **Graphics card:** DirectX 9 or later with WDDM 1.0 driver.
- **Display:** 800x600.

You can use these instructions to check the device technical specifications to confirm you can do a clean install of Windows 10.

3.2.4. Install, configure, and launch office applications:

The installation process for office applications typically involves running the installation file, following on-screen prompts, and entering license information if required.

After installation, configuring the application involves setting preferences, customizing settings, and connecting to external services if necessary.

Launching office applications can be done by clicking on their respective icons or through the Start menu or application launcher.

3.2.5. Update and Remove Software:

Regular software updates are essential to ensure security, stability, and access to new features.

Updates can be obtained through official software update mechanisms or by downloading and installing updates from the software vendor's website.

Software removal can be done through the operating system's built-in uninstallation process or using dedicated uninstaller programs.

3.3. Installing Device Drivers

Installing device drivers is the process of installing software that enables communication between the operating system and hardware devices. It involves downloading or using the installation CD provided with the device, running the installer, and following on-screen instructions. This ensures that the computer can effectively utilize the connected hardware device and ensure proper functionality.

3.3.1. Drivers and their use:

Device drivers are software programs that enable communication between hardware devices and the operating system.

They allow the operating system to understand and utilize the specific features and capabilities of different hardware components.

3.3.2. Driver installation methods:

There are different methods for installing device drivers, depending on the operating system and the device being installed. Here are a few common methods:

1. Automatic installation through operating system updates: Many operating systems, such as Windows, have built-in mechanisms to automatically detect and install drivers for compatible hardware devices. These drivers are often obtained through regular system updates.

2. Manual installation using driver installation files:

Device manufacturer's website: Most hardware manufacturers provide driver downloads on their websites. You can visit the manufacturer's support or downloads section, Find the appropriate driver for your device and operating system, and download the driver installation file (usually in the form of an executable or archive file).

Device manager: In Windows, you can access the Device Manager by right-clicking on the Start button and selecting "Device Manager." From there, Find the device you want to install a driver for, right-click on it, and select "Update driver." You can choose to search for drivers automatically or browse your computer for driver software. If you have the driver installation file, you can select the "Browse my computer for drivers" option and provide the path to the driver file.

Installing Drivers in Windows 10: A Step-by-Step Guide with Images

Keeping your drivers updated is crucial for optimal performance and security on your Windows 10 machine. Whether you're building a new PC, experiencing issues, or simply want to ensure everything's up-to-date, and here's a comprehensive guide with images:

Method 1: Using Windows Update (Automatic)

- A. Open Settings: Click on the Start menu and select the gear icon.
 - B. Navigate to Update & Security: Choose "Update & security" from the left-hand pane.
 - C. Check for updates: Click on "Check for updates" and wait for the scan to complete.
1. Install drivers: If any driver updates are available, they'll be listed under "Optional updates." Select and install them by clicking "Download and install now."

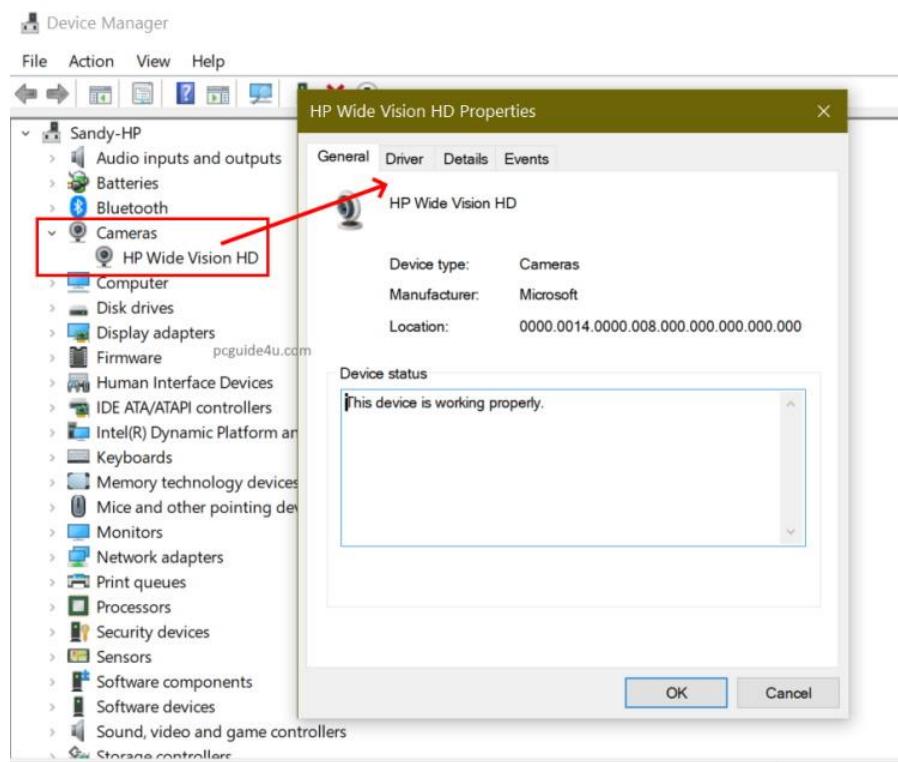
Image 1: Navigating to Update & Security

Image 2: Checking for updates and installing drivers

Opens in a new window

Method 2: Using Device Manager (Manual)

1. Open Device Manager: Right-click the Start menu and select "Device Manager."
2. Expand device categories: Click the arrow next to each category (e.g., Display adapters, Network adapters) to reveal your devices.
3. Identify outdated drivers: Right-click on a device and select "Properties." Go to the "Driver" tab and check the "Driver Date" or "Driver Version."
4. Update drivers: Right-click the device again and choose "Update driver."
5. Search automatically: Select "Search automatically for updated driver software." Windows will search for and install available updates.



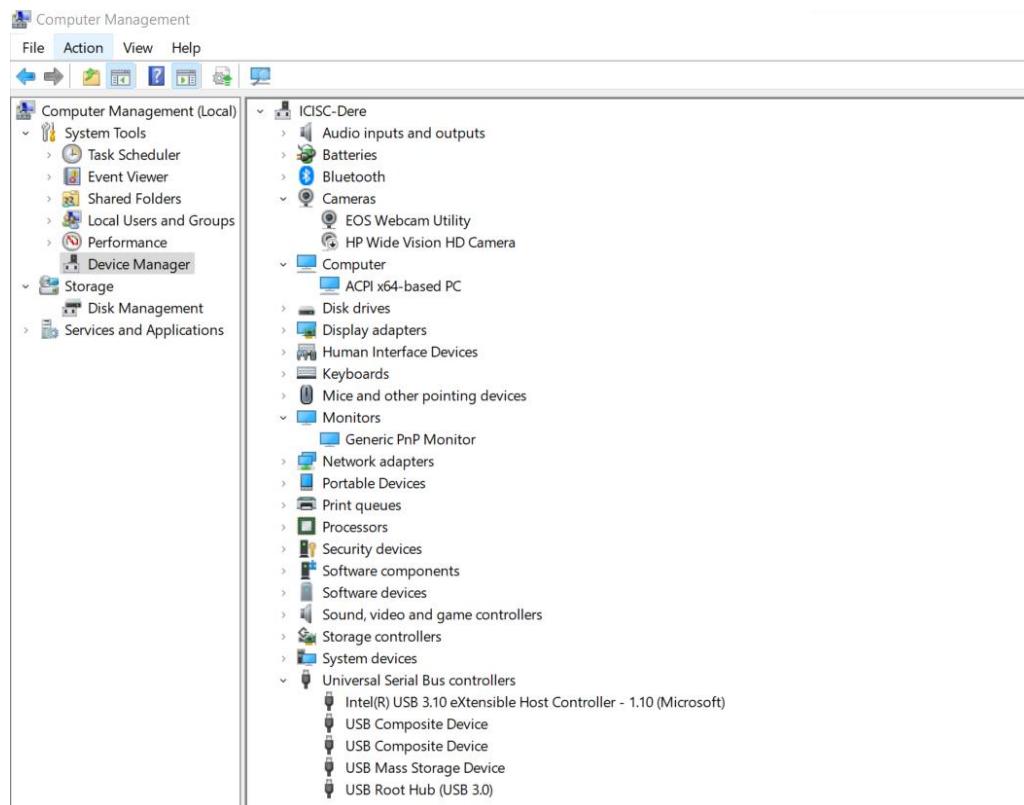


fig 3: Device Manager Window

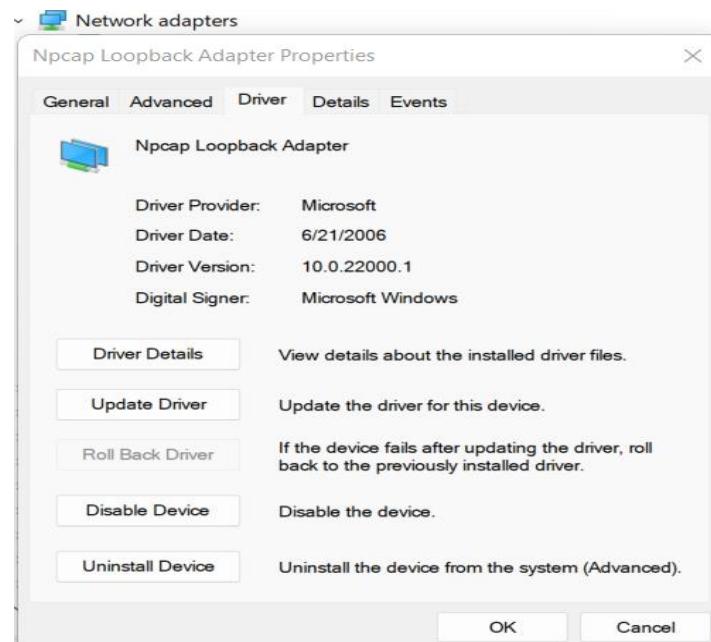


Image 4: Updating drivers manually

Method 3: Using Manufacturer's Website (Recommended)

1. Identify your hardware: Check your device's documentation or use a system information tool to identify the manufacturer and model number of your hardware (e.g., graphics card, network adapter).
2. Visit the manufacturer's website: Navigate to the website of your hardware manufacturer (e.g., NVIDIA, Intel, and Realtek).
3. Find drivers: Search for your specific device model and download the latest drivers compatible with your Windows 10 version.
4. **Install drivers:** Run the downloaded installer and follow the on-screen instructions.

Image 5: Searching for drivers on a manufacturer's website

Additional Tips:

- System restores point: Create a system restore point before updating drivers in case of any issues.
- Driver compatibility: Ensure you download drivers specifically designed for your Windows 10 version and hardware model.
- Third-party driver tools: Use them with caution, as they can sometimes install incompatible or unnecessary drivers.

These steps and choosing the most suitable method, you can effectively install and update drivers on your Windows 10 system, ensuring smooth operation and optimal performance.

3.3.3. Checking installed drivers and updates:

It's important to regularly check for installed drivers and updates to ensure optimal device performance and compatibility. Here's how you can check for installed drivers and updates:

- **Device Manager:** Gear icon with text "Right-click Start > Device Manager."
- **Windows Update:** Globe icon with text "Settings > Update & Security > Check for updates."
- **Manufacturer Website:** Globe/computer icon with text "Visit manufacturer website for specific updates."

By regularly checking for installed drivers and updates, you can ensure that your hardware devices are functioning optimally with the latest driver versions and compatibility improvements.

Unit Summary

The "Install operating system and application software" unit cover the installation and management of operating systems, application software, and device drivers. It includes an introduction to operating systems, different types of operating systems, and the process of operating system installation. It also covers application software installation, software licensing types, hardware and operating system requirements, office application setup, software updates, and removal. Additionally, it provides an understanding of device drivers, driver installation methods, and checking for driver updates. This unit equips users with the necessary knowledge and skills to set up and maintain their computer systems effectively.

Part 1: Practical work

Dear Students! Practice in your Lab – Windows 10 Installation on your Desktop Computer

Unit Review Questions

1. What is the purpose of an operating system?
 - a) To provide hardware components for a computer
 - b) To enable communication between software applications
 - c) To manage computer resources and provide a user interface
 - d) To store and organize data files
2. Which of the following is NOT a type of operating system?

| | |
|------------|---------------------|
| a) Windows | c) Microsoft Office |
| b) Linux | d) macOS |
3. How can you install an operating system on a computer?
 - a) Insert the installation CD and follow the on-screen instructions
 - b) Download the operating system from the internet and run the installer
 - c) Use a USB drive with the operating system image and boot from it
 - d) All of the above

4. What are the types of application software?
 - a) Word processing, spreadsheet, and presentation software
 - b) Graphics and design software
 - c) Web browsers and communication software
 - d) All of the above
5. What does software licensing refer to?
 - a) The process of installing software on a computer
 - b) The legal agreement between the software developer and the user
 - c) The compatibility of software with different operating systems
 - d) The cost of purchasing software
6. How can you check the minimum and maximum hardware requirements for installing software?
 - a) Refer to the system requirements listed on the software package or website
 - b) Contact the software developer's customer support for detailed specifications
 - c) Check the operating system's documentation for compatibility information
 - d) All of the above
7. What is the purpose of device drivers?
 - a) To control the hardware components of a computer
 - b) To provide software applications with additional functionality
 - c) To manage the installation of operating systems
 - d) To connect a computer to the internet
8. How can you install device drivers on a computer?
 - a) Use the Device Manager in the operating system and update drivers
 - b) Visit the manufacturer's website and download the latest driver
 - c) Use Windows Update to automatically install driver updates
 - d) All of the above
9. How can you check for installed drivers and updates?
 - a) Open the Device Manager and view the list of installed drivers
 - b) Use Windows Update to check for driver updates
 - c) Visit the manufacturer's website and check for driver updates
 - d) All of the above

10. What is the purpose of updating software?
- a) To improve software performance and add new features
 - b) To remove the software from the computer
 - c) To increase the software's compatibility with other applications
 - d) To change the software's user interface

UNIT 4

OPERATING SYSTEM ENVIRONMENT

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 4.1. Understand and Customize the Desktop Environment
- 4.2. Manage files and folders
- 4.3. Access Basic system information
- 4.4. Uninstalling and removing Applications
- 4.5. Create and manage user accounts
- 4.6. Essential keyboard shortcuts
- 4.7. Use Windows system tools and applications

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Navigate and use Operating system environment

Key Terms: Desktop Environment, system information, user accounts, applications and keyboard shortcuts.

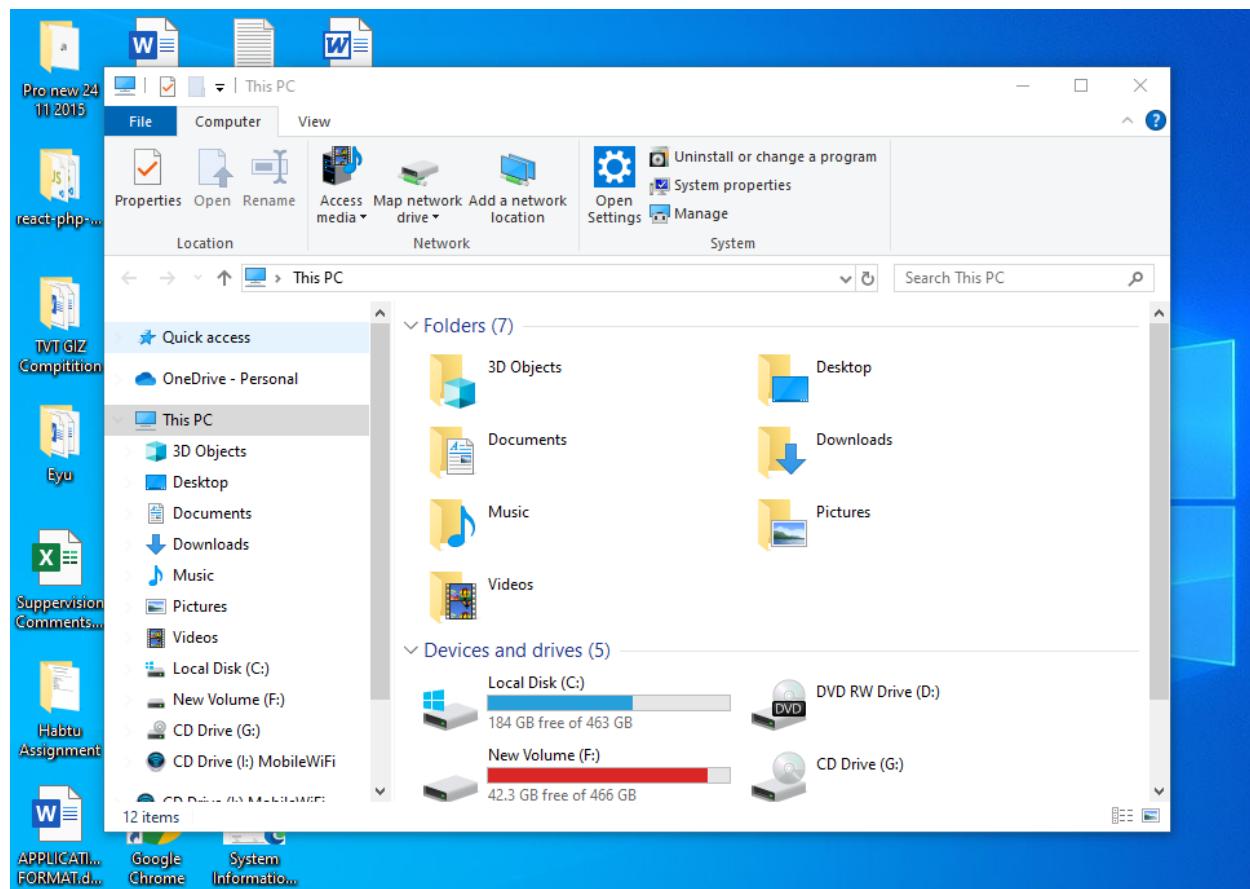
4.1. Navigate and use the Operating System environment

Navigating and using the operating system environment involves familiarizing yourself with the user interface, accessing system features, and performing various tasks. This includes interacting with the desktop, managing files and folders, accessing system information, uninstalling applications, creating and managing user accounts, and utilizing essential keyboard shortcuts.

To navigate the operating system environment, you need to understand and customize the desktop environment, which comprises components such as the desktop wallpaper, icons, and taskbar/dock, start menu/application launcher, window manager, panels/widgets, and file manager.

Managing files and folders involves creating, copying, pasting, moving, and deleting files and folders. Additionally, you can search for specific files and folders to find them quickly.

Accessing basic system information allows you to understand your computer's specifications and configuration. This information includes details about hardware components, software environment, and system settings. You can access system information through system tools or settings menus.



4.2. Desktop Environment

A desktop environment is the graphical user interface (GUI) that allows users to interact with the operating system. It provides a visual representation of the operating system, including icons, menus, windows, and various graphical elements. The components of a desktop environment typically include:

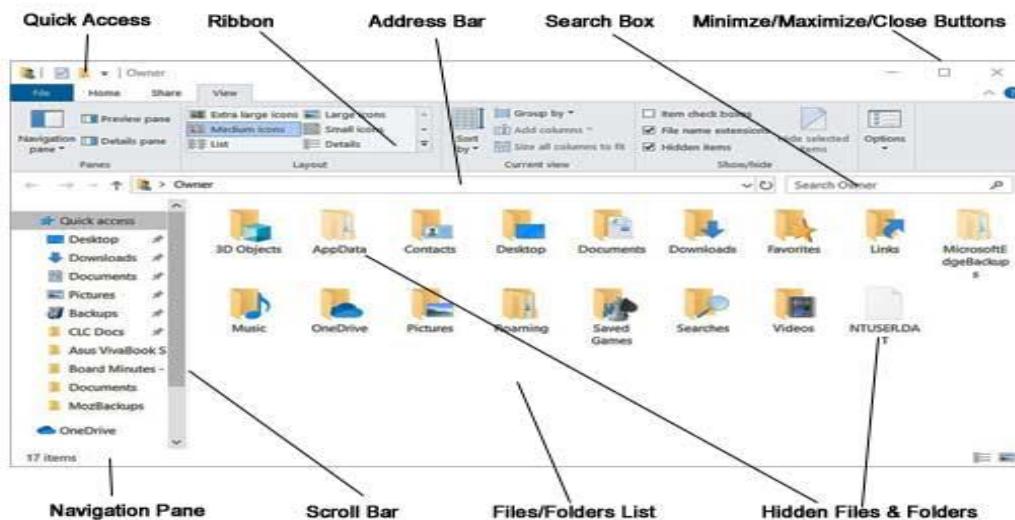
- **Desktop Wallpaper:** The background image or pattern displayed on the desktop.
- **Icons:** Small graphical representations of files, folders, applications, or system functions. Icons provide shortcuts to access and launch various items.

- **Taskbar/Dock:** A horizontal or vertical bar located on the edge of the screen that displays icons or buttons representing running applications, system tray/notification area, and a quick launch area for frequently used applications.
- **Start Menu/Application Launcher:** A menu or button that allows users to access installed applications, system settings, and search for files and programs.
- **Window Manager:** Manages the appearance, placement, and behavior of windows on the desktop. It enables users to resize, move, minimize, maximize, and close windows.
- **Panels/Widgets:** Additional bars or areas on the desktop can display various information. Such as a clock, system status, weather or quick access to specific functions.
- **File Manager:** A tool for navigating, managing, and organizing files and folders on the computer.

Customization options for the desktop environment may vary depending on the operating system and specific desktop environment being used. Users can often personalize the desktop environment by changing wallpaper, icon themes, window decorations, and adjusting settings for panels, menus, and other visual elements.

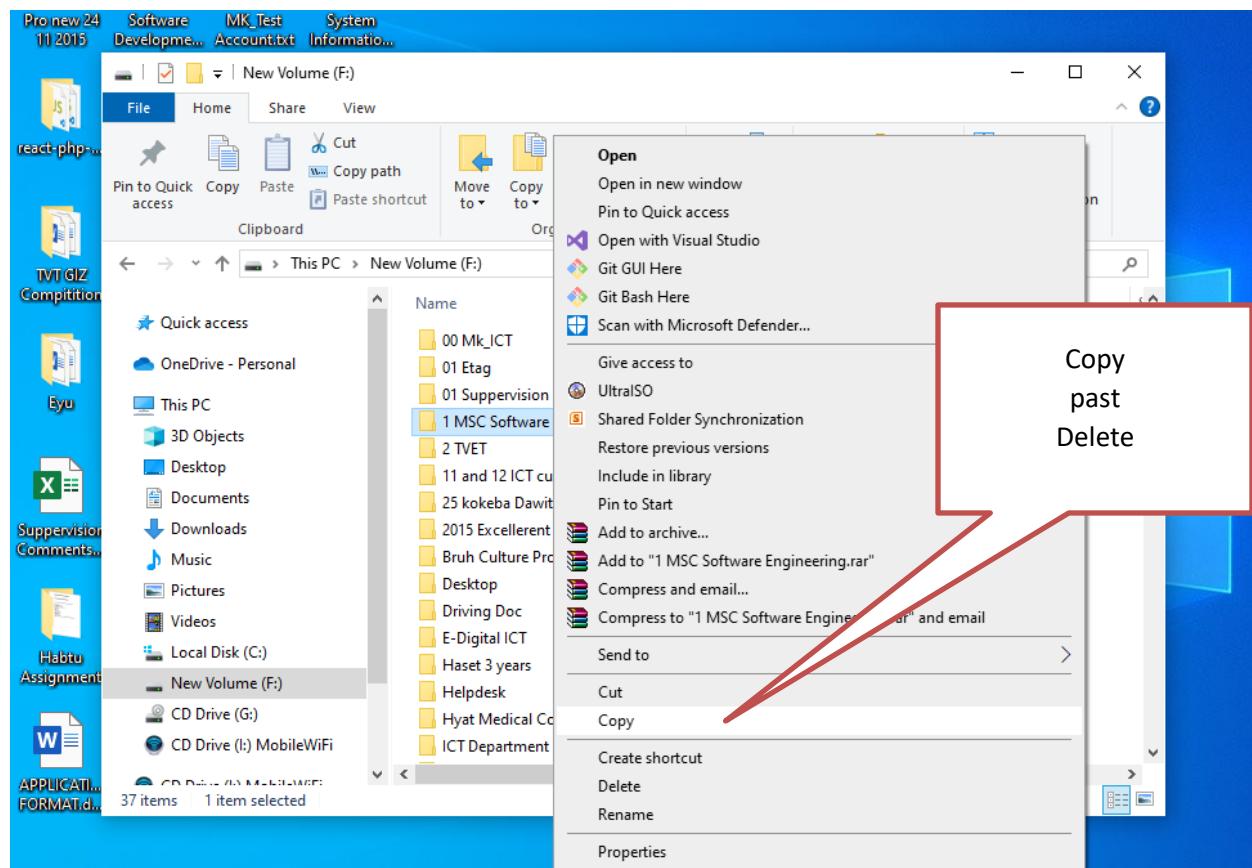
4.3. Manage Files and Folders:

Managing files and folders is an essential aspect of organizing and maintaining a structured digital environment. It involves creating, renaming, moving, copying, and deleting files and folders based on specific requirements. Effective file and folder management ensures easy access, efficient storage, and streamlined workflow. By organizing files and folders in a logical and systematic manner, users can quickly find and retrieve information, reduce clutter, and improve overall productivity. Additionally, employing consistent naming conventions, utilizing descriptive folder structures, and implementing file backup strategies contribute to maintaining data integrity and safeguarding important information.



4.3.1. Create, Copy, Paste, Move, and Delete Files and Folders:

Managing files and folders is an essential part of using an operating system. Here are some common file and folder management tasks:



- Creating Files or Folders:
 - Windows: Right-click in the desired location, select "New," and choose either "Folder" or the specific file type you want to create (e.g., "Text Document").
- Copying Files or Folders:
 - Windows: Select the file(s) or folder(s), right-click, and choose "Copy." Then navigate to the destination location, right-click, and choose "Paste."
- Moving Files or Folders:
 - Windows: Select the file(s) or folder(s), right-click, choose "Cut," navigate to the destination location, right-click, and choose "Paste."
- Deleting Files or Folders:
 - Windows: Select the file(s) or folder(s), press the "Delete" key, and confirm the deletion in the prompt.

4.3.2. Search Files and Folders:

Searching for files and folders can be helpful when you have a large number of items on your computer. Here's how to search for files and folders:

- Windows:
 - File Explorer: Open File Explorer (Windows key + E) and use the search box in the top-right corner to enter your search query. The search results will appear as you type.
 - Start Menu: Click on the Start button or press the Windows key, then start typing your search query. The search results will appear in the Start menu.
- Windows:
 - Using System Information: Press the Windows key + R to open the Run dialog, type "msinfo32" (without quotes), and press Enter. The System Information window will display various details about your system, including hardware components, software environment, and system settings.
 - Using System Settings: Open the Settings app by clicking on the Start button and selecting the gear icon. Go to "System" and then "About." Here, you can find information about your device, such as the processor, RAM, and Windows specifications.

4.4. Uninstalling and Removing Applications

Uninstalling or removing applications is necessary when you want to remove unnecessary programs from your computer. Here are instructions for different operating systems:

Control Panel:

1. Search for "Control Panel" in the Start menu.
2. Go to "Programs and Features".
3. Select the app you want to uninstall and click "Uninstall".

4.5. Create and Manage User Accounts

Group work:-

How many ways are there to break a lost password, to create and delete user accounts prepare a video tutorial/ step by step and present for the class. Please Teacher help to do this

Creating and managing user accounts is important for maintaining separate profiles and access privileges on a computer. Here's how to create and manage user accounts on different operating systems:

Settings:

- Open Settings and go to "Accounts" > "Family & other users".
- Click "Add someone else to this PC".
- Choose a local account or Microsoft account.
- Set a name and password.

Manage User Accounts:

Change account types: - Choose the desired account type from the drop-down menu:

- Standard account: Limited privileges for basic tasks.
- Administrator account: Full control over the system.
- Guest account: Temporary access with limited permissions.

4.6. Essential Keyboard Shortcuts

Keyboard shortcuts are combinations of keys that provide quick access to various functions and features within an operating system. Here are some essential keyboard shortcuts commonly used in Windows:

Practice the following Essential Keyboard Shortcuts for Windows 10

| Function | Shortcut | Notes |
|--------------------------------------|-------------------------------------|---|
| Navigation | | |
| Open Start Menu | Windows key | |
| Minimize all windows | Windows key + M | |
| Restore minimized windows | Windows key + Shift + M | |
| Switch between open apps | Alt + Tab | Cycle through apps in use |
| Switch between windows within an app | Ctrl + Tab | Cycle through open windows within the current app |
| Maximize window | Windows key + Up arrow | |
| Minimize window | Windows key + Down arrow | |
| Close window | Alt + F4 | |
| Move window | Click and hold title bar, then drag | |
| Resize window | Click and hold edge, then drag | |
| File Management | | |
| Open File Explorer | Windows key + E | |
| Search for files | Windows key + S | |
| Cut selected item | Ctrl + X | |
| Copy selected item | Ctrl + C | |
| Paste copied item | Ctrl + V | |
| Undo last action | Ctrl + Z | |
| Redo last action | Ctrl + Y | |
| Delete selected item | Delete key | |
| Rename selected item | F2 | |
| Editing | | |
| Undo last action | Ctrl + Z | |
| Redo last action | Ctrl + Y | |
| Copy selected text | Ctrl + C | |
| Paste copied text | Ctrl + V | |
| Cut selected text | Ctrl + X | |
| Select all text | Ctrl + A | |
| Bold text | Ctrl + B | |
| Italicize text | Ctrl + I | |
| Underline text | Ctrl + U | |
| System Controls | | |

| | | |
|-------------------------------------|-------------------------|---|
| Lock computer | Windows key + L | |
| Sign out of user account | Ctrl + Alt + Delete | Choose "Sign out" option |
| Shut down computer | Alt + F4 | Choose "Shut down" option |
| Take screenshot | Print Screen key | Saves screenshot to clipboard |
| Take screenshot of specific area | Windows key + Shift + S | Select area with cursor, saves screenshot to file |
| Open Task Manager | Ctrl + Shift + Esc | |
| Open Settings | Windows key + I | |
| Additional Useful Shortcuts | | |
| Refresh current window | F5 | |
| Find text in current window | Ctrl + F | |
| Rename a desktop icon | F2 | |
| Open context menu for selected item | Right-click | |
| Show desktop | Windows key + D | |
| Open Run dialog | Windows key + R | |

Note: This table only includes a selection of essential shortcuts. Many other shortcuts are available depending on the specific program or task you are performing. You can usually find a list of shortcuts within the program itself or by searching online.

These are just a few examples, and there are many more keyboard shortcuts available in Windows, depending on the version and installed applications.

4.7. Use Windows system tools and applications

4.7.1. System utility tools:

System utility tools are designed to help you manage and maintain your computer. Some common system utility tools in Windows include:

- **Task Manager:** Allows you to monitor and manage running processes, end unresponsive applications, and view system performance.
- **Disk Cleanup:** Helps you free up disk space by removing unnecessary files and temporary data.
- **Disk Defragmenter:** Optimizes the arrangement of files on your hard drive to improve performance.
- **Device Manager:** Displays and manages the hardware devices installed on your computer.

- **System Configuration:** Allows you to manage startup programs, services, and boot settings.
- Control Panel: Provides access to various system settings and configuration options.

4.7.2. Accessibility tools:

Accessibility tools are designed to enhance the usability and accessibility of the operating system for users with disabilities. Windows provides several built-in accessibility tools, including:

- **Magnifier:** Enlarges a portion of the screen to make it easier to see.
- **Narrator:** Reads aloud on-screen text and provides audio descriptions of events.
- **On-Screen Keyboard:** Displays a virtual keyboard on the screen for users who cannot use a physical keyboard.
- **High Contrast:** Adjusts the color scheme of the display to improve visibility for users with visual impairments.
- **Speech Recognition:** Allows users to control their computer using voice commands.

4.7.3. Basic built-in applications:

Windows comes with a set of basic built-in applications that cover common tasks. These include:

- **Notepad:** A simple text editor for creating and editing plain text files.
- **Paint:** A basic image editing tool for creating and modifying images.
- **Calculator:** A calculator application for performing basic arithmetic calculations.
- **Windows Media Player:** A multimedia player for playing audio and video files.
- **Microsoft Edge:** The default web browser in Windows, used for browsing the internet.

These built-in applications provide essential functionality, but you can also install additional applications from the Microsoft Store or other sources to expand your software options.

Note that the availability and features of these tools and applications may vary depending on the version of Windows you are using.

Practical Work:

1. Dear all students let us go to identify system utility tools, accessibility tools and built in application on windows.



Please teacher form in group and evaluate the students as per the presentation

Unit Summary

In order to navigate and use the operating system environment effectively, users need to understand and customize the desktop environment. This includes familiarizing themselves with the components of a desktop environment, such as the taskbar, icons, menus, and wallpaper, and customizing them to suit their preferences. Managing files and folders is another important aspect, involving tasks such as creating, copying, and pasting, moving, and deleting files and folders, as well as searching for specific files or folders. Basic system information can be accessed to gain insights into the computer's specifications and performance. Uninstalling and removing applications that are no longer needed is also essential to free up disk space and maintain system efficiency. Creating and managing user accounts allows for different individuals to have personalized settings and access controls. Learning and utilizing essential keyboard shortcuts can greatly enhance productivity. Windows system tools and applications, including system utility tools, accessibility tools, and basic built-in applications, provide additional functionality for various tasks and requirements. By mastering these skills, users can effectively navigate and utilize the operating system environment.

Unit Reviews Questions

1. Which of the following are components of a desktop environment?
 - a) Taskbar, icons, menus, and wallpaper
 - b) File Explorer, Control Panel, and Start menu
 - c) Browser, email client, and media player
 - d) Keyboard, mouse, and monitor

2. How can you create a new folder in the operating system?
 - a) Right-click on the desktop and select "New Folder"
 - b) Press Ctrl + N on the keyboard
 - c) Open File Explorer, right-click in the desired location, and select "New" > "Folder"
 - d) Press Alt + F4 on the keyboard
3. Which action allows you to search for specific files and folders in the operating system?
 - a) Ctrl + C
 - b) Ctrl + V
 - c) Ctrl + F
 - d) Ctrl + X
4. What does accessing basic system information provide?
 - a) Insights into the computer's specifications and performance
 - b) Access to the internet
 - c) Ability to install new applications
 - d) Control over user accounts
5. How can you uninstall an application from the operating system?
 - a) Delete the application's shortcut from the desktop
 - b) Drag the application's icon to the Recycle Bin
 - c) Open Control Panel, go to "Programs" or "Apps & Features," and select the application to uninstall
 - d) Press Ctrl + Alt + Delete on the keyboard
6. Why creating and managing user accounts is important?
 - a) It allows for personalized settings and access controls
 - b) It speeds up the computer's performance
 - c) It provides additional storage space
 - d) It improves internet connectivity
7. Which of the following is an essential keyboard shortcut for copying selected files or folders?

| | |
|-------------|-------------|
| a) Ctrl + C | c) Ctrl + V |
| b) Ctrl + X | d) Ctrl + D |

8. What are system utility tools in the operating system used for?
 - a) Customizing the desktop environment
 - b) Managing files and folders
 - c) Optimizing system performance and troubleshooting issues
 - d) Accessing basic system information
9. What is the purpose of accessibility tools in the operating system?
 - a) To enhance security and protect user data
 - b) To improve user interface design and aesthetics
 - c) To provide additional features for users with disabilities
 - d) To manage and organize files and folders
10. Which of the following is an example of a basic built-in application in the operating system?

| | |
|--------------------|----------------------|
| a) Microsoft Word | c) Internet Explorer |
| b) Adobe Photoshop | d) Calculator |

Answer Key for Self-Check Questions

Answers for Self-check 1-1:

1. The role of HTML in front-end development is to structure the content of a web page. HTML (Hypertext Markup Language) provides the basic building blocks, or elements, that define the different parts of a webpage, such as headings, paragraphs, images, links, and forms.
2. CSS (Cascading Style Sheets) contributes to the visual styling of a website by controlling the presentation and layout of HTML elements. CSS allows developers to define colors, fonts, sizes, spacing, and other visual properties of elements, thereby creating a consistent and visually appealing design for the website.
3. JavaScript is a programming language that adds interactivity and dynamic behavior to websites. In front-end development, JavaScript is primarily used to manipulate the HTML and CSS of a webpage, handle user interactions, and create dynamic content. It enables features like form validation, animations, interactive menus, and real-time updates without requiring a page refresh.

Answers for Self-check 1-2:

1. Some common back-end programming languages are:
 - Python, Java, Ruby, PHP, C#, Node.js (JavaScript runtime for server-side development), etc.
2. Databases play a crucial role in back-end development. They are used to store and manage data for web applications. The role of databases in back-end development includes:
 - Storing and retrieving data: Databases provide a structured way to store and retrieve data efficiently.
 - Data modeling: Databases allow defining data schemas and relationships between entities.

- Querying and manipulation: Databases support querying and manipulating data using languages like SQL (Structured Query Language).
 - Data integrity and security: Databases ensure data integrity through constraints and provide security features like authentication and access control.
3. Some popular back-end frameworks or libraries and their use cases are:
- Django (Python): Django is a high-level web framework that follows the model-view-controller (MVC) architectural pattern. It is well-suited for building complex, data-driven web applications.
 - Spring Boot (Java): Spring Boot is a Java framework that simplifies the development of Java-based web applications. It provides a robust and scalable infrastructure for building enterprise-level applications.
 - Ruby on Rails (Ruby): Ruby on Rails, often referred to as Rails, is a web application framework written in Ruby. It emphasizes convention over configuration and enables rapid development of database-backed web applications.
 - Laravel (PHP): Laravel is a PHP framework known for its elegant syntax and developer-friendly features. It provides a solid foundation for building secure and scalable web applications.
 - Express.js (Node.js): Express.js is a minimalistic and flexible web application framework for Node.js. It is widely used for building fast and lightweight server-side applications and APIs.

Answers for Self-check 1-3:

1. Full-stack development refers to the practice of working on both the front-end and back-end aspects of a web application. A full-stack developer is capable of handling both the client-side (front-end) and server-side (back-end) development tasks. In contrast, front-end development focuses on the user interface and user experience, while back-end development deals with server-side logic and database management.
2. Advantages of being a full-stack developer include:
 - Versatility: Full-stack developers have a broader skill set and can work on various aspects of a project, which makes them valuable assets in a development team.

- Independence: Full-stack developers can independently work on both front-end and back-end tasks, reducing dependencies on other team members.
- Holistic understanding: By working on both ends of the application, full-stack developers have a comprehensive understanding of the entire development process, enabling them to make informed decisions and contribute effectively to the project.

3. Skills and knowledge required for full-stack development:

- Front-end technologies: Proficiency in HTML, CSS, and JavaScript is essential for full-stack developers to build responsive and interactive user interfaces.
- Back-end programming languages: Full-stack developers should be familiar with at least one back-end programming language such as Python, Java, Ruby, or Node.js to handle server-side logic.
- Databases: Knowledge of database systems like MySQL, PostgreSQL, or MongoDB is necessary to handle data storage and retrieval.
- APIs and web services: Understanding how to design, build, and consume APIs is crucial for integrating front-end and back-end components.
- Version control: Proficiency in version control systems like Git allows full-stack developers to manage code repositories efficiently.
- Problem-solving and debugging: Full-stack developers should have strong problem-solving skills and be able to troubleshoot issues across the entire application stack.
- Project management: Full-stack developers may be involved in project management tasks, so knowledge of project management methodologies and collaboration tools can be beneficial.

Answers for Self-check 2-1:

1. Connect Hardware and Network Peripherals:

- The key components of a computer system include the central processing unit (CPU), memory (RAM), storage devices (hard drives or solid-state drives), input devices (keyboard, mouse), output devices (monitor, printer), and communication devices (network interface cards).

- Generations of computers refer to different eras or stages of technological advancements in computer systems. They are categorized as follows:
 - **First Generation:** The earliest computers built using vacuum tubes and large-scale electronic components.
 - **Second Generation:** Computers that utilized transistors, which were smaller, faster, and more reliable than vacuum tubes.
 - **Third Generation:** Computers that adopted integrated circuits, allowing for increased computing power and reduced size.
 - **Fourth Generation:** Computers that introduced microprocessors, leading to significant advancements in speed and performance.
 - **Fifth Generation:** Computers characterized by the development of artificial intelligence and advanced parallel processing.
- Characteristics that define a computer system include its ability to input, process, store, and output data. Computers are programmable, digital, and capable of performing complex calculations and tasks.

Answers for Self-check 2-2:

2. Safe Work Conditions and Procedures:

- General safety guidelines when working with computers include:
 - Ensuring proper ergonomic setup for comfortable posture and reduced strain.
 - Taking regular breaks to prevent eye strain and musculoskeletal issues.
 - Keeping the workspace clean and organized to avoid hazards and tripping.
 - Using proper cable management to prevent tripping and electrical hazards.
- Fire safety is crucial in a computer environment because computers generate heat and use electrical power. Fire hazards can arise from faulty wiring, power surges, or overheating components. Fire safety measures include:
 - Installing and regularly testing smoke detectors and fire alarms.
 - Keeping fire extinguishers accessible and knowing how to use them.

- Implementing proper ventilation and cooling systems to prevent overheating.
- Using surge protectors and uninterrupted power supply (UPS) devices to protect against electrical fires.

Answers for Self-check 2-3:**3. Equipment Protection:**

- ESD (Electrostatic Discharge) protection is a set of measures taken to prevent damage to computer equipment from static electricity. It is important because static electricity can cause immediate or latent damage to sensitive electronic components.
- Different types of power fluctuations that can affect computer systems include voltage sags (temporary drop in voltage), surges (temporary increase in voltage), spikes (sudden brief increases in voltage), interruptions (complete loss of power), and transients (rapid changes in voltage).

Answers for Self-check 2-4:**4. Environment Protection:**

- Proper disposal of computer equipment is important for the environment to prevent electronic waste from polluting landfills and releasing hazardous materials into the ecosystem. Computers contain toxic substances like lead, mercury, and flame-retardant chemicals that can contaminate soil and water if not disposed of correctly.
- Recommended disposal procedures for computer equipment include:
 - Recycling through authorized e-waste recycling programs or facilities.
 - Securely erasing data before disposal to protect sensitive information.
 - Donating or reselling functional equipment to extend its lifespan.
 - Disassembling components and separating materials for recycling.
 - Following local regulations and guidelines for handling hazardous materials during disposal.

References

1. Lambert, J. (2020). Microsoft windows Step by Step installation. Microsoft Press. Retrieved from
<https://answers.microsoft.com/en-us/windows/forum/all/steps-to-installwindows-10/a499989c-f31f-49d0-8dbb-3b6425a5f47b> and
<https://www.wikihow.com/Install-Windows-10>
2. Dumas, J. P., Fortier, P. R., & Myers, B. A. (2016). Fundamentals of Computing. Cengage Learning. Retrieved from <https://www.geeksforgeeks.org/computer-fundamentals-tutorial/?ref=lbp>
3. Norton, P. (2017). Introduction to Computers. McGraw-Hill Education. Retrieved from <https://www.britannica.com/technology/computer>
4. Sinha, P. K., & Sinha, P. (2018). Computer Fundamentals. BPB Publications. World Health Organization. (2002). Retrieved from
<https://www.slideshare.net/catherinelvillanueva1/ict-83930037>

Module II

Access and Analyze Online Information

Contents

| No | Contents | page |
|----------|--|------|
| | Module Description | 77 |
| 1 | UNIT 1: THE INTERNET AND THE WORLD WIDE WEB | 78 |
| | 1.1 Introduction to The Internet | 78 |
| | 1.2 Introduction to the World Wide Web (WWW) | 89 |
| | 1.3 Introduction to Websites | 103 |
| | Unit Summary | 107 |
| | Unit Review Questions | 107 |
| 2 | UNIT 2: WEB BROWSERS | 110 |
| | 2.1 Introduction to Web Browsers | 110 |
| | 2.2 Navigating and utilizing Web browser features | 113 |
| | 2.3 Customizing and Configuring Web browser | 132 |
| | Unit Summary | 147 |
| | Unit Review Questions | 148 |
| 3 | UNIT 3: SEARCH AND EVALUATE INFORMATION OVER THE INTERNET | 152 |
| | 3.1 Search Information | 152 |
| | 3.2 Critical Evaluation | 169 |
| | 3.3 Copyright and Data Protection | 173 |
| | Unit Summary | 176 |
| | Unit Review Questions | 176 |
| 4 | UNIT 4: ONLINE COMMUNICATION AND COLLABORATION | 179 |
| | 4.1 Online Communication | 179 |
| | 4.2 Online Collaboration | 196 |
| | Unit Summary | 226 |
| | Unit Review Questions | 226 |
| 5 | UNIT 5: SAFE ONLINE ENVIRONMENT | 228 |
| | 5.1 Overview of Security and Privacy | 229 |
| | 5.2 Threats To Computers, Devices and Data | 231 |
| | 5.3 Protecting computers, devices and data | 135 |
| | Unit Summary | 253 |
| | Unit Review Questions | 254 |
| | References | 259 |

Module Description

This student module was prepared to help you to achieve the required competency in “Web Design and development” career pathway. It describes the essential skills and knowledge required to effectively navigate, search, and evaluate information on the internet.

This module covers the following unit titles

- The internet and the world wide web
- Web browsers
- Search and evaluate information over the internet
- Online communication and collaboration
- Safe online environment

This module will also assist you to attain the following learning outcomes.

- Describe the Internet and the world wide web
- Use and configure Web Browsers
- Search effectively and critically to evaluate information over the web .
- Enhance online communication and collaboration
- Create a safe online environment

Module Instruction:

Learning Instructions: How to use this Module

For effective use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what are being discussed.
3. Accomplish the “Self-checks” which are placed following each topics. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

UNIT 1

THE INTERNET AND THE WORLD WIDE WEB

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 1.1. Introduction to the Internet
- 1.2. Introduction to the World Wide Web
- 1.3. Introduction to Websites

Unit Learning Outcomes

This unit will also assist you in attaining the following learning outcome. Specifically, upon completion of this unit, you will be able to:

- ✓ Describe the Internet and the World Wide Web

Unit Overview

Welcome to Unit One - "The Internet and the World Wide Web." This unit is like your entry pass into the awesome world of digital stuff and exploring the web. We're going on an adventure, starting by figuring out what the Internet is all about, checking out its cool history, and understanding how it does its magic. Then, we'll dive into the World Wide Web, the big deal that shapes the digital world. We'll learn about its past, how it works, and the important pieces that make it tick. By the end of this unit, you'll not only get what the Internet and WWW are but also see how they affect the way we do things, learn, and make cool stuff online.

Key Terms: *Internet, WWW, URL, IP Address, DNS, Website, Webpage, HTTP, HTTPS*

1.1. Introduction to the Internet

The Internet and the World Wide Web (WWW) are two closely intertwined technologies that have revolutionized the way we communicate, access information, and conduct business in the modern world. While often used interchangeably, they serve distinct yet complementary roles in our digital lives. In the next sections, we will look at each of the terms in detail which will help us to differentiate them.

1.1.1. Defining the internet

Dear Students! What is the Internet? Can you try to define the term based on the introduction given above? Hoping that you have defined it, let's see together some of the most common definitions of the internet.

The Internet is the global network of interconnected computers and devices, allowing them to communicate and exchange data with each other. It's a vast, decentralized network that spans the globe, connecting billions of devices, including personal computers, servers, smartphones, and more. This network operates on a set of standardized protocols and technologies. The Internet's primary function is to facilitate data transfer between devices, making it possible for us to send emails, share files, browse websites, and stream videos. It doesn't host content itself; rather, it serves as the infrastructure that enables the transfer of information.



Figure 1.1: The internet

1.1.2. History of the Internet

Dear Students! How do you think the Internet evolved from a small network of computers to the vast global network we use today? In our next section, we will uncover the fascinating history of the Internet, tracing its roots and major milestones that shaped it into the indispensable tool it is now.

The internet was born out of a need to share information and connect people around the world. On October 29, 1969, an organization called ARPANET (Advanced Research Projects

Agency) launched the first iteration of the internet (also known as ARPANET) connecting four major computers at The University of Utah, UCSB, UCLA, and Stanford Research Institute .When this network of computers was connected, universities were able to access files and transmit information from one organization to the other, as well as internally. As researchers developed the system, they continued to connect computers from other universities, including MIT, Harvard, and Carnegie Mellon. Eventually, ARPANET was renamed “internet.”

Who used the internet in this stage?

In its earliest days, the internet was only used by computer experts, scientists, engineers, and librarians who had to learn a complicated system in order to use it, but as the technology improved and consumers adapted, it became an essential tool for people around the globe.

How and when did the functionality of the internet change?

The 1970s was a serious time of transition for the internet. Email was introduced in 1972, libraries across the country were linked, and above all, information exchange became more seamless thanks to Transport Control Protocol and Internet Protocol (TCP/IP) architecture.

The invention of these protocols helped to standardize how information was sent and received over the web, making the delivery more consistent, regardless of where or how you’re accessing the internet.

When did the internet become user-friendly?

Then in 1986, the National Science Foundation took the development of the internet to the next echelon by funding NSFNET, a network of supercomputers throughout the country.

These supercomputers laid the groundwork for personal computing, bridging the gap between computers being used exclusively for academic purposes and computers used to perform daily tasks.

In 1991, The University of Minnesota developed the first user-friendly internet interface, making it easier to access campus files and information. The University of Nevada at Reno continued to develop this usable interface, introducing searchable functions and indexing.

When did consumers begin using the internet?

As the internet's development continued to evolve and shift focus, the National Science Foundation discontinued its sponsorship of the internet's backbone (NSFNET) in May of 1995. This change lifted all commercial use limitations on the internet and ultimately, allowed the internet to diversify and grow rapidly. Shortly after, AOL, CompuServe, and Prodigy joined Delphi to offer commercial internet service to consumers.

The debut of WiFi and Windows 98 in the late nineties marked the tech industry's commitment to developing the commercial element of the internet. This next step gave companies like Microsoft access to a new audience, consumers (like yourself).

WHAT DOES INTERNET USAGE LOOK LIKE TODAY?

Flash-forward to today. It's estimated that three billion people now use the internet, many of whom use it on a daily basis to help them get from Point A to Point B, catch up with loved ones, collaborate at work, or to learn more about important questions like how does the internet work?

As technology changes and the internet weaves its way into just about every aspect of our lives, even more people are expected to use it. In 2030, researchers project there will be 7.5 billion internet users and 500 billion devices connected to the internet.

1.1.3. How the Internet Works

Dear students! a pose question: how would you think devices over the internet communicate with each other? let us look at the basics of the internet's architecture and how they work.

The Internet is based on a layered architecture consisting of multiple protocols and technologies. Understanding the functioning of the Internet is crucial for anyone delving into web design and development. This section will explain the basics of the Internet's architecture.

The Internet, a globally interconnected network of devices, facilitates the exchange of information on an unprecedented scale. To comprehend how it works, we must explore its fundamental components and the intricate processes that underpin this vast digital ecosystem. The modern Internet is a vast network of networks, consisting of millions of private, public, academic,

business, and government networks of local to global scope. Here's a breakdown of how the modern Internet works, focusing on its structure:

1. Devices and Endpoints:

At the heart of the Internet are the myriad devices that form its nodes—computers, smartphones, servers, routers, and more. Each device is assigned a unique identifier known as an IP (Internet Protocol) address. This addressing system allows seamless communication between these devices across the network. Client computers connect to the internet through an ISP.

Example: Your phone connected to a mobile network or your laptop connected to WiFi.

2. Physical Network Infrastructure

The foundation of the Internet is its vast array of physical components. This includes:

- **Cables:** These include copper telephone wires, TV cables, and fiber optic cables. Undersea cables also play a crucial role in intercontinental data transmission.
- **Routers and Switches:** These devices manage traffic within and between networks by forwarding data packets to their intended IP addresses.
- **Servers:** Central to data exchange, servers store, send, and receive data. Web servers, for instance, host websites and send their data to other devices.
- **Data Centers:** These facilities house large numbers of servers and other computing hardware. They are essential for cloud computing services and extensive data storage and processing.

3. Internet Service Providers (ISPs):

Internet Service Providers (ISPs) are essential organizations that facilitate access to the internet for individuals, businesses, and organizations. They serve as gateways to the broader internet infrastructure, providing various types of internet connections including broadband (cable, DSL, fiber optics), wireless, satellite, and even traditional dial-up services. ISPs manage the transmission of data across the internet, ensuring that everything from emails to streaming content is moved efficiently between users and the wider web. Additionally, many ISPs offer services like domain registration and web hosting, allowing users to purchase domain names and host websites on ISP servers.

ISPs also play a critical role in network security by implementing measures to protect their networks and customers from cyber threats such as malware, hacking, and unauthorized access. This includes monitoring network traffic for suspicious activities and providing firewall options. Furthermore, they are responsible for maintaining and upgrading the physical network infrastructure necessary for internet connectivity, including cables, routers, and switches.

Some prominent examples of ISPs include Comcast (Xfinity) and AT&T in the USA, which offer a wide range of services including internet, TV, and telephone services; BT Group in the UK, known for its fiber-optic services and digital TV packages; Deutsche Telekom in Germany, providing extensive telecommunications services across Europe; Airtel in India, which offers mobile and broadband services; offering a broad range of telecommunications services both domestically and internationally. Also in our country Ethio telecom and Safaricom are Internet service providers that are responsible to provide the service throughout the county. Through their diverse offerings and pivotal role in data management, ISPs are foundational to the operation and accessibility of the global internet.

4. Protocols and Standards

Internet protocols are the set of rules and standards that define how data is transmitted and communicated across networks. These protocols ensure that devices on the internet can communicate with each other effectively, despite differences in hardware, operating systems, and internal configurations. Let's take a brief look at some of the key protocols that keep the Internet

- **TCP/IP (Transmission Control Protocol/Internet Protocol):** The backbone of Internet communication is the TCP/IP protocol suite. This suite manages the transmission of data by breaking it into packets, assigning addresses, and ensuring the reliable delivery of information between devices.
 - **TCP** is one of the main protocols of the Internet Protocol Suite. It ensures the reliable transmission of data between a source and destination. TCP breaks down data into smaller packets, manages the transfer of these packets across the network, and reassembles the packets at the destination. It provides error-checking and guarantees that data will arrive in order and without errors. For example, when you download a file, TCP ensures that all parts of the file reach your device correctly.

- **IP** is responsible for addressing and routing packets so that they can travel across networks and arrive at the correct destination. Devices on the Internet have unique IP addresses that identify them and enable communication. IP operates at a lower level than TCP and does not guarantee the delivery of packets; it relies on TCP for error handling and reassembly.
- **DNS (Domain Name System):** Translates human-readable domain names into machine-readable IP addresses. The DNS acts as a virtual phone book, translating human-readable domain names into IP addresses. This essential service enables users to access websites using easily memorable names, simplifying the user experience and abstracting the complexity of numerical IP addresses.
- **HTTP/HTTPS (HyperText Transfer Protocol/Secure):** Protocols for transmitting web data. HTTP is the protocol used by the Web to transfer data over the Internet. It allows for the fetching of resources, such as HTML documents. It's a request/response protocol between clients and servers. A browser, for instance, may request a webpage using HTTP, and the server will respond with the content of that page. HTTPS adds a layer of encryption to secure the data.
- **Simple Mail Transfer Protocol (SMTP):** SMTP is the standard protocol for email transmission across networks. It is used when email is sent from a client to a server or between servers. SMTP only covers the sending of emails; other protocols like IMAP and POP3 are used for retrieving and storing messages.
- **File Transfer Protocol (FTP):** FTP is used for the transfer of files from one host to another over the Internet. It supports two modes of data transfer: ASCII and binary. FTP is useful for moving large files and is commonly used by websites to upload or download multimedia files.

These protocols are fundamental to how the Internet operates, allowing for a wide range of online activities and services. Understanding these protocols helps students grasp the technical foundations of Internet communication and how data flows across the network.

5. Clients:

Clients (End-users) interact with the Internet through web browsers, such as Chrome or Firefox. These applications send requests to servers, interpret responses, and render web pages, handling diverse content types like HTML, CSS, and JavaScript to create the interactive web experience.

6. Web Servers:

Web servers store and deliver web pages in response to client requests. Equipped with software like Apache or Nginx, they process incoming requests, retrieve the requested files, and send them to the user's browser, forming the backbone of the client-server architecture.

7. Data Transmission:

Data travels across the Internet in the form of packets. These packets navigate a network of routers and switches, dynamically choosing the most efficient path based on routing algorithms to ensure optimal and timely delivery.

8. Firewalls and Security Measures:

Firewalls act as gatekeepers, protecting networks from unauthorized access and potential cyber threats. Additional security measures, such as encryption through SSL/TLS and Virtual Private Networks (VPNs), contribute to securing data during transmission.

10. IP addresses

Every computer on the internet is assigned a series of numbers, like 172.217.9.206. That series of numbers uniquely identifies the computer over the internet. If a URL like (www.google.com) is a nickname for a website, an IP Address (like 8.8.8.8) is its real name.

IPv4 (Internet Protocol Version 4) and IPv6 (Internet Protocol Version 6) are versions of the Internet Protocol, which is used to identify devices on a network and route traffic across the internet.

IPv4

IPv4 uses a 32-bit address scheme, allowing for a theoretical maximum of approximately 4.3 billion unique addresses. Due to the exponential growth of the internet and the number of devices connected to it, IPv4 addresses are now nearly depleted. An IPv4 address is typically displayed in decimal digits, separated by dots.

Example of an IPv4 Address: 92.168.1.1

IPv6

While IPv4 has limitations due to a finite address space, IPv6 was introduced to accommodate the ever-growing number of connected devices and ensure the sustainability of the Internet. To address the depletion of IPv4 addresses and It uses a 128-bit address scheme. This allows for a vastly larger number of devices to be connected directly to the internet. IPv6 addresses are written in hexadecimal and separated by colons.

Example of an IPv6 Address:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

IPv6 not only provides more address space but also includes enhancements for security and routing efficiency compared to IPv4. Transitioning to IPv6 has been gradual, with both protocols coexisting in the current internet infrastructure to ensure compatibility and seamless connectivity.

In Conclusion , the Internet operates as a seamlessly interconnected network, orchestrating the flow of data through standardized protocols, addressing schemes, and secure communication mechanisms. This intricate web of technologies and processes sustains the global exchange of information, shaping the digital landscape we navigate daily.

1.1.4. Applications of the Internet

Dear Students! Can you identify some ways that the Internet's functionality impacts how we communicate, learn, and entertain ourselves? Moving forward, we will explore the numerous applications of the Internet, seeing how it not only supports various forms of communication but also revolutionizes fields such as education, business, and entertainment.

The Internet's significance extends far beyond its technical infrastructure; it has become an integral part of contemporary society. The Internet has democratized information access, empowering individuals with a wealth of knowledge. Socially, it has facilitated global connections and transformed the way people interact. Economically, it has given rise to e-commerce and new business models.

The Internet's influence on society is profound, shaping how we communicate, work, and navigate the modern world. For example, Using the internet:

- ✓ A college student uses a smart phone to record a stunning touchdown, and then uploads the video to allow others to view it.
- ✓ While walking down the street in a city, a teenager runs an app to find others nearby who are interested in playing an online game.
- ✓ A person suffering from a chronic disease wears a battery-powered monitor that sends an update to their doctor every fifteen minutes.
- ✓ A family on vacation uses a smart phone to contact their home security system and see views of the interior of their home.
- ✓ Software developers who live in different areas of the world can collaborate on a project by working on the codes simultaneously.

Here's a brief overview of some key applications of the Internet that have had significant impacts on modern society:

1. Communication: One of the primary uses of the Internet is communication. Email services like Gmail and Outlook allow people to send messages instantly across the globe. Social media platforms such as Facebook, Twitter, and Instagram have transformed how we socialize, enabling us to share and connect with others in real-time. Video conferencing tools like Zoom and Skype support both personal and professional interactions, facilitating virtual meetings, classes, and family gatherings.

2. Information and Research: The Internet is a vast repository of information, accessible via search engines like Google, Bing, and Yahoo. Whether it's historical facts, scientific data, or the latest news, the Internet provides an unparalleled wealth of resources for research and learning. Online encyclopedias, scholarly articles, and tutorials offer invaluable support for educational and professional development.

3. Entertainment: The Internet provides a wide range of entertainment options. Streaming services like Netflix, Hulu, and YouTube offer movies, TV shows, and user-generated content available at the click of a button. Online gaming platforms connect players from around the world, and music streaming services like Spotify and Apple Music give users access to millions of songs and podcasts.

4. **E-Commerce:** Online shopping has become a staple, with websites like Amazon, eBay, and Alibaba offering the ability to purchase almost anything from the comfort of home. This extends to food delivery from restaurants, grocery shopping, and the global marketplace of goods and services, all facilitated by the Internet.

5. **Education:** The Internet has dramatically expanded access to education. Online courses, virtual classrooms, and e-learning platforms like Coursera, Udemy, and Khan Academy make it possible for anyone with an internet connection to learn new skills or earn degrees. Educational resources are more accessible than ever, catering to different learning styles and needs.

6. **Healthcare:** Telemedicine is an emerging field where the Internet plays a crucial role. Patients can consult with doctors online, receive remote diagnoses, and manage prescriptions without needing to visit a clinic in person. Online health resources, patient portals, and mobile apps for health management further exemplify the Internet's impact on healthcare.

7. **Banking and Finance:** Internet banking allows consumers to manage their finances online, from transferring funds to trading stocks. Financial services have become more accessible, and the Internet also supports the emergence of digital currencies and mobile payment systems like PayPal and Venmo.

8. **Cloud Computing:** The delivery of computing resources, including storage, servers, and software, over the Internet, providing on-demand access and scalability.



Self-check 1-1:

Dear Students! We have discussed what a web browser is and the different types of browsers. Now it is your turn to answer the following self-check questions

1. What is the internet?
 2. What are the different applications of the internet?
 3. What is (ISPs)? In our country who acts as ISP?
-

Have you answered the self-check questions? If yes, let us move on to discuss about WWW.

1.2. Introduction to the World Wide Web (WWW)

The World Wide Web, commonly referred to as the Web, is one of the most influential innovations that operates on the Internet. It has revolutionized the way we access and share information, connect with others, and engage in various online activities. Here's a breakdown of this topic to provide a clear understanding for you:

1.2.1. Definition the World Wide Web

Dear Students! What is the world wide web (WWW)? Can you try to define the term based on the introduction given above? Hoping that you have defined it, let's see together some of the most common definitions of the Web.

The World Wide Web is a system of interlinked hypertext documents accessed via the Internet. It provides a graphical interface for navigating and accessing information, allowing users to browse websites and interact with web-based applications. With a web browser, users can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks. The Web utilizes the Internet to transmit its data, making it a vast information-sharing model built on top of the existing infrastructure.

1.2.2. The Internet VS The World Wide Web

Dear Students! How do you think the Web differs from the Internet itself? Reflect on this question as we prepare to explore the distinctions between these two fundamental concepts in our next section.

The terms Internet and World Wide Web are used interchangeably in nonprofessional terms they are not the same. It is common practice to say "going on the Internet" whenever we open any web page on a browser. However, the World Wide Web or the Web is one of the most popular Internet services available. The Web is a collection of interconnected documents (web pages) and other web resources, linked by hyperlinks and URLs. It uses HTTP as an underlying protocol for information transfer, which is one of the several other protocols available in networking theory.

- **The Internet** is a massive network of networks, a networking infrastructure that connects millions of computers globally, allowing them to communicate.

- **The World Wide Web** is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet.

1.2.3. History of the World Wide Web

Dear Students! explored the differences between the Internet and the World Wide Web, do you wonder how the Web came to be? What historical developments allowed it to become the vast, interconnected network we rely on today? As we move forward, we'll dive into the history of the World Wide Web.

The WWW was invented by Sir Tim Berners-Lee in the late 1980s while working at CERN, the European Organization for Nuclear Research. His goal was to create a system that would allow researchers to easily share and access information across different computer platforms.

The World Wide Web has undergone a significant evolution since its inception. Initially, it consisted mainly of static web pages with basic text and images. Over time, advancements in web technologies led to dynamic content, interactive features, and multimedia integration. The emergence of Web 2.0 introduced user-generated content, social media, and collaborative platforms. Further developments, including the rise of mobile internet and the semantic web, continue to shape the WWW's evolution, making it a dynamic and ever-expanding platform for information dissemination and online interaction.

1.2.4. How the World Wide Web Works

Dear Students! How do you think a web page appears on your screen after you type a URL? In our next section, we'll uncover the workings of the Web, looking closely at how web pages are requested, transmitted, and displayed in your browser.

Now, we have understood that WWW is a collection of websites connected to the internet so that people can search and share information. Now, let us understand how it works! The Web works as per the internet's basic client-server format as shown in the following image. The servers store and transfer web pages or information to user's computers on the network when requested by the users. A web server is a software program which serves the web pages requested by web users using a browser. The computer of a user who requests documents from a server is known as a

client. Browser, which is installed on the user's computer, allows users to view the retrieved documents.

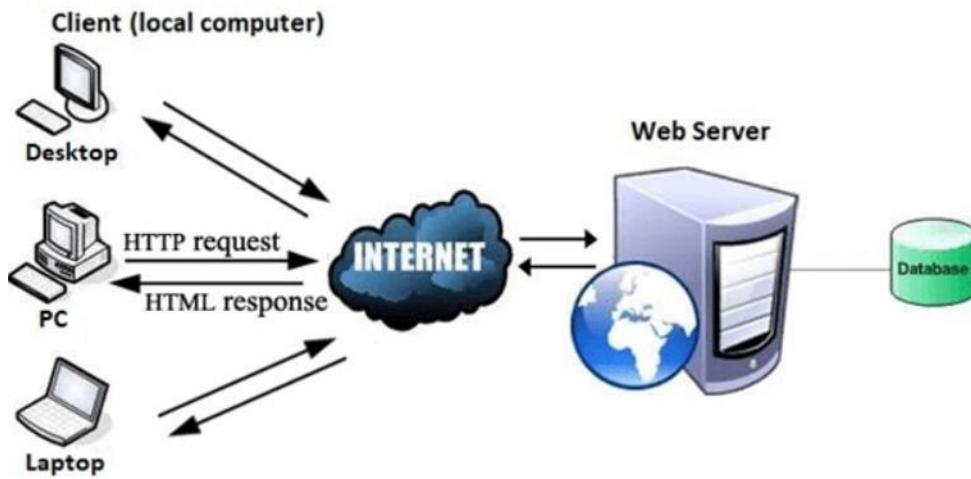


Figure 1.2: How the World Wide Web Works

All the websites are stored in web servers. Just as someone lives on rent in a house, a website occupies a space in a server and remains stored in it. The server hosts the website whenever a user requests its Webpages, and the website owner has to pay the hosting price for the same. The moment you open the browser and type a URL in the address bar or search something on Google, the WWW starts working. There are three main technologies involved in transferring information (web pages) from servers to clients (computers of users). These technologies include Hypertext Markup Language (HTML), Hypertext Transfer Protocol (HTTP) and Web browsers.

The World Wide Web is a complex system that enables the easy retrieval and sharing of information over the internet. To understand how the Web works, it is crucial to look at its structure, components, and the step-by-step process of accessing web content. Here's a breakdown to illustrate this:

1.2.4.1 Structure and Components

- **Web Pages:** Web pages are documents, typically written in HTML (HyperText Markup Language), that can contain text, images, videos, and other multimedia. They are the building blocks of websites.

- **Web Browsers:** Web browsers, such as Google Chrome, Mozilla Firefox, and Safari, are software applications used to access web pages. They interpret HTML files and display them as web pages.
- **Web Servers:** Web servers are computers set up to respond to requests for web pages. They store, process, and deliver web pages to users.
- **Hyperlinks:** Hyperlinks, or links, connect web pages to one another. They allow users to navigate between related pages and websites easily.
- **URLs (Uniform Resource Locators):** URLs are web addresses that point to a specific web page. Each URL is unique and directs the browser to a particular resource on the web. we will look this concept in detail below in section (**1.2.4.5**)
- **HTTP/HTTPS (HyperText Transfer Protocol/Secure):** These protocols govern how data is transmitted between web browsers and servers. HTTPS adds a layer of encryption for security. we will look this concept in detail below in section (**1.2.4.3**)

1.2.4.2 Step-by-Step Process of Accessing the Web

1. Entering a URL:

The process begins when a user enters a URL into a web browser. For example, entering <https://www.example.com>.

2. DNS Lookup:

The browser performs a DNS (Domain Name System) lookup to translate the URL into an IP address that points to a specific web server. For instance, www.example.com might translate to 192.0.2.1.

3. Browser Sends HTTP Request:

The browser sends an HTTP request to the web server at the resolved IP address, asking for the web page associated with the URL.

4. Server Processes Request:

The web server receives the request, processes it, and sends back the requested web page, often along with associated resources like images and CSS files.

5. Browser Displays Web Page:

The browser receives the data and renders the web page for the user to view. This includes formatting based on HTML and CSS, and executing any JavaScript to make the page interactive.

6. Navigation Using Hyperlinks:

While viewing the web page, the user can click on hyperlinks to access other related pages. Each click results in new HTTP requests to the server, repeating the process.

Understanding these components and steps provides insight into the fundamental workings of the World Wide Web, explaining how it allows for easy, efficient access to a vast amount of information across the globe.

1.2.4.3 HTTP and HTTPS

HTTP (Hypertext Transfer Protocol):

HTTP is an client-server protocol that allows clients to request web pages from web servers. It is an application level protocol widely used on the Internet. Clients are usually web browsers. When a user wants to access a web page, a browser sends an HTTP Request message to the web server. The server responds with the requested web page. By default, web servers use the TCP port 80.

Clients and web servers use request-response method to communicate with each other, with clients sending the HTTP Requests and servers responding with the HTTP Responses. Clients usually send their requests using GET or POST methods, for example GET /homepage.html. Web servers responds with a status message (200 if the request was successful) and sends the requested resource. An example will clarify this process:



Figure 1.3: How the World Wide Web Works

The client wants to access <http://Example.com> and points his browser to the URL <http://example.com> (this is an example of an HTTP Request message). The web server hosting <http://example.com> receives the request and responds with the content of the web page (the HTTP response message).

However, HTTP operates over an unencrypted channel, which means the data exchanged between the user's browser and the server is susceptible to interception and manipulation by malicious entities. This lack of encryption poses a security risk, especially when transmitting sensitive information such as login credentials or personal details. While HTTP is suitable for static websites and information that doesn't involve sensitive data, its vulnerability to interception makes it less ideal for secure transactions or interactions that demand data integrity and confidentiality. To address these security concerns, a more secure variant, HTTPS, was developed. In Browsers, you can immediately recognize that a web site is using HTTP. it appears like the following image.



HTTPS (Hypertext Transfer Protocol Secure):

HTTPS, or Hypertext Transfer Protocol Secure, is an extension of HTTP designed to provide a secure and encrypted connection between the user's browser and the web server. The security is implemented through the use of SSL (Secure Sockets Layer) or its successor, TLS (Transport Layer Security), protocols. The encryption ensures that the data exchanged between the user and the server remains confidential, preventing eavesdropping and tampering.

One of the key features of HTTPS is the use of SSL/TLS certificates. Websites that implement HTTPS have these certificates, which are verified by trusted third-party entities called Certificate Authorities. When a user accesses an HTTPS-enabled website, their browser checks the authenticity of the SSL/TLS certificate. If valid, the browser establishes a secure connection, and the web address usually displays "https://" along with a padlock icon, indicating a secure connection.

HTTPS is essential for any website that handles sensitive information, such as online transactions, login credentials, or personal data. It provides users with confidence in the security of their interactions with the website, fostering trust and protecting against various cyber threats, including man-in-the-middle attacks.

In Internet Browsers, you can immediately recognize that a web site is using HTTPS because a lock appears to the right of the address bar:



HTTP Vs HTTPS

In summary, while HTTP is suitable for basic web browsing and information retrieval, HTTPS adds a layer of security crucial for safeguarding sensitive data on the internet. The adoption of HTTPS has become increasingly important, not only for e-commerce and banking websites but also for any platform that values user privacy and data security. As the internet evolves, HTTPS continues to be the recommended standard for secure communication, ensuring the integrity and confidentiality of data exchanged between users and websites.

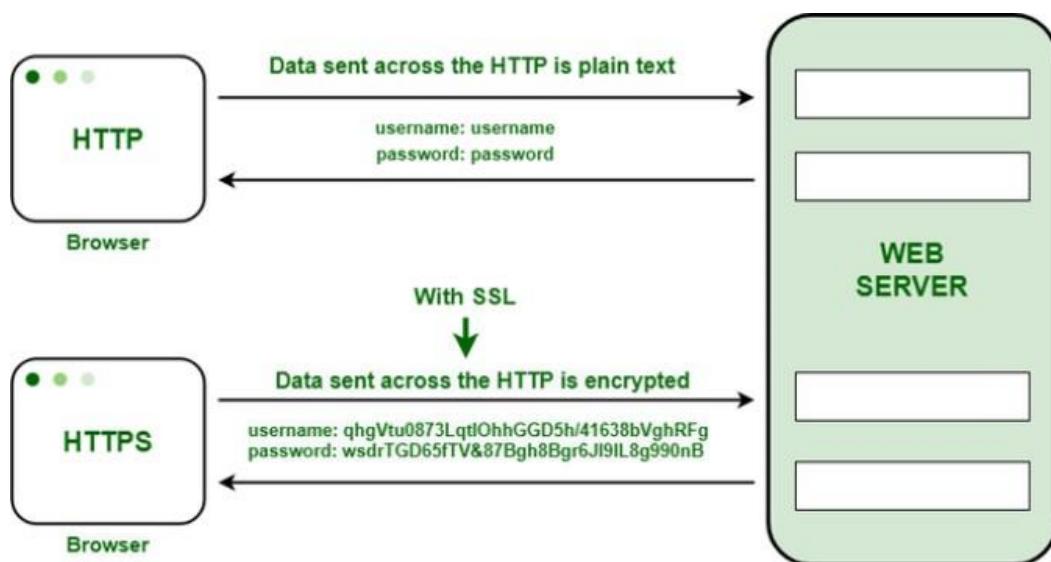


Figure 1.4: HTTP vs HTTPS

1.2.4.4 The Domain Name System (DNS)

Understanding the structure of the DNS is essential for web designers and developers. The Domain Name System (DNS) is a hierarchical naming system that translates domain names into IP addresses. The Domain Name System (DNS) operates in a hierarchical structure, which plays a vital role in translating human-readable domain names into IP addresses that computers can understand. This hierarchical nature allows for efficient and decentralized management of domain names across the Internet.

The Domain Name System (DNS) can be thought of as the directory of the Internet. We find an online page or website by typing in the URL – like ethiotelecom.com or some-site.com. Our web browsers, on the other hand, need to translate the URL to Internet Protocol (IP) addresses to find the correct site. It is a DNS that translates domain names to IP addresses so our browsers can resolve, or connect to, requested Internet resources.

Every single device on the Internet has a unique IP address by which it can be uniquely identified by the other online devices. A DNS server eliminates the need for us to memorize these IP addresses every time we want to visit a site or connect to a device. It is much easier for us to type in a URL than IPv4 IP addresses (E.g. 192.168.1.1) or, worse, the more complex IPv6 addresses (E.g. 2400:cb00:2048:1::c629:d7a2).

The DNS architecture consists of a hierarchical and decentralized name resolution system for computers, services or any other resources connected to the Internet or a private network. It stores the various associated information of the domain names assigned to each of the resources. Queries for a resource pass through the DNS – with the URLs as parameters. The DNS then takes the URLs, translates them into the target IP addresses, and sends the queries towards the correct resource.

How DNS (Domain Name System) Works:

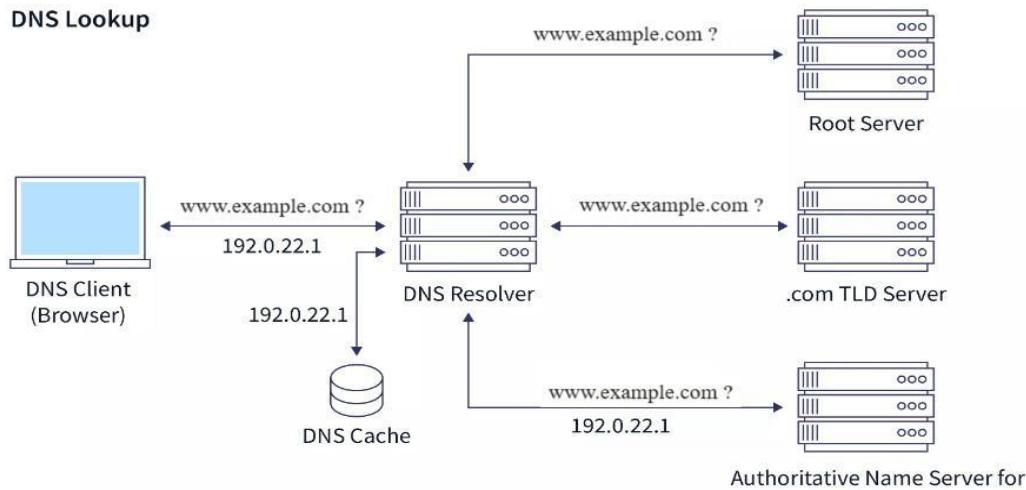


Figure 1.5: How the DNS Works

- 1.2.4.4.1 If you type `www.example.com` into a web browser, the query is transmitted over the Internet and received by a DNS resolver.
- 1.2.4.4.2 The DNS resolver then queries a DNS root nameserver.
- 1.2.4.4.3 After then, the root server responds to the DNS resolver with the address of a TLD DNS server (such as `.com` or `.net`), which keeps the information for the resolver's domains. Our request for `scaler.com` is directed to the `.com` top-level domain (TLD).
- 1.2.4.4.4 The DNS resolver then requests the `.com` TLD after receiving the address of the TLD by the root server.
- 1.2.4.4.5 The IP address of the domain nameserver, `example.com`, is then returned by the TLD server.
- 1.2.4.4.6 Finally, the DNS resolver sends a query to the domain's nameserver.
- 1.2.4.4.7 The nameserver returns the IP address, for `example.com`, to the resolver.
- 1.2.4.4.8 The DNS resolver then returns the IP address of the domain that was requested originally to the web browser.

1.2.4.4.9 An HTTP request is sent to the IP address by the browser.

1.2.4.4.10 The server returns the webpage to be rendered in the browser at that IP.

1.2.4.4.11 Finally, after all the processes mentioned above, the user can now view the web page on their machine.

1.2.4.5 Uniform Resource Locator (URL)

A Uniform Resource Locator (URL) is a string of characters that provides the address of a specific resource on the web. This section will introduce the concept of URLs, explaining their structure and components. A URL (Uniform Resource Locator) is a string of characters that provides the address of a resource on the internet. It serves as a unique identifier for web pages, files, images, or any other resource accessible via the web.

URL Examples:

- HTTP: <http://www.example.com/index.html>
- HTTPS: <https://www.example.com/login>
- FTP: <ftp://ftp.example.com/files/document.pdf>
- File: <file:///C:/Documents/report.docx>

Let's explore the topic of URLs in detail and cover some related subtopics:

Definition and Structure

URL: The concept and definition of a URL as a standardized format for addressing resources on the internet. A URL consists of several components:

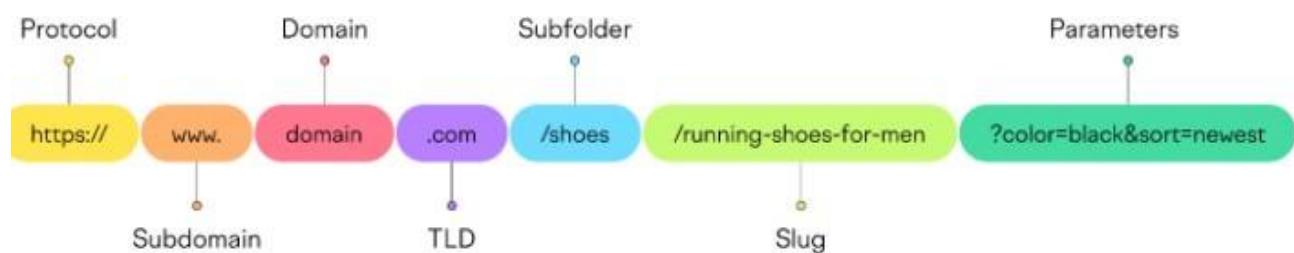


Figure Error! No text of specified style in document.-2 Parts Of a URL

1. Protocol



The protocol tells your browser how to connect to a webpage. It could be HTTP (hypertext transfer protocol) or HTTPS (HTTP secure). The main difference between the two is that HTTPS encrypts and protects any data transmitted between the server and browser.

URL Examples with different Protocols:

- HTTP: http://www.example.com/index.html
- HTTPS: https://www.example.com/login
- FTP: ftp://ftp.example.com/files/document.pdf

1. Subdomain



A subdomain is a string of letters or a complete word that appears before a URL's first dot. The most popular subdomain is www. It stands for World Wide Web, communicating that the URL is a web address. In the past, it was common to use www. But nowadays you can omit it from your URLs if you want. Then there are other subdomains—blog, store, support, news, careers, and so on—that are used for managing and organizing sections of a website that serve a specific function.

2. Domain



A domain is the main part of the URL that identifies the website. Like MOE, Amazon, Expedia, or Ethiotelcom.

3. Top-Level Domain (TLD)



The TLD (also called domain extension) is the part that comes after the name of your website, like “.com.” You’ll come across many TLDs on the internet.

Examples of TLDs include:

Generic Top-Level Domains (gTLDs):

- .com: Commercial entities.
- .org: Non-profit organizations.
- .net: Network-related entities.
- .edu: Educational institutions.
- .gov: U.S. government agencies.
- .mil: U.S. military.

Country Code Top-Level Domains (ccTLDs):

- .et: Ethiopia
- .us: United States.
- .uk: United Kingdom.
- .ca: Canada.
- .jp: Japan.
- .au: Australia.

New Generic Top-Level Domains (new gTLDs):

- .app: Application-related websites.
- .blog: Blogs and publications.
- .tech: Technology-related sites.
- .guru: Expert or advice-focused domains.

TLDs serve to categorize websites based on their purpose, origin, or content. They provide users with information about the nature of a website and help organize the vast expanse of the internet into more manageable segments. Choosing an appropriate TLD is often influenced by the type of content or organization associated with the domain.

4. Subfolder



https://www.domain.com/shoes/running-shoes?color=black

A subfolder is a folder or directory that is located within the top-most directory (or main directory) in your site hierarchy. For example, consider the URL www.domain.com/shoes/. In this URL, the “shoes” subfolder is located within the main directory of the website, which is www.domain.com.

- Similar to subdomains, subfolders are used to separate website content into logical sections.

5. Slug



https://www.domain.com/shoes/running-shoes?color=black

A slug is the part of a URL that identifies a specific page or a post on a website. It helps users understand the context and content of a page. Look at this URL slug, for example: “/best-baby-shampoos/”. Reading this slug alone, users can get an idea of what the page is about. But sometimes, you’ll also come across URL slugs that read like this: “/785321/”. The primary goal of the URL slug is to describe the content of a page. So when you create a slug, make sure it’s descriptive.

6. URL Parameters



https://www.domain.com/shoes/running-shoes?color=black

URL parameters (or query strings) are part of a URL that comes after a question mark (?). They’re composed of keys and values, separated by an equal sign (=). The key tells you what kind of information is being passed. The value is the actual information being passed.

Let’s look at an example:

In the URL below, “color” is the key and “blue” is the value. This parameter will apply a filter to a webpage to display only blue products. You can add multiple parameters to a URL by separating them with an ampersand (&).



Figure 1.6: URL structure

Now, there are two parameters: “color” with the value “blue” & “sort” with the value “newest”

This applies a filter to a webpage to show blue products and sorts them by the newest first.

Parameters serve multiple use cases:

- **Searching** parameters allow you to search results from a website’s internal search engine
- **Filtering** parameters let you sort and filter listings on category pages. Listings will often be products, jobs, hotels, flights, etc. And they can be filtered by various attributes, like price, availability, size, brand, salary, location, flight time, delivery time, ratings, etc.
- **Tracking** parameters help you track traffic from your ads and marketing campaigns
- **Paginating** parameters are helpful in organizing blog archive pages and forum threads in a series of pages



Self-check 1-2:

Dear Students! So far, internet and the world wide web is defined from different angle, the way how they work, its purposes are outlined. Now it is your turn to answer the following self-check questions

1. What is WWW?
2. What is DNS?
3. What is the difference between Internet and The Web?
4. What is a URL? list the different components of a URL

Have you answered the self-check questions? If yes, let us move on to discuss about websites.

1.3. Introduction to Websites

Dear Students! What elements make up a typical website, and why are they crucial for businesses and individuals alike? Let's delve into our next topic, Introduction to Websites, where we'll uncover what websites are, how they are structured, and their diverse functionalities in the digital age.

A website is a collection of web pages and related content that are hosted on a web server and can be accessed via the World Wide Web. It serves as a virtual space where individuals, organizations, businesses, and other entities can share information, showcase products or services, communicate with users, and engage in various online activities. Websites are designed to be viewed using web browsers, and they can range from simple single-page sites to complex multi-page portals. This presence is critical in today's digital world where accessibility, visibility, and engagement play crucial roles in communication and transaction.



Figure 1.6: Example website (moe.edu.et)

1.3.1. Purpose of Websites

Websites serve a myriad of purposes across various sectors. Here are some of the key functions they fulfill:

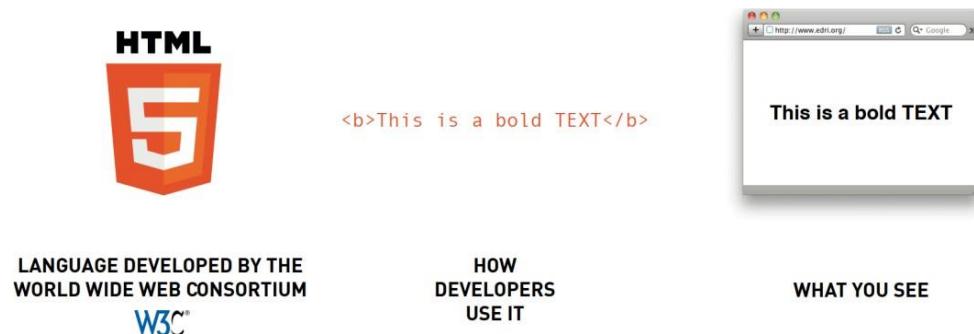
- **Information Dissemination:** Many websites are created to provide information. These can range from news platforms and educational sites to blogs and informational portals that offer articles, reports, and guides on a wide array of topics.
- **E-commerce:** Online stores and e-commerce platforms allow businesses to sell products and services directly to consumers across the globe. Sites like Amazon and eBay are prime examples, facilitating billions of transactions.

- **Marketing and Branding:** Websites are essential tools for marketing and branding efforts. They offer a controlled environment to showcase a brand's identity, communicate with potential customers, and build relationships through engaging content.
- **Customer Service:** Websites also function as a customer service portal, providing answers to frequently asked questions, live chat support, and resources for customers seeking help.
- **Community Building:** Social media platforms, forums, and discussion boards on websites enable the formation of communities where people can share interests, discuss topics, and provide peer support.
- **Entertainment:** Many websites are dedicated to entertainment, offering streaming services, games, and interactive media to engage users.

Dear Students! Dear Students! Now that we've introduced what websites are and explored their diverse roles and functionalities, let's narrow our focus a bit. Can you think of what makes up these websites? What are the essential building blocks that bring every website to life? In our next section on Web Pages, we'll explore the fundamental components that assemble a website, how they are created, and how they interact to present the seamless web experiences you encounter daily.

1.3.2. Web Pages

Web pages are documents written in HTML, which is a markup language that defines the structure and layout of content on the page. HTML tags are used to markup elements such as headings, paragraphs, images, links, and more.



1.3.2.1. Web Page Structure and Elements:

- **Web Page Structure:** The organization of a web page using HTML elements, including headers, paragraphs, lists, images, tables, forms, and more.
- **Hyperlinks:** Hyperlinks, often referred to simply as links, are the foundation of the Web. They allow users to navigate between web pages by clicking on text or objects that are linked to other pages or resources. Hyperlinks enable the interconnectedness and easy traversal

of information on the Web.

- **Multimedia Elements:** The inclusion of images, videos, audio, and other media types within web pages to enhance the user experience.
- **Forms and User Input:** The use of form elements to collect user input, such as text fields, checkboxes, radio buttons, dropdown menus, and submit buttons.

1.3.3. Types of Websites

Dear Students! Having understood what web pages are and how they form the basic structure of every website, let's consider the variety they offer. What different types of websites have you encountered in your internet browsing? How do they serve unique purposes? Next, in our discussion on the Types of Websites, we'll categorize the various websites you might come across and delve into what makes each type distinctive and suited to specific functions.

Websites can be broadly categorized into two types: static and dynamic. Each type serves different purposes and suits different needs, depending on how the content is generated and managed.

A. Static Websites

Static websites are made up of web pages with fixed content, which are coded directly in HTML. The content of each page does not change unless it is manually updated by the webmaster.

Characteristics:

- **Fixed Content:** Each page displays the same content to every visitor, making it consistent and reliable.
- **Quick to Load:** With no need to process data on the server side, static pages usually load faster than dynamic pages.
- **Simplicity:** They are easier to create and host, as they require less server-side infrastructure.

Typical Uses:

- **Landing Pages:** Often used for specific campaigns where information does not need to be dynamically generated.
- **Small Business Websites:** Suitable for businesses that need an online presence but do not require frequent updates.
- **Documentation:** Perfect for documentation sites where information changes are infrequent.

B. Dynamic Websites

Dynamic websites are capable of displaying different content from the same source code. They use server-side scripting languages like PHP, Python, or Node.js to dynamically build a page when a user visits the site.

Characteristics:

- **Interactive and Personalized Content:** Content can change based on user interactions or input, providing a personalized experience.
- **Server-side Processing:** These sites rely on server technology for functions like logging in users, managing user input, and storing data.
- **Flexibility:** More adaptable to complex functionalities like online stores or forums where content needs to be updated frequently.

Typical Uses:

- **E-commerce Sites:** These require dynamic pages for product listings, user accounts, and shopping carts.
- **Social Media Platforms:** Dynamic websites are necessary to handle real-time updates, user interactions, and content personalization.
- **Blogs and News Portals:** Where content is constantly updated and may need to be archived or searched.

Understanding the distinctions between static and dynamic websites can help in choosing the right approach depending on the project's needs and the desired level of interaction and content management.



Self-check 1-3:

Dear Students! So far, we have discussed what websites are and the different types of websites.

Now it is your turn to answer the following self-check questions

1. What is a website?
2. What is a web page?
3. What is the difference between static and dynamic websites?

Have you answered the self-check questions? If yes, let us Summarize the unit.

Unit Summary

In Unit One, we explore the basics of the Internet and the World Wide Web (WWW). The Internet is a vast global network connecting computers worldwide. We learn about its history and how it transformed over time. The WWW, created by Sir Tim Berners-Lee, allows seamless information sharing across computers. We see its evolution from simple text to today's dynamic, interactive web.

Understanding the WWW's mechanics is key. We look at the client-server model, where web servers house websites and respond to user requests through web browsers. Important technologies like HTML, HTTP, and web browsers make this communication possible.

Key components like web protocols, URLs, web servers, pages, hyperlinks, and browsers shape our online experiences. We explore HTTP and its secure version, HTTPS, for added security. The Domain Name System (DNS) acts as the Internet's directory, translating human-readable domain names into IP addresses.

We then dive into URLs, understanding their components and syntax. Transitioning to websites, we define them as collections of web pages on the WWW. Web pages, written in HTML, structure content and can be static or dynamic, serving different purposes.

The unit ends with a look at various website types, from personal blogs to e-commerce platforms. Understanding these types gives us a comprehensive view of the digital landscape. This unit is a foundation for deeper insights into the Internet, WWW, and websites.

Unit Review Questions

Part I: Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What is the Internet?
 - A. A local computer network
 - B. A global network of interconnected computers
 - C. An application for web browsing
 - D. A type of web browser

2. Who is credited with inventing the World Wide Web?
 - A. Mark Zuckerberg
 - B. Bill Gates
 - C. Tim Berners-Lee
 - D. Steve Jobs
3. Which protocol is fundamental for communication on the World Wide Web?
 - A. TCP
 - B. FTP
 - C. HTTP
 - D. IP
4. What distinguishes HTTPS from HTTP?
 - A. HTTPS is faster
 - B. HTTPS uses a different browser
 - C. HTTPS is more secure
 - D. HTTPS is only for static websites
5. What is the primary function of DNS?
 - A. Data encryption
 - B. IP address translation
 - C. Web page rendering
 - D. Browser installation

Part II: Short Answer Questions:

Instruction: Give brief answers for the following review questions:

1. Define the Internet and explain its role in global communication.
 1. How does the Domain Name System (DNS) facilitate the translation of domain names into IP addresses?
 2. List and explain the components of a Uniform Resource Locator (URL).
 3. Describe the characteristics of static and dynamic websites, highlighting their differences.
 4. Provide examples of different Top-Level Domains (TLDs) and their classifications.

Answer key for Self-check questions

Self-check 1-1: Answer

1. The internet is a vast global network connecting computers worldwide.
2. Different applications of the internet include email, web browsing, online gaming, and social media.
3. ISPs (Internet Service Providers) provide internet access. In Ethiopia, Ethio Telecom acts as an ISP.

Self-check 1-2: Answer

1. WWW (World Wide Web) is a revolutionary component of the internet for information sharing.
2. DNS (Domain Name System) translates human-readable domain names into IP addresses.
3. The internet is the global network, while the web is a system of interconnected documents on the internet.

Self-check 1-3: Answer

1. A website is a collection of web pages and associated content accessible through the WWW.
2. A web page is a document written in HTML, a component of a website.
3. Static websites have fixed content, while dynamic websites can change content based on user interactions.

UNIT 2

WEB BROWSERS

Unit Coverage

This unit is designed to provide you with the necessary information and practice regarding the following content coverage:

- 2.1. Introduction to Web browsers
- 2.2. Navigating and utilizing Web browser features
- 2.3. Customizing and Configure Web browser

Unit Learning Outcomes

This unit will also assist you in attaining the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Use and configure Web Browsers

Unit Overview

In the previous unit, you have gone through the concepts relating to Internet and the world wide web. The internet is a vast global network connecting computers worldwide. From the various services we get from the internet , world wide web is the core service.to surfe the web we need a special type of software called web browser. This unit focuses on providing students with a comprehensive understanding of web browsers, their features, and the effective utilization and customization of these tools. From the fundamental introduction to various advanced configurations, you will gain practical insights into navigating the digital landscape seamlessly. The unit is designed to equip you with the skills necessary to optimize their browsing experience and lays the groundwork for more advanced web development concepts.

Key Terms: *Browser, WWW, Configuration, Navigation, Bookmark, Synchronization*

2.1. Introduction to Web Browsers

Dear Students! What is a web browser? Can you mention the most popular browsers? Hoping that you have defined it and mentioned some browsers, let's see together what a browser is, The evolution of web browsers and some of the most common types of browsers.

What is a Web Browser?

A web browser is a software application that allows users to access and interact with content on the World Wide Web (WWW). It serves as a user interface for navigating the Internet, retrieving web pages, and displaying them on a computer or device. Web browsers are essential tools for accessing a wide range of online resources, including websites, web applications, multimedia content, and more. They enable users to input web addresses (URLs), search for information, view text and images, watch videos, and interact with web-based forms and services.

The primary functionality of web browsers is to interpret and render web content. They act as intermediaries between users and web servers, translating HTML, CSS, and other code into the visually appealing websites we see. Browsers enable users to navigate through different pages, follow links, and interact with web-based applications.

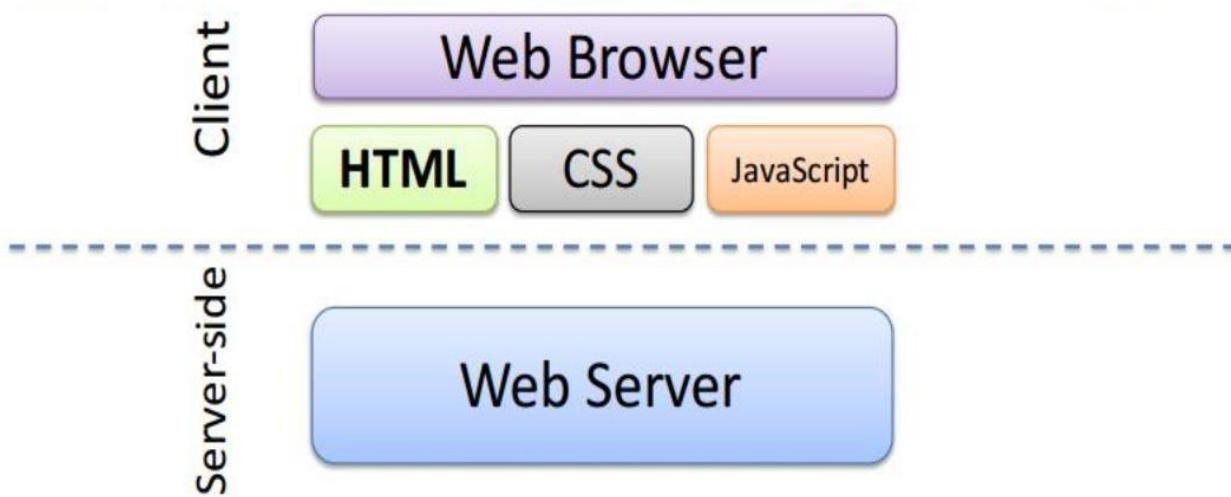


Figure 2.1: Browser Architecture

2.1.1. Evolution of Web Browsers

The evolution of web browsers traces a fascinating history of technological advancements. Starting from simple text-based browsers like Lynx, the journey progressed to graphical browsers such as Mosaic. Netscape Navigator played a pivotal role in popularizing the web, leading to the eventual dominance of Internet Explorer. The browser landscape has continually evolved, with modern browsers like Chrome, Firefox, Safari, and Edge offering enhanced speed, security, and features.

2.1.2. Popular Web Browsers:

Several web browsers cater to diverse user preferences and requirements. Understanding the features and functionalities of popular web browsers empowers users to choose the one that aligns with their needs. Commonly used browsers include:



Figure 2.1: Common Web Browsers

- 1. Google Chrome:** Developed by Google, Chrome is one of the most popular web browsers. It is known for its speed, simplicity, and seamless integration with Google services. Chrome offers a clean, user-friendly interface, and it supports a vast library of extensions and add-ons, making it highly customizable.
- 2. Apple Safari:** Safari is Apple's web browser, available on macOS and iOS devices. It is recognized for its speed and energy efficiency. Safari offers seamless integration with Apple's ecosystem and is designed to deliver a smooth, user-friendly experience on Apple hardware.
- 3. Mozilla Firefox:** Firefox is an open-source web browser developed by the Mozilla Foundation. It is celebrated for its strong privacy features and customizable add-ons. Firefox emphasizes user security, offering enhanced privacy settings and features like "container tabs" to keep browsing activities isolated.
- 4. Microsoft Edge:** Microsoft Edge is the default web browser for Windows 10 and later. It is built on the Chromium engine (the same core technology as Chrome) and is known for its speed and compatibility with web standards. Edge also integrates with Microsoft's ecosystem, offering features like Cortana and Windows Hello.

5. Opera: Opera is a feature-rich web browser with a focus on speed and privacy. It includes a built-in ad blocker and a free VPN service. Opera's sidebar offers quick access to various tools and services, making it a popular choice for users who value convenience and customization.

These web browsers share core features such as tabbed browsing, bookmark management, and password storage. However, they differentiate themselves through speed, privacy features, customizability, and integration with other services and platforms. Users can choose the web browser that best suits their preferences and needs for web browsing.



Self-check 2-1:

Dear Students! We have discussed what a web browser is and the different types of browsers. Now it is your turn to answer the following self-check questions.

1. What is a web browser?
2. Name at least three popular web browsers and one unique feature of each.
3. How do web browsers contribute to web development?

Have you answered the self-check questions? If yes, let us Navigate and utilize Web browser features.

2.2. Navigating and utilizing Web browser features

Dear Students! How do you think we can effectively navigate and utilize the features of these browsers? In our next topic, we will learn about navigating and utilizing web browser features to enhance our browsing experience.

2.2.1. Browsing and Navigating the Browser Interface

Browsing also called Surfing the process of looking into the information available in the web. One of the preconditions for browsing is that you should be connected to the Internet. Web browsers offer a range of features to enhance user experience and streamline navigation. Understanding how to efficiently use and navigate these features is crucial for a seamless online experience.

Effective information management when using web browsers is crucial in today's digital age. It can significantly enhance productivity, save time, and help users stay organized. Here are some key areas of focus when discussing the importance of effective information management:

- **Time Efficiency:** Proper information management helps users find what they need quickly, reducing the time spent searching for websites, bookmarks, or saved pages. Time is saved by having easy access to frequently visited websites through bookmarks or a well-organized homepage.
- **Productivity:** Organized bookmarks and saved web pages can increase productivity by allowing users to focus on tasks instead of searching for information repeatedly. Customized homepages can provide a quick overview of important information, such as news, emails, or calendar events, to kickstart the workday.
- **Content Retention:** Effective information management ensures that valuable content is retained for future reference. Saved web pages and organized bookmarks help users remember and revisit articles, resources, or tools they found useful.
- **Cross-Device Accessibility:** Synchronization of browser settings and bookmarks allows users to access their information on various devices, including computers, smartphones, and tablets. This is essential for maintaining continuity and staying productive across different platforms.
- **Enhanced Security:** Effective information management includes using browser features like password managers to secure sensitive data. Managing cookies and privacy settings can also help protect personal information from potential threats.

2.2.1.1. Basic Navigation in a browser window

Navigating the browser interface involves mastering the basics of forward and backward navigation, refreshing pages, and utilizing the address bar for direct access to websites. Understanding keyboard shortcuts and mouse gestures can further enhance navigation efficiency. As stated earlier, a web browser is an application that allows you to interact with various web sites on the Internet. You can run a browser from your Windows Desktop. *Figure 8* below shows some of the important components that are available on a google chrome browser window:

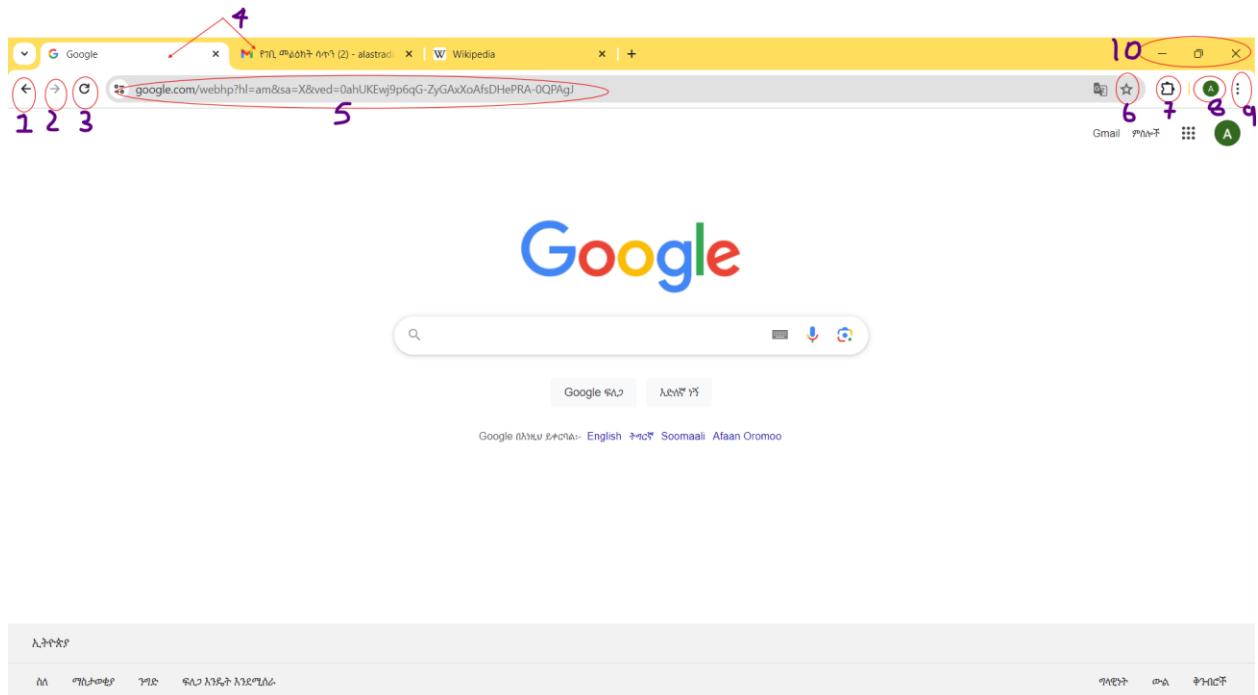


Figure 3: Google Chrome Interface

Understanding the various components of the Google Chrome browser window is crucial for efficient navigation and browsing. Here's a brief explanation of each key element:

1. **Back Button** : Clicking The Back button allows you to return to the previous page you were viewing.
2. **Forward Button** : Clicking The Forward button takes you to the next page if you have navigated back and want to go forward again.
3. **Refresh Button** : Click the Refresh button (a circular arrow) to refresh the page content. This is useful if a page did not load correctly or you want to see the latest updates.
4. **Tabs:** Browser tabs allow opening many websites on a single web browser's window - very helpful when reading several websites at the same time. Click the "+" button next to the last tab to open a new tab. You can switch between open tabs by clicking on them or using keyboard shortcuts (Ctrl+Tab).
5. **Address bar:** The Address Bar displays the URL of the current web page and can be used to navigate to other websites. Type a URL or search term into the Address Bar and press Enter to go to a specific site or perform a web search.

6. **Bookmark buttons** : The Bookmark button allows you to save your favorite web pages for easy access later.
7. **Extensions** : The Extensions button provides access to the browser extensions you have installed. Click the puzzle piece icon to view and manage your extensions. This is where you can enable, disable, or configure extensions.
8. **Profile icon** : The Profile icon shows which Google account is currently signed in and allows you to switch profiles. Click the Profile icon (a user icon) to switch between different Google accounts, manage your account settings, or access Chrome profiles.
9. **Menu:** The Menu provides access to various browser settings and tools. Click the three vertical dots in the upper-right corner to open the Menu. From here, you can open new tabs/windows, access history and downloads, and adjust browser settings.
10. **Windows Management icons** : These buttons control the size and visibility of the browser window.
 - Minimize:** Click the "-" button to minimize the browser window to the taskbar.
 - Maximize:** Click the "□" button to expand the browser window to fill the screen.
 - Close:** Click the "X" button to close the browser window completely.

2.2.2. Utilizing and Managing Web Browser Features

Effective web browsing involves mastering fundamental components that enhance navigation, organization, and personalization within a web browser. These components serve as the building blocks for a seamless and productive online experience. Understanding and harnessing these elements contribute to efficient and customized browsing habits. In the next section you will practically explore The key components.

in this lesson For the practical activities we will use the Google Chrome web browser. For ease of access, we will open Google Chrome from the taskbar. using the steps below add (pin) the Google Chrome app to the taskbar.



Steps: To add the Google Chrome app to the taskbar:

1. Click on the search icon  in the taskbar.
2. Type: **Google Chrome**.
3. Click on the **Google Chrome** app icon. This will launch the web browser.
4. Right click on the **Google Chrome** icon on the taskbar.
5. Select **Pin to Taskbar** from the menu.



6. Click the **Google Chrome** icon on the taskbar to open the web browser.

2.2.2.1. Tabs and Windows Management

Tabs and windows are fundamental components of web browsing. This refers to the ability to open multiple tabs and manage them within a browser window, allowing users to navigate between different web pages efficiently without cluttering their screen with multiple windows. Different browsers support these features but for this unit will explore how to manage tabs in Google Chrome.

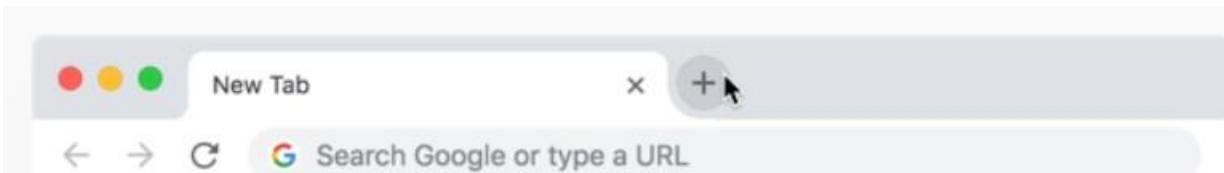
How to Manage Tabs in Google Chrome ?

Managing tabs effectively in Google Chrome is essential for a streamlined web browsing experience. Here are the different tab management skills you should master, along with the steps for each skill:



Steps: Tab Management in Google Chrome

- 1. Opening a New Tab:** Opening a new tab allows you to visit another webpage without leaving your current one.



Steps:

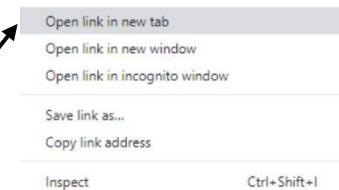
1. Click the "+" icon next to the last tab.

Or Press **Ctrl + T** (Windows/Linux) or **Cmd + T** (Mac).

2. Opening a Link in a New Tab or in new window: Opening a link in a new tab lets you explore linked content without navigating away from your current page. And Opening a tab in a new window helps you manage different tasks or projects separately.

Steps:

1. Right-click on the link you want to open.
2. Select "Open link in new tab." Or "Open link in new window."



Or Press **Ctrl** (Windows/Linux) or **Cmd** (Mac) and click the link.

3. Closing a Tab: Closing tabs you no longer need keeps your workspace tidy.

Steps:

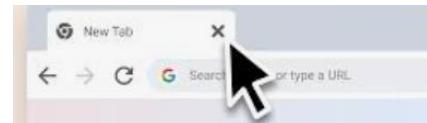
1. Click the "X" icon on the tab you want to close.

or

Press **Ctrl + W** (Windows/Linux) or **Cmd + W** (Mac).

or

Right-click on the tab and select "Close."



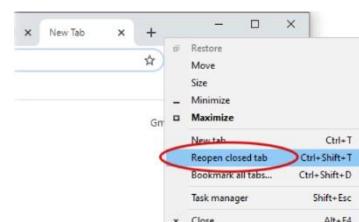
4. Reopening a Closed Tab: Reopening a closed tab helps you quickly recover tabs you might have closed by mistake.

Steps:

1. Right-click on any open tab and select "Reopen closed tab."

Or

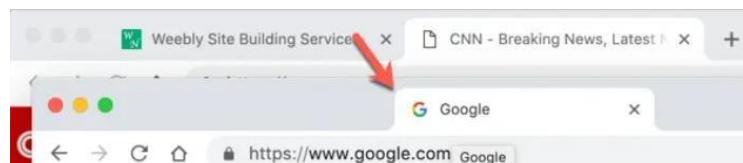
2. Press **Ctrl + Shift + T** (Windows/Linux) or **Cmd + Shift + T** (Mac).



5. Moving a Tab: Rearranging your tabs can help you keep related content together.

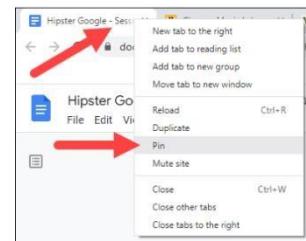
Steps:

1. Click and hold the tab you want to move.
2. Drag it to the desired position in the tab bar.
3. Release the mouse button to drop the tab in place.

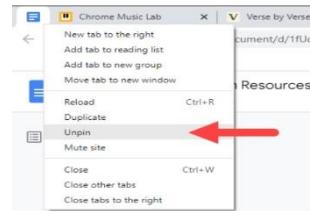


6. Pinning a Tab: Pinning tabs minimizes them and keeps frequently used tabs accessible.**Steps:**

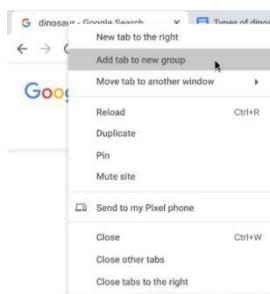
1. Right-click on the tab you want to pin.
2. Select "Pin tab."
3. The tab will shrink and move to the leftmost part of the tab bar.

**7. Unpinning a Tab:** Unpinning a tab returns it to its normal size and position in the tab bar.**Steps:**

1. Right-click on the pinned tab.
2. Select "Unpin tab."
3. The tab will return to its normal size and position.

**8. Grouping Tabs :** Grouping tabs helps you organize related tabs together, making it easier to manage multiple tasks.**Steps:**

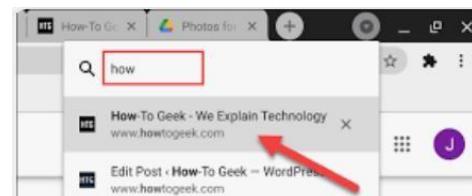
1. Right-click on a tab.
2. Select "Add tab to new group."
3. Name the group and choose a color if desired.
4. Drag other tabs into the group to add them.

**9. Ungrouping Tabs:** Ungrouping tabs removes them from a group and places them back in the regular tab bar.**Steps:**

1. Right-click on the tab group name.
2. Select "Ungroup."

**10. Using the Tab Search Feature:** The tab search feature helps you quickly find specific tabs when you have many open.**Steps:**

1. Click the down arrow icon at the top right of the tab bar.
2. Type keywords to search for a specific tab.
3. Click on the desired tab from the search results.



2.2.2.2. Managing Bookmarks

Bookmarks and favorites are tools for saving and organizing frequently visited websites. Learning to create, edit, and organize bookmarks enables users to quickly access their preferred online destinations, enhancing overall productivity. Bookmarks allow you to easily revisit webpages, without the need to search for or memorize URLs. This is a way of saving webpages, so next time you want to visit the webpage you just have to choose it from a menu. You can use folders so that you can group related bookmarks, or bookmarks with a common theme. Organizing bookmarks will make it easier for you to quickly find a webpage you want to revisit.



Note: that in some web browsers, such as Microsoft Edge bookmarks are referred to as Favorites.

How to Managing bookmarks in Google Chrome?

Create bookmarks so Chrome can remember your favorite and frequently visited websites. When you sign in to Chrome with your Google Account, you can use bookmarks and other info on all your devices. Here are the steps to manage bookmarks in google chrome These skills will help you efficiently manage your bookmarks in Google Chrome, enhancing your browsing experience and organization.



Steps: Bookmark Management in Google Chrome

1. **Add a bookmark:** Adding a bookmark saves the current webpage for easy access later.

Steps:

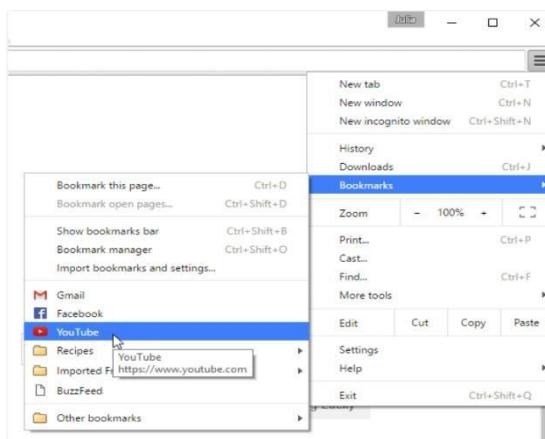
1. On your computer, open Chrome.
2. Go to the site you want to visit again in the future.
3. To the right of the address bar, select **Bookmark ☆**.



2. Accessing Bookmarks: Accessing bookmarks allows you to quickly visit your saved websites.

Steps:

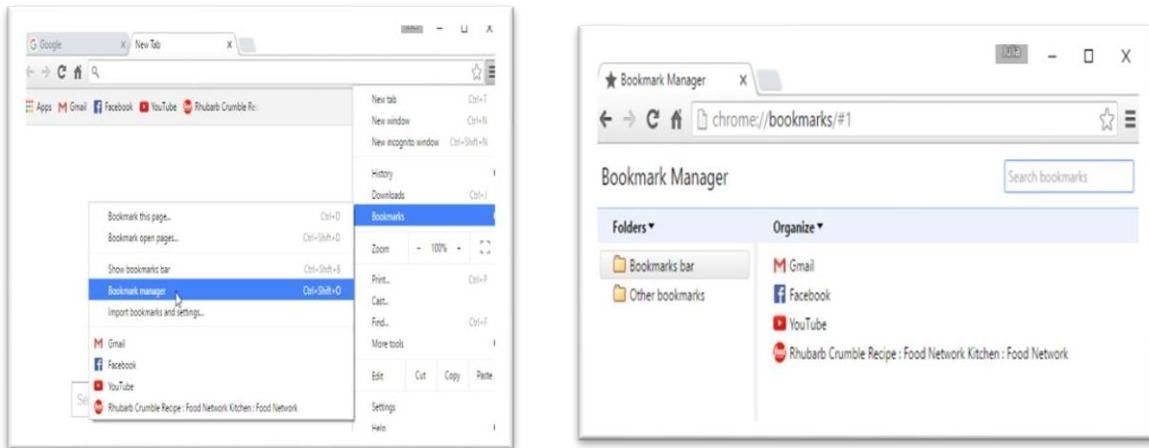
1. Click the three dots menu in the upper-right corner of Chrome.
2. Hover over "**Bookmarks**."
3. Select the bookmark you want to visit from the list.



3. Organizing Bookmarks into Folders: Organizing bookmarks into folders helps you keep related bookmarks together.

Steps:

1. Click the three dots menu in the upper-right corner of Chrome.
2. Hover over "**Bookmarks**" and select "**Bookmark manager**."
3. Click the three dots menu in the Bookmark manager and select "**Add new folder**."
4. Name the folder and drag and drop bookmarks into it.



4. Edit a bookmark: Editing bookmarks allows you to change the name, URL, or folder location of a bookmark.

Steps

1. On your computer, open Chrome.
2. At the top right, select More > **Bookmarks and lists** > **Bookmark Manager**.

3. Right click on the bookmark you want ➤ click **Edit**.

5. Delete a bookmark

Steps

1. On your computer, open Chrome.
2. At the top right, select More ⋮ ➤ **Bookmarks and lists** ➤ **Bookmark Manager**.
3. Right click on the bookmark you want ➤ Click **Delete**.
6. **Importing Bookmarks:** Importing bookmarks allows you to bring bookmarks from another browser into Chrome.

Steps:

1. On your computer, open Chrome.
 2. At the top right, select More ⋮ ➤ **Bookmarks and lists** ➤ **Bookmark Manager**.
 3. On the right, select More ⋮ ➤ **Import bookmarks**
 7. **Exporting Bookmarks:** Exporting bookmarks allows you to save your Chrome bookmarks for use in another browser.
- Steps:**
1. On your computer, open Chrome.
 2. At the top right, select More ⋮ ➤ **Bookmarks and lists** ➤ **Bookmark Manager**.
 3. On the right, select More ⋮ ➤ **Export bookmarks**

2.2.2.3. Managing Browser History

Browser history is a record of all the web pages a user has visited over a certain period of time, maintained by the web browser. When you use any web browser, various types of data can be stored in your browsing history. This stored data helps improve your browsing experience by making it easier to revisit previously visited sites, autocomplete forms, and resume activities where you left off. Here's a detailed list of the different types of data that can be stored in your browser history:

- **Browsing History:**
 - **URLs of Visited Pages:** The web addresses (URLs) of the pages you have visited.
 - **Page Titles:** The titles of the web pages you have visited.

- **Visit Timestamps:** The dates and times when you visited each page.
- **Download History:** A record of files you have downloaded, including filenames and download dates (note that the actual files are stored on your device separately).
- **Cookies and Site Data:**
 - **Cookies:** Small files stored by websites to remember your login status, preferences, and other information.
 - **Session Data:** Information about your current browsing session, such as items in a shopping cart.
- **Cached Images and Files:** Copies of web pages, images, and other media content saved to speed up loading times on future visits.
- **Passwords (Saved Login Credentials):** Usernames and passwords for websites you choose to save for easy access.
- **Autofill Form Data (Form Entries):** Information you have entered into forms, such as names, addresses, phone numbers, and email addresses.
- **Search History (Search Queries):** Records of the search terms you have entered into search engines.
- **Bookmarks:** URLs of pages you have bookmarked for quick access.
- **Site Settings (Permissions):** Settings you have configured for specific websites, such as whether a site can use your camera, microphone, or location.
- **Hosted App Data:** Data from Chrome apps and extensions, including settings and stored information.

But It doesn't store:

- Chrome pages like chrome://settings
- Pages you've visited in private browsing
- Deleted pages from your browsing history

How to View and Manage Stored Data in Google Chrome?

Google chrome stores Your History lists the pages you've visited in the last 90 days. The following steps will help you to Find your history, Find a page from your history, Manage Saved Passwords, Managing Autofill Form Data and Delete your history in google chrome.



Tip:

If you're signed in to Chrome and sync your history, then your History also shows pages you've visited on your synced devices.

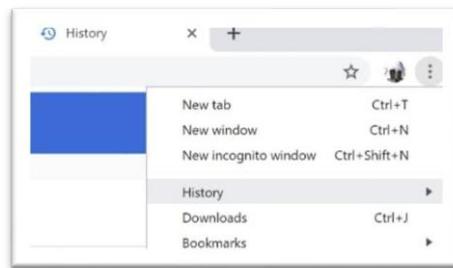


Steps: Manage Stored Data in Google Chrome

1. **Access Browser History:** Accessing your browser history lets you review and revisit the websites you've previously visited.

Steps:

1. On your computer, open Chrome.
2. At the top right, click More .
3. Click **History > History**.



2. **Find a Page from Your History :** Finding a page from your history allows you to quickly locate and revisit specific webpages.

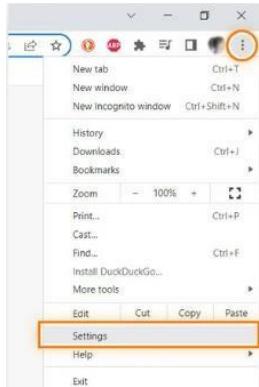
Steps:

1. On your computer, open Chrome.
2. In the address bar, enter `@history`.
3. Press **tab** or **space**. You can also click Search History  in the suggestions.
4. Enter keywords for the page you previously visited.
5. Select the page from the list.

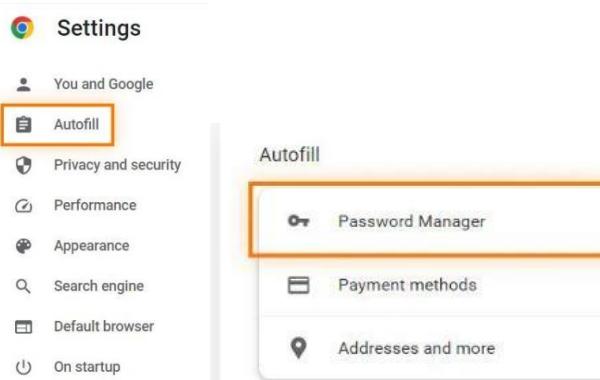
3. **Manage Saved Passwords:** Managing saved passwords helps you keep track of and secure your login credentials for various websites.

Steps:

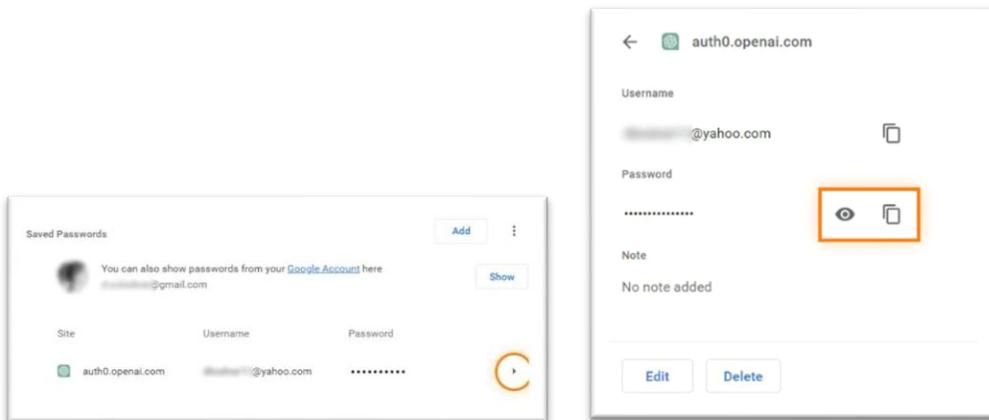
1. Click the three dots menu in the upper-right corner of Chrome.
2. Select "Settings" from the dropdown menu.



3. Under "Autofill," click "Passwords."



4. Here you can view, add, edit, or delete saved passwords.



4. Managing Autocomplete Form Data in Google Chrome: Autocomplete form data in Google Chrome helps save time by automatically filling in frequently used information such as addresses, phone numbers, and email addresses. Here's a step-by-step guide on how to manage your autocomplete form data:

Steps:

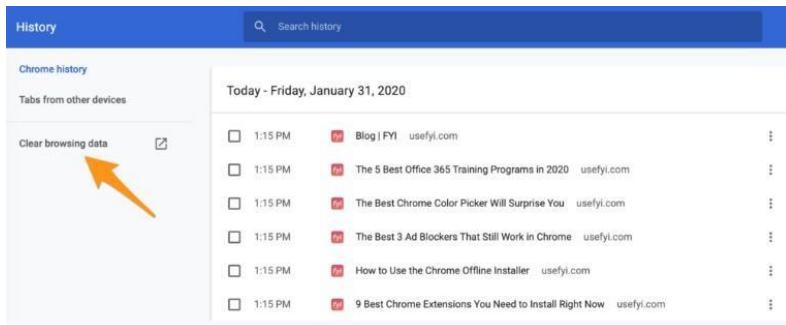
1. Click the three dots menu in the upper-right corner of Chrome.
 2. Select "Settings" from the dropdown menu.
 3. Under "Autocomplete," click "Addresses and more."
 4. Here you can add, edit, or delete saved addresses and payment methods.
- 5. Delete Your History:** Deleting your history helps you maintain privacy by removing records of the websites you've visited. If you don't want a record of pages you visited in Chrome, you can delete all or some of your browsing history. When you delete your browsing history in Google Chrome, you have the option to clear various types of data.

Steps:

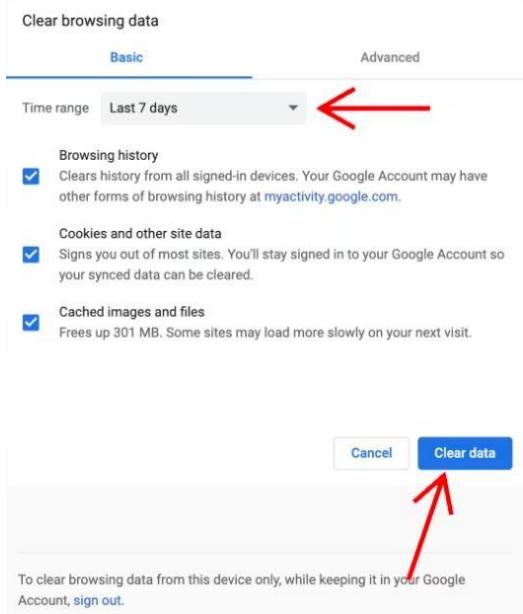
1. On your computer, open Chrome.
2. At the top right, click More .
3. Click History > History.



4. On the left, click **Clear browsing data**.



5. Select how much history you want to delete.
 - To clear everything, select **All time**.
6. Check the boxes for the info you want Chrome to clear, including **Browsing history**.

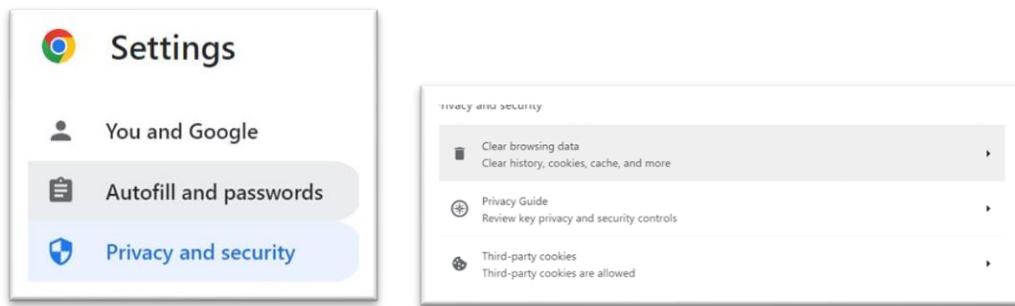


7. Click **Clear data**.

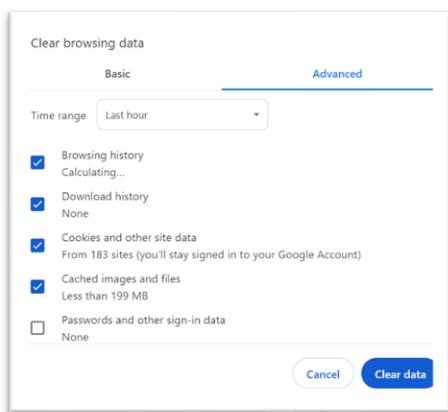
6. **Delete Your Activity:** Deleting your activity removes various types of browsing data to enhance privacy. When you use Google sites, apps, and services, some of your activity is saved in your Google Account. You can find and delete this activity in My Activity, and you can stop saving most activity at any time.

Steps:

1. Click the **three dots ::** menu in the upper-right corner of Chrome.
2. Select "**Settings**" from the dropdown menu.
3. Scroll down and click "**Privacy and security**."



4. Click "**Clear browsing data**."
5. Choose the time range and types of data you want to delete, including browsing history, cookies, cached images, and more.



6. Click "**Clear data**."

2.2.2.4. Downloading and Managing Files

This feature allows users to download files from the internet and save them to their computer or another specified location. It's essential for acquiring documents, images, videos, and other files

from the web for personal or professional use. Different browsers support file downloading however in this section we will demonstrate the file download and related operations using google chrome.

How to Download and Manage Files in Google Chrome ?



Steps: Manage Stored Data in Google Chrome

1. **Download a file:** To save a file or image on your computer or device, download it. The file will be saved in your default download location.

Steps:

1. On your computer, open Chrome.
2. Go to the site where you want to download the file.
3. Save the file:
 - **Most files:** Click the download link. You can also right-click on the file and choose **Save as**.
 - **Images:** Right-click on the image and choose **Save Image As**.
 - the video's owner or hosting site has prevented downloads.
 - **PDFs:** Right-click on the file and choose **Save Link As**.
 - **Web pages:** At the top right, click More ☰ > More Tools > **Save Page As**.
4. If asked, choose where you want to save the file, then click **Save**.
 - **Executable files (.exe, .dll, .bat):** If you trust the file, click **Save**. If you're not sure about the contents of the download, click **Discard**.
5. When you begin a download, a Download in progress icon (a blue circle with a white downward arrow) appears on the top right next to the address bar. Once the download completes, the Download tray opens.
6. To open your file, click Open new (a small square icon).
 - You can also click the file to open it.

Tips:

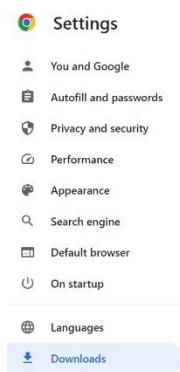
- To show extra actions like Show in Folder, point to the filename.
- If you download a file, or if you've recently downloaded a file, the Download tray will appear. Recently downloaded files will appear to the right of the address bar.
- To view all downloads if the Download tray isn't present to the right of the address bar, click More > **Downloads**.
- You can drag a downloaded file to another folder, program, or website. To move a downloaded file, in the Download tray, click the file and drag it to the target location.

2. Find a list of files you've downloaded

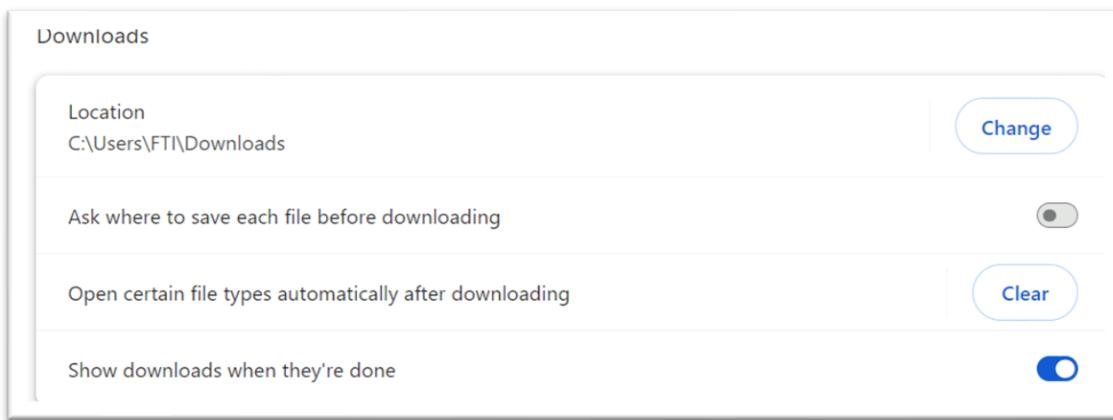
Steps:

1. On your computer, open Chrome.
2. At the top right, click More ⚙ > Downloads.
 - To open a file, click its name. It will open in your computer's default application for the file type.
 - To remove a download from your history, to the right of the file, click Remove ✘. The file will be removed from your Downloads page on Chrome, not from your computer.
3. **Change download locations:** You can choose a location on your computer where downloads should be saved by default or pick a specific destination for each download.

1. On your computer, open Chrome.
2. At the top right, click More ⚙ > Settings > Downloads.
3. Adjust your download settings:
 - To change the default download location, click **Change** and select where to save your files.
 - If you'd rather choose a specific location for each download, turn on **Ask where to save each file downloading**.



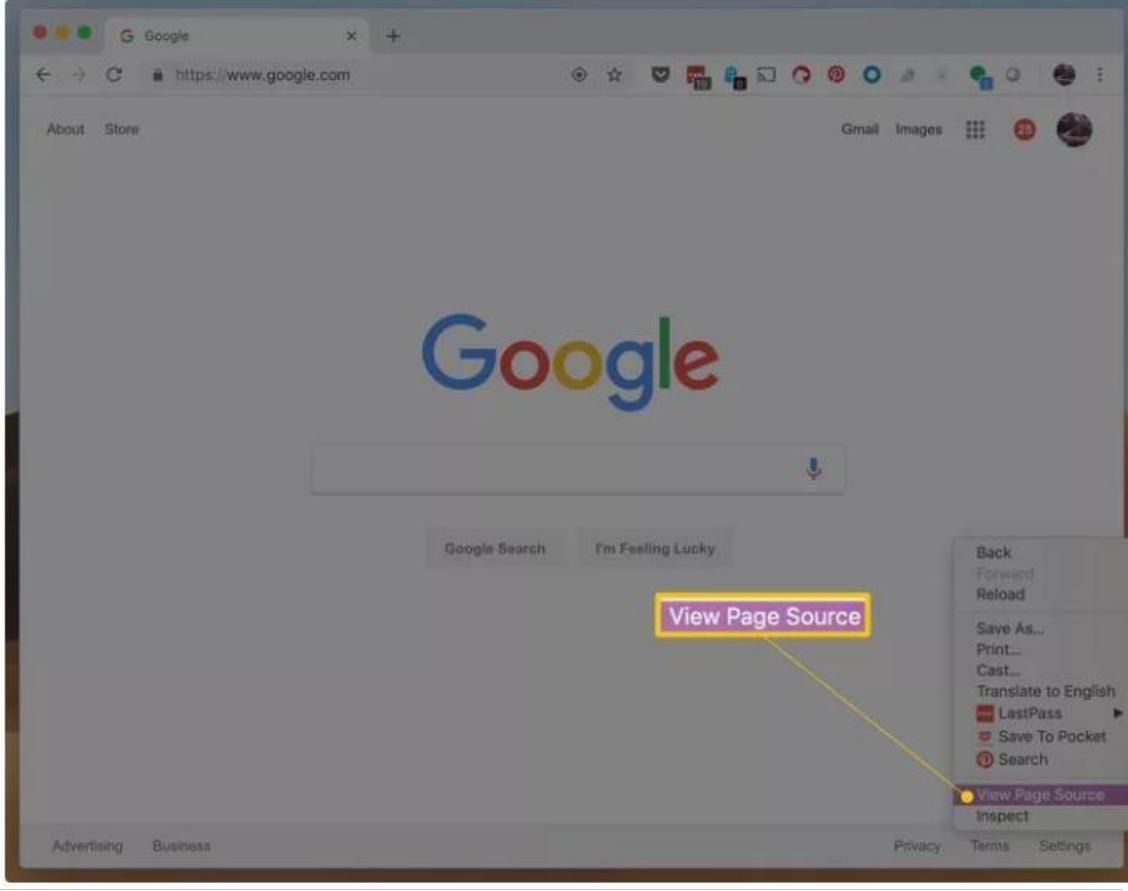
before



2.2.2.5. View source code of a website

Page Source is the literal code that makes up a web page. Viewing the source code of a website allows you to see the HTML, CSS, and JavaScript that make up the web page. This can be useful for learning how websites are built, debugging issues, or simply satisfying curiosity about the underlying structure of a web page. Viewing the source code of a website is a straightforward process that can be done using either the "View Page Source" option or the Developer Tools. The "View Page Source" option provides a static view of the HTML code, while the Developer Tools offer a more dynamic and interactive environment for exploring and manipulating the HTML, CSS, and JavaScript of the web page. This capability is invaluable for web developers, students, and anyone interested in understanding the inner workings of web pages.

| | |
|---|---|
|  | Steps: Manage Stored Data in Google Chrome |
| | 1. View Source Code in Chrome Steps: <ol style="list-style-type: none">1. Open the Google Chrome web2. Navigate to the web page you would like to examine.3. Right-click the page and look at the menu that appears. From that menu, click View page source. |



The screenshot shows a Google search results page in Google Chrome. A right-click context menu is open on the page, with the option 'View Page Source' highlighted by a yellow box and a yellow arrow pointing to it from below.

4. The source code for that page will now appear as a new tab in the browser.
5. Alternatively, you can also use the keyboard shortcuts of **Ctrl+U** on a PC to open a window with a site's source code displayed. On a Mac, this shortcut is **Command+Option+U**.

2. Inspect Code Using Developer Tools:

Steps:

1. Open the **Google Chrome** web
2. Navigate to the **web page you would like to examine**.
3. Right-click **the page** and look at the menu that appears. From that menu, click **View page source**.

OR Press **Ctrl+Shift+I (Windows/Linux) or **Cmd+Option+I** (Mac) to open the Developer Tools.**



Self-check 2-2:

Dear Students! So far, We have discussed how to Utilizing and Managing Web Browser Features and how to configure different Browser settings. Now it is your turn to answer the following self-check questions

1. What are browser tabs, and why are they useful?
2. What is the function of the browser's address bar?
3. What is the role of browser history in web browsing?
4. What is the use of Bookmarks?

Have you answered the self-check questions? If yes, let us continue to the next section.

2.3. Customizing and Configuring Web browser

Dear Students! Can you try to recall the key features and navigation tools we discussed in the previous section? Hoping that you have explored these features, let's see together how they enhance our browsing experience. How do you think customizing and configuring a web browser can improve your browsing experience? In our next topic, we will explore the various customization and configuration options available in web browsers to make them work best for you.

Efficient browsing goes beyond navigation; it involves configuring your browser to align with your preferences and ensure a secure online experience. Configuring the basic settings of your web browser is a crucial step in personalizing your online experience, enhancing security, and optimizing performance. These settings provide users with the ability to tailor the browser to their preferences and requirements. In this section, we will explore various aspects of browser configuration, covering topics such as General settings, privacy and security controls, Cookies and pop-ups control, managing extensions, and syncing browsing data across devices.

2.3.1. General Settings

Web browsers like Google Chrome offer a variety of settings that you can configure to personalize your browsing experience, enhance security, and improve functionality. These settings include homepage and startup options, search engine preferences, appearance customization, language and advanced settings for system performance. Understanding and configuring these settings can help

you optimize your browser to suit your needs. The following section will demonstrate you about the steps to configure General Settings That Can Be Configured in Google Chrome

2.3.1.1. Change your language on the web

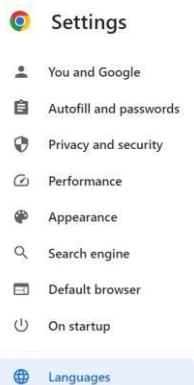
Google services are available in all Google languages. You can change the display language to your preferred language at any time. These instructions are to change your preferred language used in Google services on the web only. To change the preferred language for your device update the language settings on your device.



Steps: Change your web language settings in Google Chrome

Steps to Change your web language settings:

1. Click the **three dots** menu in the upper-right corner of Chrome.
2. Select "**Settings**" from the dropdown menu.
3. Under "settings," click **Language >**



4. Search for and select your preferred language.
5. Click **Select**.
6. If you understand multiple languages, click + **Add another language**.

Preferred languages



After you change your language preferences, close and reopen your browser.

2.3.1.2. Modify Homepage and startup options:

The website that appears every time you open your web browser is known as the home page. Customizing the browser's homepage and startup options allows users to tailor their browsing experience. You can set the home page to any website you wish. In addition, Browsers allows you to set multiple tabs (pages) to open on start-up.

How to Set your homepage and startup page in google chrome ?

You can customize Google Chrome to open any page for the homepage or startup page. There are three options available to set a homepage in google chrome. You can Have a new tab open, or you can Continue from where you left off or you can also Open a specific set of pages. Here are the steps:

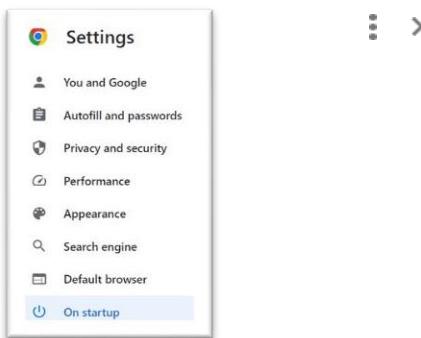


Steps: To set your homepage and startup page in Google Chrome

1. **Have a new tab open:** You can set Chrome to open a New Tab page whenever you open a new window.

Steps:

1. On your computer, open Chrome.
2. At the top right, select More **Settings**.



3. On the left, select **Open the New Tab page**.

On startup

- Open the New Tab page
- Continue where you left off
- Open a specific page or set of pages

2. **Continue where you left off:** You can tell Chrome to re-open the same pages you were looking at when you quit.

Steps:

1. On your computer, open Chrome.
2. At the top right, select More  > Settings.
3. On the left, select **On startup** > **Continue where you left off**.

Your cookies and data are saved, so any websites you were logged into before open again.

4. **Open a specific set of pages:** You can tell Chrome to open to any webpage.

Steps:

1. On your computer, open Chrome.
2. At the top right, select More  > Settings.
3. Under "On startup," select **Open a specific page or set of pages**.
4. You can either:
 - Select **Add a new page**.
 1. Enter the web address.
 2. Select **Add**.
 - Select **Use current pages**.

Tip: To update your pages, on the right, select More  > **Edit or Delete**.

2.3.1.3. Themes and Appearance Settings

Applying themes and adjusting appearance settings can significantly change the look and feel of your browser, allowing for a more personalized and enjoyable browsing experience. Personalize Chrome on your computer with a fun theme. The theme appears around the border of the browser and as the background when you open a new tab. Your Chrome themes are stored in your Google

Account, so when you sign in to Chrome on any computer, you see the same theme. Here are the steps to add and remove themes in google chrome.



Steps: Applying themes and appearance in Google Chrome

1. Download and add a Chrome theme

Steps:

1. On your computer, open Chrome.
2. At the top right, click More > Settings.
3. Under "Appearance," click Themes. You can also go to the gallery by visiting Chrome Web Store Themes.
4. Click the thumbnails to preview different themes.
5. When you find a theme you'd like to use, click Add to Chrome.

The theme will be applied immediately. If you change your mind, at the top, click Undo.

2. Remove a Chrome theme

Steps:

1. On your computer, open Chrome.
2. At the top right, click More > Settings.
3. Under "Appearance," click Reset to default. You'll see the classic Google Chrome theme again.

2.3.1.4. Setting search engine

Managing search engines in a web browser involves adding, removing, and setting preferences for different search engines. This allows users to customize their search experience based on their preferences. The default search engine is the search engine that a web browser uses when you type a query into the address bar (also known as the omnibox in Google Chrome) and press Enter. It's the search engine that the browser automatically directs your search queries to unless you specify otherwise.

Common Default Search Engines:

- **Google:** Known for its speed, relevance, and vast index of web pages.
- **Bing:** Microsoft's search engine, which integrates well with Windows and Microsoft services.
- **Yahoo:** Provides a variety of services including search, news, and email.
- **DuckDuckGo:** Focuses on user privacy and does not track search activity.

Here are the steps to Set your default search engine and Manage search engines and site shortcuts in google chrome.



Steps: Applying themes and appearance in Google Chrome

1. Set your default search engine

Steps:

1. On your computer, open Chrome.
2. At the top right, select More > Settings.
3. Select Search engine.
4. Next to "Search engine used in the address bar," select the Down arrow .
5. Select a new default search engine.

3. Manage search engines and site shortcuts:

You can add, edit, or remove site search shortcuts and set a default search engine.

Steps:

1. On your computer, open Chrome.
2. At the top right, select More > Settings.
3. On the left, select Search engine > Manage search engines and site search.
4. To change site search shortcuts:
 - **Add:** To the right of "Site search," select Add. After you fill out the text fields, select Add.
 - **Edit:** To the right of a site search shortcut, select Edit .
 - **Set as default:** To the right of a site search shortcut, select More > Make default.
 - **Deactivate:** To the right of a site search shortcut, select More > Deactivate.
 - **Delete:** To the right of a site search shortcut, select More > Delete.

2.3.2. Managing Extensions

Extensions and add-ons enhance browser functionality. Managing these features allows users to tailor the browser to their specific needs, adding tools and capabilities that align with individual preferences.



Tip: What are Extensions?

Extensions are small software programs that enhance the functionality of web browsers. They add new features, modify existing functionalities, or provide tools to improve the overall browsing experience. Examples: Ad blockers, password managers, language translators, and productivity tools.

How to Manage Extensions in Google Chrome?

Managing extensions involves installing, enabling, disabling, updating, and removing extensions from the browser. Here are the steps to manage extensions in Google Chrome:



Steps: Managing Extensions in Google Chrome

1. Accessing the Extensions Menu

1. Open Google Chrome.
2. Click the three vertical dots in the upper-right corner to open the menu.
3. Hover over "More tools."
4. Click on "Extensions."

2. Installing Extensions

1. Access the Extensions menu as described above.
2. Click the three horizontal lines (menu icon) in the upper-left corner of the Extensions page.
3. Click "Open Chrome Web Store."
4. Browse or search for the desired extension.
5. Click on the extension to view details.
6. Click "Add to Chrome."
7. A dialog box will appear. Click "Add extension" to confirm the installation.

3. Enabling and Disabling Extensions

1. Access the Extensions menu as described above.
2. Find the extension you want to enable or disable.
3. Toggle the switch next to the extension to enable (switch on) or disable (switch off) it.

4. Removing Extensions

1. Access the Extensions menu as described above.
2. Find the extension you want to remove.
3. Click the "Remove" button.
4. A confirmation dialog will appear. Click "Remove" again to confirm.

5. Updating Extensions

1. Access the Extensions menu as described above.
2. Toggle the "Developer mode" switch in the upper-right corner.
3. Click the "Update" button to update all extensions to the latest version.

2.3.3. Privacy and Security Settings

Privacy and security settings are paramount for safeguarding personal information while browsing. Configuring these settings empowers users to manage tracking, protect against phishing attempts, and control website access to sensitive data and it Enhance Online Security. We will explore more about these setting in unit five of this module.

2.3.4. Syncing Browsing Data Across Devices

Synchronization, often referred to as "sync" in web browsers, is a feature that allows you to link your web browser settings, bookmarks, history, saved passwords, and other data across multiple devices, such as your computer, smartphone, or tablet. This ensures a consistent and seamless browsing experience across all your devices, as changes made on one device are automatically reflected on others. Synchronization is particularly useful for those who use multiple devices to access the internet and want to have the same browsing experience everywhere. Managing sync settings allows you to control what data is synced and ensure your information is secure and up-to-date. While sync has many advantages it has also some disadvantages. The following section will mention some advantages and disadvantages.

Advantages of Syncing Across Devices

- **Consistent Experience:** Access the same bookmarks, history, passwords, and settings on all your devices. Start browsing on one device and continue on another without losing your place.
- **Convenience:** Automatically fill in passwords, payment methods, and addresses stored in your Google account. Keep extensions and themes consistent across devices.
- **Backup and Recovery:** Easily recover your data if you lose or replace a device.

Disadvantages of Sync:

- **Security Risks:** If your Google account is compromised, all synced data, including passwords and browsing history, can be accessed by unauthorized individuals. So don't sync on public devices.

- **Data Usage:** Syncing can consume data, which might be a concern for users with limited data plans or slow internet connections.
- **Device Dependency:** If sync settings are not managed properly, changes made on one device can affect your experience on all synced devices, which may not always be desirable.

How to configure Sync in Google Chrome?



Steps: Managing Extensions in Google Chrome

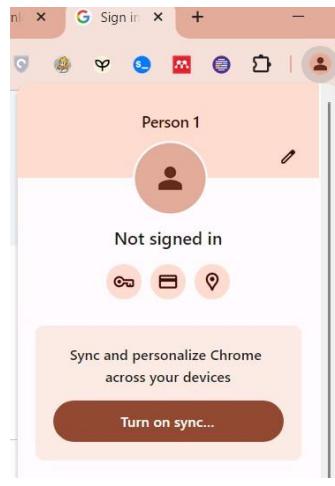
1. Sign in & turn on sync

To sign in to Chrome and turn on sync, you must have a **Google Account**.

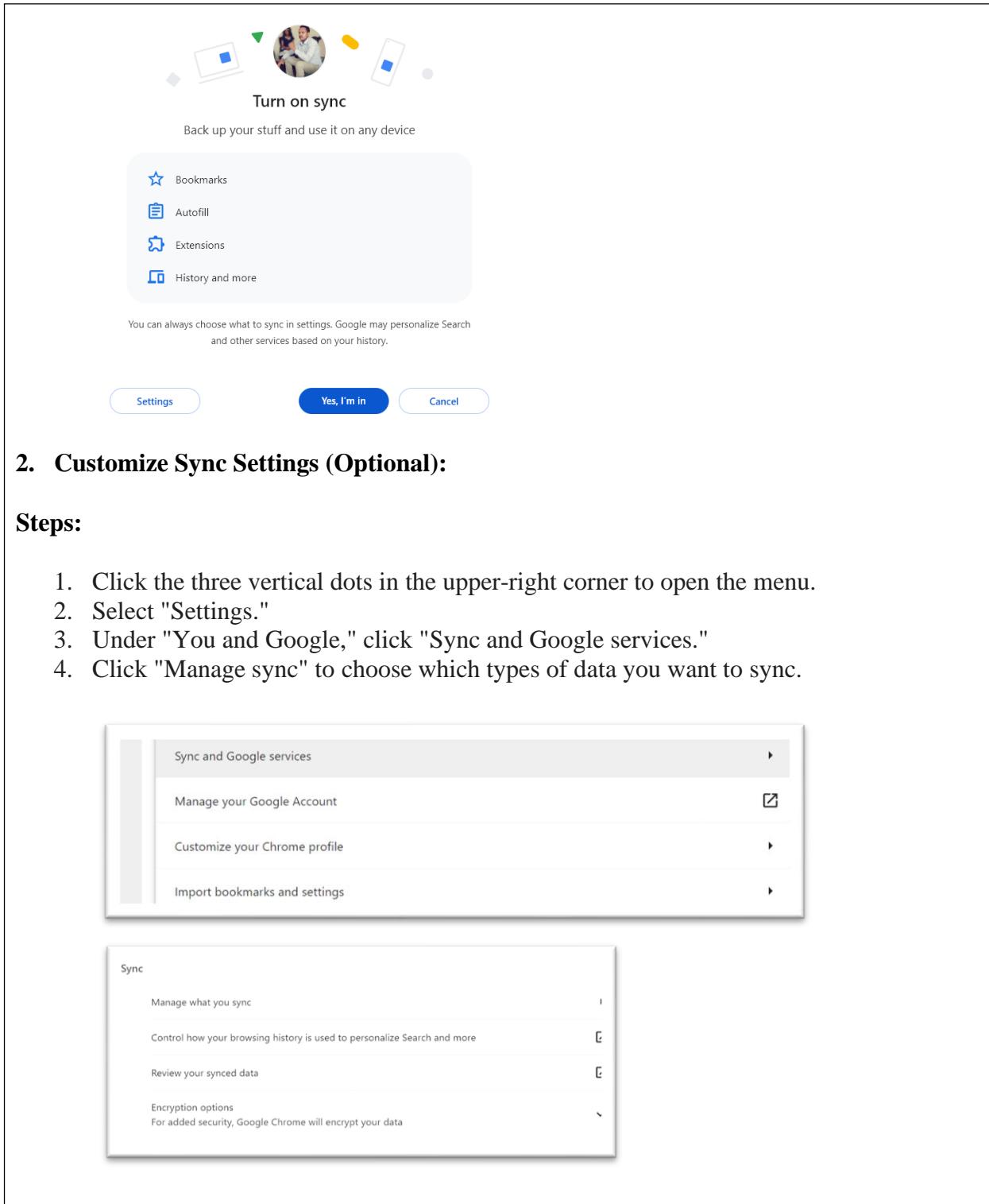
Important: Only turn on Chrome sync with devices that you own. If you use a public computer, use guest mode instead.

Steps:

1. On your computer, open Chrome.
2. At the top right, click Profile .



3. Click **Turn on sync....**
 - If you're not signed into your Google Account, you'll be prompted to sign in.
4. Click **Yes, I'm in.**



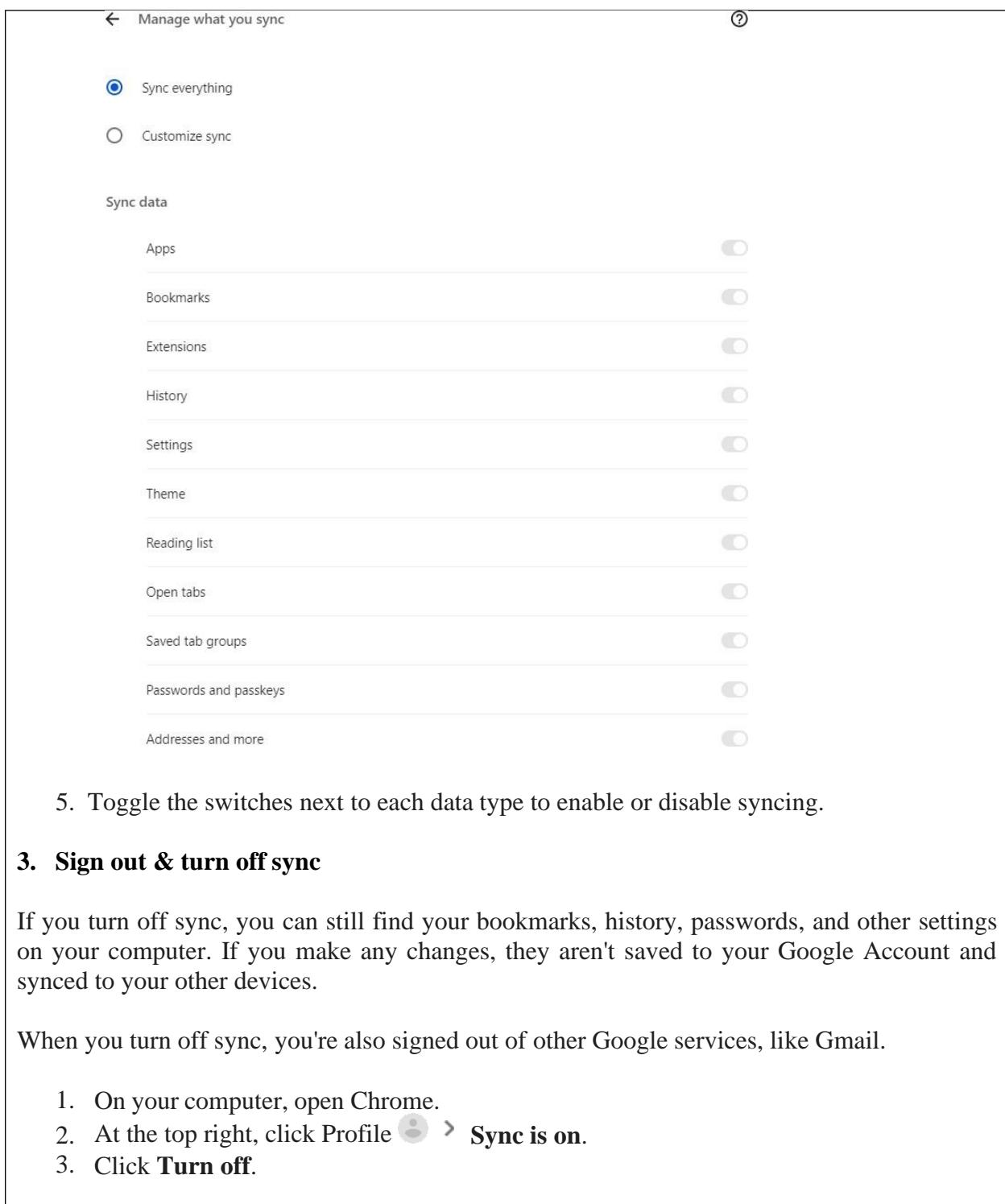
The screenshot shows the "Turn on sync" setup page. At the top, there are icons of a laptop, a smartphone, and a tablet, with a circular profile picture of two people in the center. Below this is the heading "Turn on sync" and the sub-instruction "Back up your stuff and use it on any device". A list of items to sync includes "Bookmarks", "Autofill", "Extensions", and "History and more". A note states, "You can always choose what to sync in settings. Google may personalize Search and other services based on your history." At the bottom are three buttons: "Settings" (light blue), "Yes, I'm in" (dark blue), and "Cancel" (light blue).

2. Customize Sync Settings (Optional):

Steps:

1. Click the three vertical dots in the upper-right corner to open the menu.
2. Select "Settings."
3. Under "You and Google," click "Sync and Google services."
4. Click "Manage sync" to choose which types of data you want to sync.

The second part of the screenshot shows the "Sync and Google services" settings menu. It lists "Sync and Google services", "Manage your Google Account" (with a checked checkbox), "Customize your Chrome profile", and "Import bookmarks and settings". Below this is a "Sync" settings panel with sections for "Manage what you sync", "Control how your browsing history is used to personalize Search and more", "Review your synced data", and "Encryption options" (with a note about encrypting data).



The screenshot shows the 'Manage what you sync' settings page in Google Chrome. At the top, there are two radio button options: 'Sync everything' (selected) and 'Customize sync'. Below this is a section titled 'Sync data' with a list of items, each preceded by a toggle switch:

- Apps (switch off)
- Bookmarks (switch off)
- Extensions (switch off)
- History (switch off)
- Settings (switch off)
- Theme (switch off)
- Reading list (switch off)
- Open tabs (switch off)
- Saved tab groups (switch off)
- Passwords and passkeys (switch off)
- Addresses and more (switch off)

Below the list, step 5 of a process is described: "5. Toggle the switches next to each data type to enable or disable syncing."

3. Sign out & turn off sync

If you turn off sync, you can still find your bookmarks, history, passwords, and other settings on your computer. If you make any changes, they aren't saved to your Google Account and synced to your other devices.

When you turn off sync, you're also signed out of other Google services, like Gmail.

1. On your computer, open Chrome.
2. At the top right, click Profile  **Sync is on.**
3. Click **Turn off**.

**Self-check 2-3:**

Dear Students! So far, We have discussed how to Utilizing and Managing Web Browser Features and how to configure different Browser settings. Now it is your turn to answer the following self-check questions

1. What are browser extensions and how do they enhance the functionality of a web browser?
 2. Explain the purpose of themes and appearance settings in a web browser.
 3. What are the advantages and disadvantages of syncing browsing data across devices?
-

Dear Students! Have you answered the self-check questions? If yes, let us move on and summarize this unit.

Operation Sheet 2-1:**Mozilla Firefox Browser Operations**

Objective: This operation sheet provides step-by-step instructions on how to perform various essential tasks in the Mozilla Firefox browser, including tabs and windows management, customizing tab and home page settings, creating bookmark folders, bookmarking favorite pages, opening bookmarked pages, managing browser history, and downloading and saving files to a location.

1. Tabs and Windows Management:

- To open a new tab: Click on the "+" icon next to the existing tabs or press Ctrl + T (Cmd + T on Mac).
- To close a tab: Click on the "X" icon on the tab, or right-click on the tab and select "Close Tab."
- To switch between tabs: Click on the desired tab.
- To open a new window: Go to the Firefox menu (three horizontal lines) > New Window, or press Ctrl + N (Cmd + N on Mac).

2. Settings Up New Tab Page and Home Page:

- To customize the new tab page: Go to Firefox menu > Preferences (Options on Windows) > Home. Choose "Firefox Home" or a custom URL for new tabs.
- To set the home page: Go to Firefox menu > Preferences (Options on Windows) > Home. Enter the desired URL in the "Homepage and new windows" field.

3. Create Bookmark Folder:

- To create a bookmark folder: Click on the bookmark icon (star icon) in the address bar > Choose "Bookmark This Page" > Select "More" > Choose "New Folder" > Name the folder > Click "Done."

4. Bookmark Favorite Pages:

- To bookmark a page: Click on the bookmark icon (star icon) in the address bar > Choose "Bookmark This Page" > Select the desired folder or create a new one > Click "Done."

5. Open Bookmarked Pages:

- To open a bookmarked page: Click on the bookmark icon (bookmarks toolbar or menu) > Select the desired bookmarked page from the list.

6. Managing Browser History:

- To view browsing history: Go to History menu > Show All History, or press Ctrl + H (Cmd + Shift + H on Mac).
- To clear browsing history: Go to History menu > Clear Recent History, or press Ctrl + Shift + Del (Cmd + Shift + Del on Mac).

7. Download, Save Files to a Location:

- To download a file: Click on a download link > Choose the download location > Click "Save."
- To access downloaded files: Go to the Firefox menu > Downloads, or press Ctrl + J (Cmd + J on Mac).

Operation Sheet 2-2:**Group Activity Operation Sheet****Title: Comprehensive Google Chrome Management Skills**

Objective: To equip students with practical skills in managing tabs, bookmarks, stored data, and settings in Google Chrome to enhance their browsing experience and productivity.

Instructions for Students:

- Form groups of 3-4 students.
- Complete each activity step-by-step.
- Discuss and help each other if anyone faces difficulties.
- Record your findings and experiences for each exercise.
- Present your experiences and solutions to the class at the end of the session.

Activity 1: Tab Management

Objective: Practice opening, closing, pinning, and organizing tabs.

1. Open and Close Tabs

- Open five new tabs and navigate to different websites.
- Close each tab and then reopen them using the history or shortcut keys.

2. Pin and Unpin Tabs

- Pin two tabs.
- Move the pinned tabs around.
- Unpin the tabs.

3. Create and Manage Tab Groups

- Create a tab group with three related tabs.
- Add and remove tabs from the group.

4. Mute and Unmute Tabs

- Open a tab with a video playing and mute it.
- Unmute the tab.

5. Move Tabs to a New Window

- Move a tab to a new window.
- Bring the tab back to the original window.

Activity 2: Bookmark Management

Objective: Learn how to add, organize, and manage bookmarks.

1. Add Bookmarks

- Add bookmarks for five of your favorite websites.
- 2. Create and Organize Folders**
 - Create a folder named "School."
 - Organize your bookmarks into the "School" folder.
- 3. Edit and Delete Bookmarks**
 - Edit the name of one of your bookmarks to something more descriptive.
 - Delete a bookmark you no longer need.
- 4. Import and Export Bookmarks**
 - Import bookmarks from another browser.
 - Export your bookmarks and save the file on your computer.

Activity 3: Managing Stored Data

Objective: Manage browser history, saved passwords, and autofill form data.

- 1. Access and Find Browser History**
 - Access your browser history.
 - Find a page you visited yesterday.
- 2. Manage Saved Passwords**
 - Save a new password for a website.
 - Edit the saved password.
- 3. Manage Autofill Data**
 - Add a new address to the autofill form data.
 - Delete the added address.
- 4. Clear Browsing Data**
 - Clear your browsing history for the last hour.
 - Clear all your browsing data, including cookies and cached images, for the past week.

Activity 4: Customize Chrome Settings

Objective: Customize Chrome's language, homepage, appearance, search engine, extensions, and sync settings.

- 1. Change Language Settings**
 - Change the language setting in Chrome to a different language.
 - Revert it back to the original language.
- 2. Set Homepage and Startup Page**
 - Set a specific website as your homepage.
 - Verify that the homepage opens on startup.
- 3. Customize Themes and Appearance**
 - Apply a new theme.
 - Customize the appearance settings.
- 4. Set Default Search Engine**
 - Change the default search engine.
 - Perform a search to verify the change.

5. Manage Extensions

- Add a new extension.
- Configure the extension's settings.
- Remove the extension.

6. Configure Sync

- Enable sync in Chrome.
- Customize sync settings to include only bookmarks and passwords.

If you completed the above activities, you will become proficient in using various features and settings in Google Chrome, which will enhance your web browsing experience and productivity.

Unit Summary

In Unit 2 of your textbook, you learned about web browsers – the tools that allow us to explore the internet. You discovered that a web browser is like a gateway to the internet. It's a software that helps us access and interact with websites, search for information, watch videos, and much more. You explored how they have evolved from basic text-based browsers to the modern, feature-rich ones we use today, like Chrome and Firefox.

You got to know some of the most commonly used web browsers, such as Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, and Opera. Each has its own unique features and advantages. And You learned that effective browsing involves managing information efficiently. This means being able to find what you need quickly, staying organized, and keeping your information secure. You discovered the key elements that make up web browsing, like tabs and windows management, bookmarking favorite pages, and managing browser history. These tools help us navigate the internet smoothly.

Finally, you explored how to customize your web browser to suit your preferences. This included adjusting settings for language, appearance, privacy, security, cookies, pop-ups, and more. By the end of this unit, you gained a deeper understanding of web browsers and how to make the most out of them. You're now better equipped to explore the vast world of the internet with confidence!

Unit Review Questions

Part I: Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What is the primary function of a web browser?
 - A) Play multimedia files
 - B) Access and interact with content on the World Wide Web
 - C) Send emails
 - D) Edit documents

2. Which of the following browsers is known for its strong privacy features and customizable add-ons?
 - A) Google Chrome
 - B) Mozilla Firefox
 - C) Microsoft Edge
 - D) Apple Safari

3. What is the purpose of managing browser history?
 - A) Enhance browser appearance
 - B) Improve navigation efficiency
 - C) Secure sensitive data
 - D) Block pop-up ads

4. What are cookies in the context of web browsing?
 - A) Small text files used to store website data on a user's device
 - B) Virtual assistants
 - C) Pop-up notifications
 - D) Browser extensions

5. Which feature allows users to link their browser settings, bookmarks, and history across multiple devices?
 - A) Cookie management
 - B) Pop-up blockers
 - C) Tabbed browsing
 - D) Synchronization

Part II: Short-Answer Questions:

Instruction: Give brief answers to the following review questions:

1. Explain the importance of effective information management in web browsing.
2. Describe the evolution of web browsers from text-based to modern graphical browsers.
3. Name three popular web browsers and briefly explain one unique feature of each.
4. What are bookmarks, and how do they enhance browsing efficiency?
5. Define browser history and discuss its significance in web browsing.
6. How can users configure privacy and security settings in a web browser?
7. What are cookies, and what role do they play in web browsing?
8. Briefly explain the purpose of pop-up and redirect settings in a web browser.
9. Describe the function of content settings related to images, JavaScript, and plugins.
10. How can users manage extensions and add-ons in their web browsers?

Answer key for Self-check questions

Self-check 2-1: Answers

1. A web browser is a software application that allows users to access and interact with content on the World Wide Web (WWW).
2. **Google Chrome:** Known for its speed and integration with Google services.
Mozilla Firefox: Celebrated for its strong privacy features and customizable add-ons.
Safari: Recognized for its speed and energy efficiency, especially on Apple devices.
3. Web browsers interpret and render web content, acting as intermediaries between users and web servers, enabling web developers to test and view their web applications and sites.

Self-check 2-2: Answers

1. Browser tabs allow you to open multiple web pages within a single browser window, making it easier to switch between different websites and manage multiple tasks simultaneously.
2. The browser's address bar is used to enter URLs or search terms to navigate to specific websites or perform web searches.
3. Browser history keeps a record of all the web pages a user has visited, allowing users to revisit previously viewed pages and providing insights into their browsing activity.
4. Bookmarks allow users to save their favorite or frequently visited web pages for quick and easy access later.

Self-check 2-3: Answers

1. Browser extensions are small software programs that add new features, modify existing functionalities, or provide tools to improve the overall browsing experience. Examples include ad blockers, password managers, language translators, and productivity tools.

2. Themes and appearance settings allow users to personalize the look and feel of their browser, making it more visually appealing and tailored to their preferences. This can enhance the overall browsing experience and improve productivity.
3. **Advantages:** Provides a consistent experience across all devices, allows easy access to bookmarks, history, passwords, and settings, and offers convenience in managing information.

Disadvantages: Can pose security risks if your Google account is compromised, consumes data, and changes made on one device can affect all synced devices, which may not always be desirable.

UNIT 3

SEARCH AND EVALUATE INFORMATION OVER THE INTERNET

Unit Coverage

This unit is designed to provide you with the necessary information and practice regarding the following content coverage:

- 3.1. Search information
- 3.2. Critical Evaluation
- 3.3. Copyright, Data Protection

Unit Learning Outcomes

This unit will also assist you in attaining the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Search effectively and critically to evaluate information over the web .

Unit Overview

In the previous unit, you used the browser and surfed the web by configuring different settings as per your needs. Now it is time to use search engines and search for and evaluate information. This comprehensive unit aims to empower students with the skills needed to navigate the vast landscape of online information, ensuring they can search effectively, critically evaluate sources, and navigate legal and ethical considerations in the digital realm.

Key Terms: *Search Engines, Search Strategies, advanced search features, CRAAP, Copyright*

3.1. Search Information

Discussion

Dear Students! you have explored various browsers and their usage; can you think about how you find specific information using these browsers?

Form small groups of 3-4 students. Discuss the techniques you use to search for information online. Consider the following points:

- What are search engines and can mention some examples?
- What techniques do you use to search for information online?
- How do you evaluate the credibility of the information you find?

"Search for Information" refers to the process of actively seeking and retrieving specific data, facts, or knowledge from various sources, often utilizing search engines or databases. This can be done for a variety of purposes, such as academic research, professional projects, or personal curiosity. The key steps in a search for information typically include formulating a clear query or question, using appropriate search tools, evaluating search results, and extracting relevant information to meet the intended purpose.

Navigating the sea of information requires more than just typing keywords. In the context of web browsing, "Search for Information" often involves using search engines like Google, Bing, or Yahoo to locate relevant web pages, articles, or resources related to a particular topic. Users input keywords or phrases into the search bar, and the search engine returns a list of results ranked by relevance. Effective searching requires refining queries, understanding search operators, and critically evaluating the credibility and reliability of the information obtained. It is a fundamental skill in the digital age for accessing and utilizing the vast amount of information available on the internet. In this section, you will delve into the fundamentals of accessing information on the internet.

3.1.1 Search Engines and Their Functions

A search engine is a software system embedded in a website that is used to search for information on the World Wide Web. They are powerful tools that enable users to find information on the

Internet quickly and efficiently. They index millions of web pages and provide results based on the relevance to the user's query. Understanding how search engines work and their functions is essential for navigating the vast amount of information available online. The search results are generally presented in a line of results, often referred to as search engine results pages (SERPs). The information may be a mix of web pages, images, videos, infographics, articles, research papers, and other types of files. The information may be a mix of web pages, images, videos, infographics, articles, research papers, and other types of files.

Popular Search Engines

Several search engines are widely used around the world, each with unique features:

Google: The most widely used search engine, known for its speed, relevance, and comprehensive index. Google also offers various tools like Google Images, Google Maps, and Google Scholar.

- **Bing:** Microsoft's search engine, which integrates well with Windows and Microsoft services. Bing offers unique features like image search, video search, and integration with Microsoft Office.
- **Yahoo:** Provides a variety of services including search, news, email, and more. Yahoo search is powered by Bing.
- **DuckDuckGo:** Focuses on user privacy and does not track search activity, making it a popular choice for those concerned with privacy.



Figure 3.1: Examples of search engines

Key Functions of Search Engines

- **Keyword Search:** Users type in words or phrases (keywords) to find relevant web pages.
- **Advanced Search:** Offers options to refine search queries with filters for date, language, region, and more.
- **Image and Video Search:** Allows users to find images and videos related to their queries.
- **Voice Search:** Enables users to perform searches using voice commands, which is especially useful on mobile devices.
- **Local Search:** Helps users find businesses, services, and points of interest in their local area.

The Role of Search Engines in Web Development

For web developers, understanding how search engines work is crucial for optimizing websites to achieve higher rankings in search results. This process, known as Search Engine Optimization (SEO), involves:

- **Keyword Research:** Identifying the keywords that potential visitors use to find related content.
- **On-Page SEO:** Optimizing the content and structure of the website to be search-engine friendly. This includes using proper HTML tags, meta descriptions, and alt text for images.
- **Off-Page SEO:** Building backlinks from other reputable sites to improve the website's authority and ranking.
- **Technical SEO:** Ensuring that the website is accessible to crawlers, loads quickly, and is mobile-friendly.

By leveraging these techniques, web developers can enhance their site's visibility, attract more visitors, and achieve their online goals.



TIPS

Many people think that when you use a search engine such as Google, it will search the entire Web and find information that you require. However, a search engine like Google will only search through a list of maintained sites that have been registered with that particular search engine. This in part explains the differing results you sometimes get when you search using different search engines. Also, each search engine has different criteria for ranking search results.

Search Strategies and techniques

Dear Students! Dear Students! you have explored search engines and understood their functions. What techniques might help you find exactly what you're looking for? Now that Let's delve into search strategies and techniques to enhance your ability to find precise and valuable information efficiently.

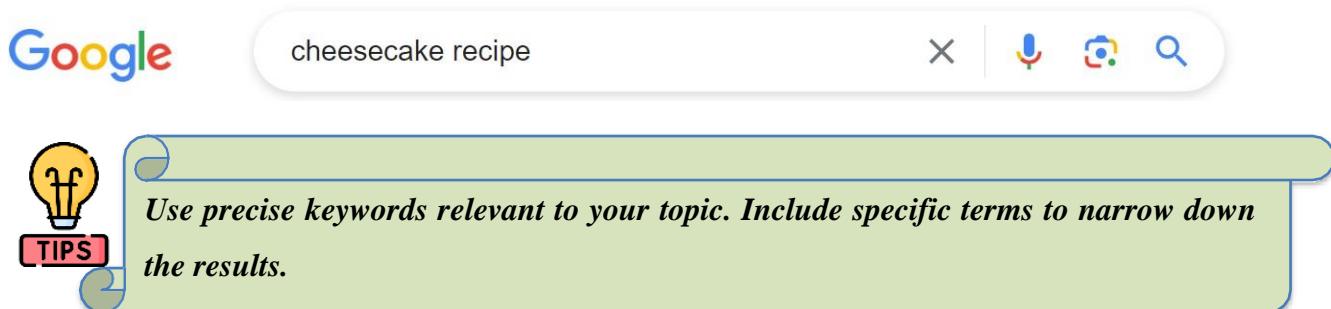
Search strategy is the action plan which is drawn to conduct a search. It encompasses several steps and levels of work in information retrieval. There are many issues that need to be considered while formulating an appropriate search statement. These are:

- i. the concepts or facets to be searched and their order;
- ii. the term(s) that appropriately represent(s) the search concept;
- iii. the feature(s) of the retrieval system concerned; and
- iv. the measures to be taken in revising a search statement.

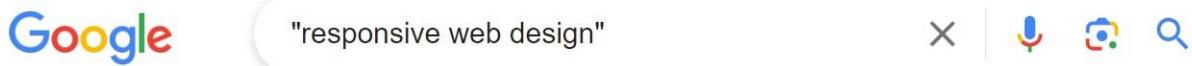
Developing a good search strategy requires knowledge about the nature and organization of target database(s) and also the exact needs of the user. Knowledge of the user's exact requirement can greatly affect the actual search and retrieval process. In some cases, the user may want only a few relevant items on a given topic, in which case the task of searching will obviously be limited. Conversely, the user may wish to obtain all the relevant items (obviously with as small a number of non-relevant items as possible), in which case the search must be exhaustive. The search results that a given search will generate is largely dependent on the search word or phrase used. Different techniques are available to conduct searches effectively so that maximum relevant information are retrieved according to users need. The more exact you can be in your choice of word(s) the more

accurate the search results. Google, as the world's leading search engine, offers a range of searching strategies and techniques to help users find relevant information. For web developers and students, mastering these techniques can significantly enhance research capabilities and productivity. Here are some common strategies and techniques:

1. **Basic Search:** A search can be conducted by entering a single search term or a phrase comprising more than one term. Keyword search is the simplest form of search, facility offered by a search system. Basic searches involve entering a few keywords or phrases related to the topic you're interested in. Google then returns results that include web pages containing those keywords. For example, you can search for "Cheese **cake recipe**" to find recipes for making cheese cake.



2. **Exact Match Search:** An exact match search is a technique used to find results that contain the exact phrase or terms specified in the search query. By using quotation marks around a phrase or set of words, Google will return only those results that contain the precise sequence of words in the same order as typed. This method is particularly useful when looking for specific phrases, names, quotes, or any exact string of text. For example, if you are searching for information on "responsive web design," you would type:



Then, Google will return results that contain the exact phrase "responsive web design" in that order, ensuring more relevant and precise matches.

The screenshot shows a Google search results page with the query "responsive web design" entered in the search bar. The first result is a link to Wikipedia, titled "Responsive web design". Below it is a snippet of text from the Wikipedia page: "Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or ...". The second result is a link to W3Schools, titled "Responsive Web Design Introduction". Below it is a snippet of text from the W3Schools page: "RWD Intro RWD Viewport RWD Grid View RWD Media Queries RWD Images RWD Videos RWD Frameworks RWD Templates ... Responsive web design makes your web page look...". The third result is a link to MDN Web Docs, titled "Responsive design - Learn web development | MDN". Below it is a snippet of text from the MDN page: "Apr 29, 2024 — Responsive web design (RWD) is a web design approach to make web pages render well on all screen sizes and resolutions while ensuring good".

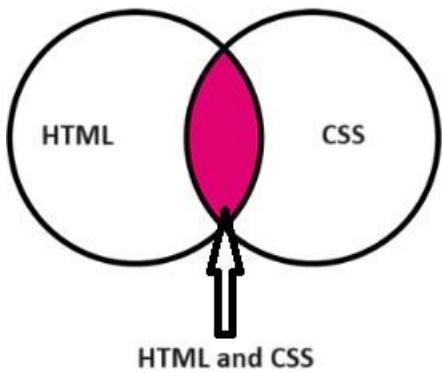
This technique is particularly useful for locating specific phrases, quotes, names, and product information, making your search process more efficient and effective.

3. **Boolean Search:** This is a very common search technique that combines search terms according to the Boolean logic. Three types of Boolean search are possible: AND search, OR search and NOT search.

A. **Boolean AND search:** allows users to combine two or more search terms using the Boolean AND operator. A Boolean AND search will retrieve all those items where all the constituent terms occur. For example, the following search expression "Internet and WWW" will retrieve all those records where both the terms occur. Boolean AND search adds more restrictions to a search expression by adding more search terms. Therefore, the more search terms are ANDed, the more restricted, or specific, will be the search, and as a result the less will be the search output. Sometime, a search may not produce any result if too many search terms are ANDed.

For example, "HTML AND CSS" will show results containing both HTML and CSS.

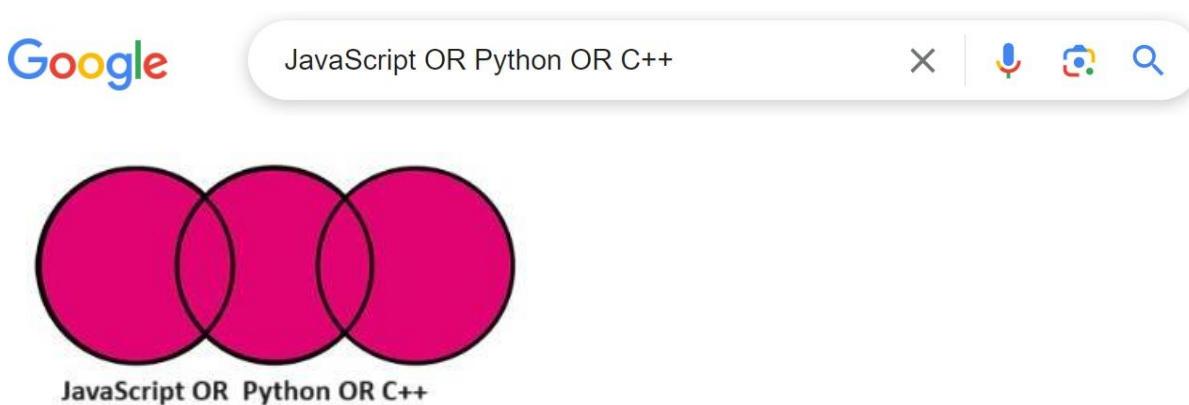
The screenshot shows a Google search results page with the query "HTML AND CSS" entered in the search bar. The search bar also includes a microphone icon for voice search and a camera icon for image search.



- B. Boolean OR search:** allows users to combine two or more search terms such that the system retrieves all those items that contain either one or all of the constituent terms. Thus, the following search expression "Colleges or Universities" will retrieve all those records (1) where the term Colleges occurs, (2) where the term Universities occurs, and (3) where both the terms occur. Note that this is contrary to the use of the term 'or' in normal English. Boolean OR search; though adds more terms to a search expression, adds less restrictions to a given search expression, because the search is conducted for occurrence of each single ORed term irrespective of whether the other term(s) occurs or not. Consequently, the output of OR

searches will be more. When too many search terms are ORed, the search output may be too big to handle.

For example, "JavaScript OR Python OR C++" will show results containing either JavaScript or Python or C++.



C. Boolean NOT: Excludes results containing the specified term. search allows users to specify those terms that they do not want to occur in the retrieved records. For example, the following search expression "Search engines NOT Hotbot" will retrieve all the records on search engines except those where the term 'Hotbot' occurs. Boolean NOT searches add restrictions to a search by forcing the search system to discard those items where the NOT term(s) occur. Hence the search output will decrease with increase in the NOT terms.

For example, "JavaScript NOT Java" will show results about JavaScript but exclude those about Java.



4. Site Search: To search within a specific website, use the "site:" operator.

Example: Searching for "CSS Grid site:w3schools.com" will return results about CSSGrid specifically from w3schools.com

Google search results for "CSS Grid site:w3schools.com". The search bar shows the query. Below it, the "All" tab is selected. The first result is from w3schools.com, titled "CSS Grid Layout Module". The snippet describes the CSS Grid Layout Module as a grid-based layout system. Below the snippet are links to "Display: grid", "CSS Grid Container", "Grid elements", and "Try it Yourself".

5. Wildcards (*) search: Using an asterisk (*) as a placeholder for any word or part of a word.

Google allows the use of asterisks (*) as placeholders for missing words.

For example, searching for "web * design" will return results for web development design, web graphic design, web responsive design, etc.

Google search results for "web * design". The search bar shows the query. Below it, the "All" tab is selected. The first result is from Designhill, titled "Modern Website Designs - Best Web Page Designers". The snippet describes Designhill's services. Below the snippet is a rating of "★ ★ ★ ★ ★ Rating: 4.7 · 40,980 votes". The second result is from Rock Content, titled "What is web design, how to do it right and best skills". The snippet describes web design as applied to a website's visuals. It involves colors, fonts, and layout.

6. File Type Search: Using the filetype: operator to search for specific types of files.

Example: Searching for "HTML5 cheat sheet filetype:pdf" will return PDF files related to HTML5 cheat sheets.

Google search results for "HTML5 cheat sheet filetype:pdf".

HTML Cheat Sheet
Stanford University
<https://web.stanford.edu/group/csp/htmlche...> [PDF] ::

HTML 5 Cheat Sheet (PDF)
johndecember.com
<https://johndecember.com/html/spec/HTML...> [PDF] ::

HTML Cheat Sheet
Stanford University
<https://web.stanford.edu/group/csp/htmlche...> [PDF] ::

HTML 5 Cheat Sheet (PDF)
johndecember.com
<https://johndecember.com/html/spec/HTML...> [PDF] ::

9. Definitions: To get a definition of a word or term, simply type "define:" followed by the word. For

Example, "*define:onomatopoeia*" will provide the definition of that word.

Google search results for "define:onomatopoeia".

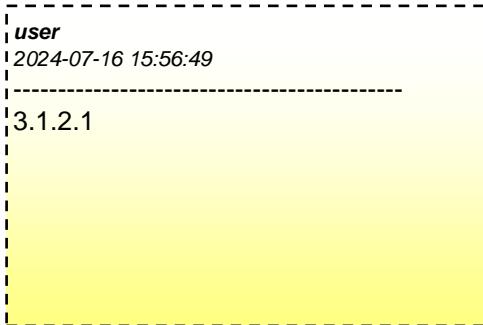
Dictionary
Definitions from [Oxford Languages](#) · Learn more

onomatopoeia
/ ɒnəmətə'pi:ə /

noun

the formation of a word from a sound associated with what is named (e.g. cuckoo, sizzle).
"a relatively large number of bird names arise by onomatopoeia"

- the use of onomatopoeia for literary effect.
"the language in her picture books is notable for its playfulness and onomatopoeia"



3.1.1.1. Refine a search using advanced search features

Google's Advanced Search Options provide a powerful way to refine and customize your searches, enabling you to find more specific and relevant information. These options allow you to narrow down your search results based on various criteria. When you have carried out a search using the Google search engine, it is possible to refine your results using advanced search features like: Date, Media types, Language, Usage rights. Here's a description of the key Advanced Search Options:

- **Exact Words:** Find results that include an exact phrase enclosed in quotation marks. This ensures that search results match the phrase precisely.
- **Exclude Words:** Exclude specific terms from your search to filter out irrelevant results.
- **Site or Domain:** Search for content within a specific website or domain by using the "Site or domain" field. This is useful when you want to find information from a particular source.
- **File Type:** Specify a file type, such as PDF, to search for documents in that format. This is helpful when looking for specific types of files.
- **Usage Rights:** Filter results based on usage rights, which helps you find content that can be reused, modified, or shared commercially. This is useful for content creators or educators.
- **Region:** Narrow your search results to a specific region or country to find information relevant to a particular location.
- **Last Update:** Choose a time frame (e.g., past day, past week, past month) to find content that has been updated recently.
- **SafeSearch:** Activate SafeSearch to filter out explicit or adult content from search results, making your search experience more family-friendly.
- **Language:** Specify the language in which you want your search results to appear

These features can help pinpoint a closer match to what you are searching for. For example:

- you may only want to find information from the last month, so you may refine your search by date.
- Or you may only want to find videos or images that have a creative commons license. In this case you may refine your search by media type and usage rights.
- Or you may want to find results only in a specific language. In this case you may refine your search by language. By default, results are returned in the default language set for Google Chrome.

Operation Sheet 3-1:



Exploring Advanced Search Options

Objective: To familiarize students with Google's Advanced Search options and encourage them to use these features to refine and customize their online searches.

Instructions:

- 1 Open web browser and go to Google (www.google.com).
- 2 Start with a basic keyword search. Choose a topic you're interested in, such as "climate change."
- 3 Click on "Settings" that is to the right side of the search bar.

- 4 Click on "Advanced Search." On the popup window.

Advanced search window will be displayed.

Advanced Search

Find pages with...

all these words: To do this in the search box.
Type the important words: tri-colour rat terrier

this exact word or phrase: Put exact words in quotes: "rat terrier"

any of these words: Type OR between all the words you want: miniature OR standard

none of these words: Put a minus sign just before words that you don't want: -rodent, -"Jack Russell"

numbers ranging from: to Put two full stops between the numbers and add a unit of measurement: 10..35 Kg, £300..£500, 2010..2011

Then narrow your results by...

language: Find pages in the language that you selected.

region: Find pages published in a particular region.

last update: Find pages updated within the time that you specify.

site or domain: Search one site (like wikipedia.org) or limit your results to a domain like .edu, .org or .gov

terms appearing: Search for terms in the whole page, page title or web address, or links to the page you're looking for.

file type: Find pages in the format that you prefer.

usage rights: Find pages that you are free to use yourself.

Advanced Search

5 Participants should now explore and use the following advanced search options:

- Exact Words:** enter a specific phrase enclosed in quotation marks, such as "global warming effects on wildlife."
- Exclude Words:** exclude a term from their search, such as "renewable energy - solar."
- Site or Domain:** search for information within a specific website or domain using the "Site or domain" field. For example, "site: wikipedia.org greenhouse effect."
- File Type:** Encourage participants to search for a specific file type, like "PDF" or "PPT." For example, "filetype:pdf climate change report."
- Usage Rights:** Show how to filter results based on usage rights, which can be helpful when looking for content that can be reused or modified.
- Region:** Explore the option to specify results based on a particular region or country.
- Last Update:** Use the "Last update" option to find information updated within a specific time frame, such as the past year or month.

3.1.1.2. Reverse Image Search

Google Reverse Image Search is a feature provided by Google that allows users to search for information, related images, and sources using an image as the search query instead of text. This tool is valuable for discovering more about a particular image, identifying its source, finding visually similar images, and exploring web pages where the image appears. Google's Reverse

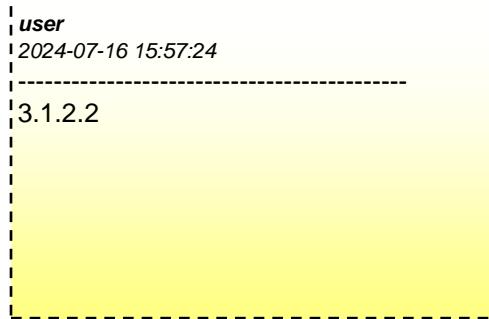


Image Search uses advanced algorithms to analyze the content of the uploaded image and retrieve relevant information.

Key Use Cases for Google Reverse Image Search:

- **Verifying Image Authenticity:** Check if an image has been edited or manipulated by comparing it with the original or other versions online.
- **Identifying Sources and Attribution:** Find the original source of an image to provide proper attribution or credit, especially for creative works.
- **Researching Products or Objects:** Use reverse image search to identify objects, products, landmarks, or artworks within an image.
- **Detecting Fake Profiles or Scams:** Verify the authenticity of profile pictures on social media or dating platforms to identify potential scams or fake accounts.
- **Locating Higher-Resolution Versions:** Find higher-resolution versions of images for various purposes, such as printing or design projects.

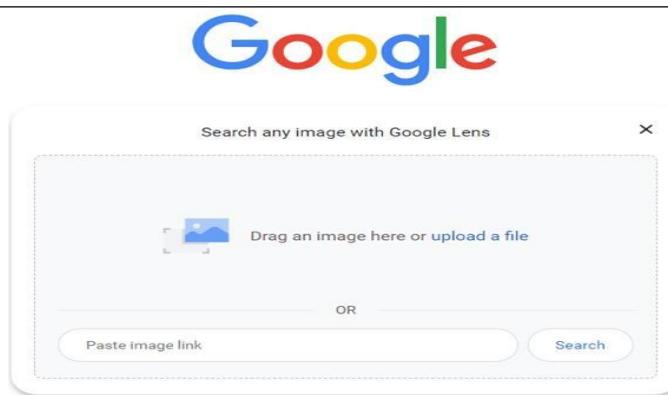
Google Reverse Image Search is a versatile tool that empowers users to delve deeper into the content and context of images found online. It is particularly useful for fact-checking, image verification, and obtaining more information about visual content.

Operation Sheet 3-2:**To carry out a search using an image:**

1. In the search box in the Google Images search engine, select the camera icon.



2. Enter an image URL in the search field, or drag an image into the search field. or alternatively, click the Upload an image tab, click Choose file and browse to an image on your computer or device, select the desired file and click Open.

**Self-check 3-1:**

Dear Students! So far, we have discussed about Search Engines and the different Search Strategies and techniques. Now it is your turn to answer the following self-check questions

1. What is the primary function of a search engine?
2. What is a Boolean search and how does it refine search results?
3. Give an example of how to use the 'site:' operator in a search query.

Dear Students! Have you answered the self-check questions? If yes, let us move on to the next topic.

3.2. Critical Evaluation

Discussion



- In small groups, Reflect on the vast information available online and why it's crucial to evaluate sources

In the digital age, it's essential to critically evaluate online information to ensure its accuracy, reliability, and credibility. When you start searching for suitable resources, you will want to select resources that improve your knowledge and your learning experiences. And you will want to select accurate and reliable content.

However, selecting appropriate resources can sometimes seem overwhelming. There are so many options to choose from and so many decisions to make about their suitability. And knowing when content is accurate and from a reliable source, especially when searching online, isn't always straightforward.

There are many sites, blogs and social media posts that aim to share inaccurate information to mislead people, either to further a political or religious agenda or for financial gain. There are hoax sites and “Fake news” sites and posts that are written as though they are factual and true but they are actually fabricated. There are also sites and posts that use misinformation and inaccuracies to promote radical ideologies or to support a particular point of view. It is important for students to always critically evaluate online content to make sure that it is from a reliable source and can be trusted. In order to select suitable, effective and trustworthy resources, you should always critically evaluate resources using A well-defined set of evaluation criteria.

3.2.1. Source Credibility Assessment

Users must critically evaluate the credibility of online sources. Understand the criteria for assessing reliability, bias, and expertise. Evaluating is about determining and assessing the quality and credibility of the information you find. It encourages you to think critically about the reliability, validity, accuracy, authority, timeliness, point of view or bias of information sources. There are certain frameworks that information professionals have put together to help people think critically about the information provided. One of them and a highly used framework is the CRAAP framework.

The CRAAP framework is a set of criteria designed to help individuals evaluate information sources for their credibility, reliability, and relevance. The acronym "CRAAP" stands for Currency, Relevance, Authority, Accuracy, and Purpose. Let's delve into each component in detail:

Currency:

Definition: Refers to the timeliness of the information, i.e., how recent it is.

Questions to Ask:

- When was the information published or last updated?
- Is the information still current and applicable to the topic?

Relevance:

Definition: Focuses on the importance of the information in relation to your research or information needs.

Questions to Ask:

- Does the information relate to your topic or research question?
- Is it suitable for your academic or informational purposes?

Authority:

Definition: Examines the source's credibility and the qualifications of the author or organization providing the information.

Questions to Ask:

- Who is the author, and what are their credentials or qualifications?
- Is the author affiliated with a reputable institution or organization?

Accuracy:

Definition: Assesses the reliability and truthfulness of the information.

Questions to Ask:

- Are the facts and claims supported by evidence or citations?
- Can the information be verified through other reputable sources?

Purpose:

Definition: Explores the intent behind the information, whether it is to inform, persuade, entertain, or sell a product.

Questions to Ask:

- What is the purpose of the information? Is it to inform, entertain, or persuade?
- Is there potential bias or a conflict of interest?

Applying the CRAAP framework involves considering each of these criteria when evaluating a source. It helps individuals develop a comprehensive understanding of the information's quality

and reliability. Keep in mind that the importance of each criterion may vary depending on the context of your research or information needs. This framework is commonly used in academic settings to guide students and researchers in critically evaluating sources before incorporating them into their work.

Operation Sheet 3-3:



Digital Information Hunt and critical Evaluation

Activity Title: "Digital Information Hunt"

Objective: This activity aims to apply and reinforce the search strategies and techniques learned, enabling you to find relevant digital information effectively. In this task you are also going to evaluate a resource of your choosing to see whether it is appropriate or not for your needs.

Instructions:

- 1 Make small groups (2-3 students per group) or pairs.
- 2 choose a specific topic or research question related to your field of interest, industry, or academic focus. the chosen topic should require gathering information from online sources.
- 3 Students will brainstorm and plan their search strategy. You should consider searching techniques they learned: using quotes, operator, specific filter of Keywords, types of sources they, picture search or advanced search related to their chosen topic.
- 4 Use search engines and start executing your search strategies
- 5 Evaluate the resource to determine its effectiveness in supporting and enhancing your lesson. Use the CRAAP Test Worksheet below.
- 6 Fill out the CRAAP Test Worksheet for their chosen source.
- 7 The evaluation result includes whether you intend to use the resource and why.
- 8 Share your findings with the class.

The CRAAP Test Worksheet

Title of Source: _____

Use the following list to help you evaluate sources. Answer the questions as appropriate, and then rank each of the 5 parts from 1 to 10 (1 = unreliable, 10 = excellent). Add up the scores to give you an idea of whether you should use the resource (and whether your instructor would want you to!).

Currency: the timeliness of the information.....

- When was the information published or posted? _____
- Has the information been revised or updated? _____
- Is the information current or out-of-date for your topic? _____
- Are the links functional? _____

Relevance: the importance of the information for your needs.....

- Does the information relate to your topic or answer your question? _____
- Who is the intended audience? _____
- Is the information at an appropriate level? _____
- Have you looked at a variety of sources before choosing this one? _____

Would you be comfortable using this source for a research paper? _____

Authority: the source of the information.....

- Who is the author/publisher/source/sponsor? _____
- Are the author's credentials or organizational affiliations given? _____
- What are the author's credentials/organizational affiliations if given? _____
- What are the author's qualifications to write on the topic? _____
- Is there contact information, such as a publisher or e-mail address? _____

Does the URL reveal anything about the author or source? _____

Accuracy: the reliability, truthfulness, and correctness of the content.....

- Where does the information come from? _____
- Is the information supported by evidence? _____
- Has the information been reviewed or refereed? _____
- Can you verify any of the information in another source? _____
- Does the language or tone seem biased and free of emotion? _____

Are there spelling, grammar, or other typographical errors? _____

Purpose: the reason the information exists.....

- What is the purpose of the information? _____
- Do the authors/sponsors make their intentions or purpose clear? _____
- Is the information fact? opinion? propaganda? _____
- Does the point of view appear objective and impartial? _____
- Are there political, ideological, cultural, religious, institutional, or personal biases? _____

45 - 50 Excellent | 40 - 44 Good
35 - 39 Average | 30 - 34 Borderline Acceptable
Below 30 - Unacceptable

Total:



Self-check 3-2:

Dear Students! In the above section we have discussed information evaluation strategies. Now it is your turn to answer the following self-check questions

1. What is the primary function of a search engine?
2. What is a Boolean search and how does it refine search results?
3. Give an example of how to use the 'site:' operator in a search query.

Dear Students! Have you answered the self-check questions? If yes, let us move on to the next topic.

3.3. Copyright and Data Protection

Dear Students! We have just explored how to critically evaluate information to ensure its accuracy and reliability. Can you think of why it is important to not only evaluate but also understand the legal and ethical aspects of using information online?

Now Let's delve into copyright and data protection, crucial topics that help us respect intellectual property and safeguard personal information in our digital interactions.

There are a vast number of digital resources online that you can use. However, there are rules and responsibilities when using online resources. These rules apply to anyone using online resources including students, teachers, researchers, leaders, and managers so ensure that they know their responsibilities as well. In the digital age, understanding copyright and data protection is crucial, especially for web developers. These concepts not only protect creators and their work but also safeguard personal data and privacy. As web developers, it's essential to be aware of these laws and regulations to create ethical and secure websites.

3.3.1. Copyright Laws in the Digital World

Copyright laws are designed to protect the rights of creators by giving them control over how their work is used. In the digital world, these laws apply to various forms of media, including text, images, music, and videos. For web developers, respecting copyright is vital to avoid legal issues and to maintain the integrity of their work.

Copyright grants the creator of original work exclusive rights to its use and distribution. This applies to literary, dramatic, musical, and artistic works, as well as sound recordings, films, and

broadcasts. In the context of web development, this means that any content you use on a website, from images and music to code snippets and text, must either be original, properly licensed, or used with permission.

The digital world makes it easy to share and copy content, which can lead to unintentional copyright infringement. For example, using a photo found on the internet without permission or proper attribution can lead to legal consequences. Web developers must ensure that all assets used in a website are either created by them, licensed for their use, or available under a Creative Commons license. Additionally, understanding the duration of copyright, typically the creator's lifetime plus 70 years, helps in knowing when works enter the public domain and can be used freely.

Infringement of copyright can result in significant penalties, including fines and legal actions. To avoid these issues, always obtain the necessary permissions or licenses for the content you use. For instance, if you're using a piece of music on your website, ensure it is either royalty-free or you have the appropriate license. Using tools like Google's "Usage Rights" search filter or websites that offer free-to-use or licensed content can help in finding legally usable materials.

3.3.2. Data Protection rights and obligations

Data protection laws are designed to safeguard individuals' personal information from misuse and unauthorized access. These laws outline the rights of individuals regarding their data and the obligations of organizations that collect and process this data. For web developers, understanding and implementing these laws is crucial to build trust with users and ensure compliance with legal standards.

Personal data includes any information that can identify an individual, such as names, addresses, email addresses, and social security numbers. Data protection laws like the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States give individuals control over their personal data. These laws grant rights such as access to personal data, the right to rectify incorrect data, the right to have data erased, and the right to data portability.

For web developers, this means implementing features that allow users to exercise these rights. For instance, providing a user-friendly interface where users can update their personal information

or request the deletion of their account is essential. Moreover, developers need to implement robust security measures to protect personal data from breaches. This includes using encryption, secure protocols like HTTPS, and regularly updating software to fix vulnerabilities.

Data controllers, typically the organizations that own the websites, have specific obligations under data protection laws. They must ensure that personal data is collected for legitimate purposes, kept secure, and not retained longer than necessary. For example, a web developer working on an e-commerce site must ensure that customers' payment details are securely processed and stored, complying with PCI DSS standards.

Regular data protection audits and impact assessments can help in identifying potential risks and ensuring compliance. Training for all team members on data protection principles and best practices is also crucial. Informing users about how their data is collected, used, and protected through clear privacy policies helps build trust and transparency.

In summary, understanding and adhering to copyright and data protection laws is essential for web developers. These principles not only protect creators and individuals but also promote ethical practices and trust in the digital world. By respecting copyright and safeguarding personal data, web developers can create secure, compliant, and trustworthy websites.



Self-check 3-1:

Dear Students! So far, We have discussed about Copyright and Data Protection. Now it is your turn to answer the following self-check questions

1. What is copyright and why is it important in the digital world?
 2. What are some key aspects of data protection laws like GDPR and CCPA?
 3. How can web developers ensure compliance with data protection laws?
-

Dear Students! Have you answered the self-check questions? If yes, let us move on and summarize this unit.

Unit Summary

In Unit Three, you delved into the essential skills for searching and evaluating information over the internet. You learned how to effectively use search engines like Google, Bing, Yahoo, and DuckDuckGo. We covered various search techniques, including basic keyword searches, exact match searches using quotation marks, Boolean searches with operators like AND, OR, and NOT, as well as site-specific and file type searches. These strategies help you locate precise and relevant information quickly and efficiently.

You also explored critical evaluation techniques to assess the credibility and reliability of online sources. Using the CRAAP framework—Currency, Relevance, Authority, Accuracy, and Purpose—you are now equipped to scrutinize the quality of information you find. This is crucial for discerning trustworthy sources and avoiding misinformation or fake news, ensuring that the information you use and share is accurate and reliable.

Finally, the unit emphasized the importance of understanding copyright and data protection laws. You learned how copyright protects creators' rights and the legal implications of using copyrighted material without permission. Additionally, we discussed data protection laws like GDPR and CCPA, highlighting the responsibilities of web developers to safeguard personal information and implement secure data handling practices. By mastering these concepts, you can create ethical, secure, and legally compliant websites.

Unit Review Questions

Part I: Multiple Choice Questions

1. What is the main function of a search engine?
 - A) To store web pages
 - B) To index, crawl, and retrieve information from web pages
 - C) To create websites
 - D) To design web pages

2. Which Boolean operator would you use to exclude a term from your search results?
 - A) AND
 - B) OR
 - C) NOT
 - D) NEAR
3. What is the purpose of using quotation marks in a search query?
 - A) To search for any word in the phrase
 - B) To find the exact phrase as typed
 - C) To exclude certain words
 - D) To find synonyms of the phrase
4. What does the 'site:' operator do in a search query?
 - A) Limits search results to a specific website
 - B) Finds similar websites
 - C) Displays images only
 - D) Excludes certain websites from the search
5. What is the CRAAP framework used for?
 - A) Designing websites
 - B) Evaluating the credibility of information sources
 - C) Encrypting data
 - D) Creating search engines

Part II: Short Answer Questions

Instruction: Give brief answers for the following review questions:

1. Explain what a keyword and phrase search is and provide an example.
2. Describe how Boolean search works with an example of a Boolean AND search.
3. What is the significance of using a wildcard (*) in a search query?
4. Why is it important for web developers to understand copyright laws?
5. What are the main aspects of data protection that web developers should be aware of?

Answer key for Self-check questions

Self-check 3-1: Search Information

1. What is the primary function of a search engine?

The primary function of a search engine is to index, crawl, and retrieve information from web pages, making it easily accessible to users.

2. What is a Boolean search and how does it refine search results?

A Boolean search uses logical operators (AND, OR, NOT) to combine or exclude keywords, refining search results by including or excluding specific terms.

3. Give an example of how to use the 'site:' operator in a search query.

To search within a specific website, use the 'site:' operator. For example, "CSS Grid site: w3schools.com" will return results about CSS Grid specifically from w3schools.com.

Self-check 3-2: Critical Evaluation

1. What is the CRAAP framework and what does it stand for?

The CRAAP framework is a set of criteria used to evaluate information sources for credibility, reliability, and relevance. It stands for Currency, Relevance, Authority, Accuracy, and Purpose.

2. Why is it important to critically evaluate online sources?

It is important to critically evaluate online sources to ensure the information is accurate, reliable, and free from bias or misinformation.

3. How can you verify the authenticity of information found online?

You can verify the authenticity of information by cross-checking with multiple reliable sources, looking for citations and evidence, and using fact-checking websites.

Self-check 3-3: Copyright, Data Protection

1. What is copyright and why is it important in the digital world?

Copyright is a legal right that grants creators control over the use and distribution of their original works. It is important in the digital world to protect intellectual property and avoid legal issues.

2. What are some key aspects of data protection laws like GDPR and CCPA?

Key aspects include safeguarding personal information, granting individuals rights over their data (such as access and deletion), and ensuring organizations implement secure data handling practices.

3. How can web developers ensure compliance with data protection laws?

Web developers can ensure compliance by implementing secure protocols (like HTTPS), conducting regular data protection audits, providing clear privacy policies, and ensuring data is collected and stored securely.

UNIT 4

ONLINE COMMUNICATION AND COLLABORATION

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 4.1. Online communication
- 4.2. Online collaboration

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Enhance online communication and collaboration

Unit Overview

In this unit, we delve into the dynamic realm of online communication and collaboration, recognizing its pivotal role in the landscape of web development. We begin by examining the profound significance of effective communication and collaboration, laying the foundation for understanding their impact on successful web projects.

Key Terms: *communication, Collaboration, Email, etiquette, Email Signature, Autoreply, G Suite Microsoft 365, cloud computing*

4.1. Online communication

Dear Students! How do you think we can leverage digital tools to enhance communication and teamwork in our projects? In the upcoming section, we will uncover the essential tools and techniques for effective online communication and collaboration, crucial for successful web development and professional interactions.

Effective communication stand as fundamental pillars in the realm of seamless web development, playing pivotal roles in fostering innovation and productivity. These cornerstones form the basis

for successful collaboration among web development teams, ensuring that projects progress smoothly and efficiently. Here's how they contribute to the overall process:

- **Shared Understanding:**

Effective communication ensures that all team members share a common understanding of project goals, objectives, and requirements. Collaboration facilitates the exchange of ideas, aligning the team's vision and reducing the likelihood of misunderstandings.

- **Efficient Problem Solving:**

Rapid and open communication allows for quick identification and resolution of issues, preventing potential roadblocks. Collaborative efforts bring together diverse perspectives, enabling creative problem-solving and innovative solutions.

- **Streamlined Workflows:**

Well-established communication channels and collaboration tools streamline workflows, reducing delays and optimizing the development process. Clear communication of tasks and responsibilities ensures that everyone is on the same page, enhancing overall efficiency.

- **Innovation through Diversity:**

Collaboration brings together individuals with diverse skill sets and perspectives. Effective communication in this environment sparks creativity and innovation, leading to the development of unique and groundbreaking solutions.

- **Optimized Resource Utilization:**

Collaboration helps in the effective distribution and utilization of resources, preventing redundancy and optimizing productivity. Communication about resource availability and needs ensures that the right tools and talents are applied where they are most needed.

- **Adaptability to Changes:**

Open communication channels allow teams to adapt swiftly to changes in project requirements or scope. Collaboration ensures that adjustments can be made collectively, leveraging the strengths of the entire team.

- **Continuous Improvement:**

Feedback loops established through communication channels enable continuous improvement. Collaborative efforts in refining processes and methodologies contribute to the evolution of best practices within the development team.

In essence, effective communication and collaboration not only enable the seamless execution of web development projects but also create an environment conducive to innovation, adaptability, and continuous improvement. By recognizing these aspects as cornerstones, teams can unlock their full potential and deliver high-quality, cutting-edge solutions in the ever-evolving landscape of web development.

4.1.1. Functions and features of communication tools

4.1.1.1. Effective communication

Effective communication is crucial for personal and professional success. In today's digital age, various tools facilitate seamless communication, enabling individuals and teams to stay connected, share information, and collaborate efficiently. Effective communication is essential for web developers to collaborate with team members, clients, and stakeholders. Various digital tools facilitate seamless communication, ensuring that projects run smoothly and efficiently. This section explores key communication tools that are crucial for web developers, including email, messaging apps, video conferencing, and social media platforms.

4.1.1.2. Communication tools

A. Email:

Email is a fundamental communication tool that allows users to send and receive messages electronically. It is widely used for personal and professional communication, providing a reliable way to exchange information, documents, and media. Email is a vital tool for web developers to communicate formally with clients and colleagues, send project updates, share documents, and manage project correspondence.

Applications:

- Sharing detailed project updates.
- Sending official documentation and reports.
- Facilitating one-on-one or group discussions.

Example Platforms: Gmail, Outlook, Yahoo Mail



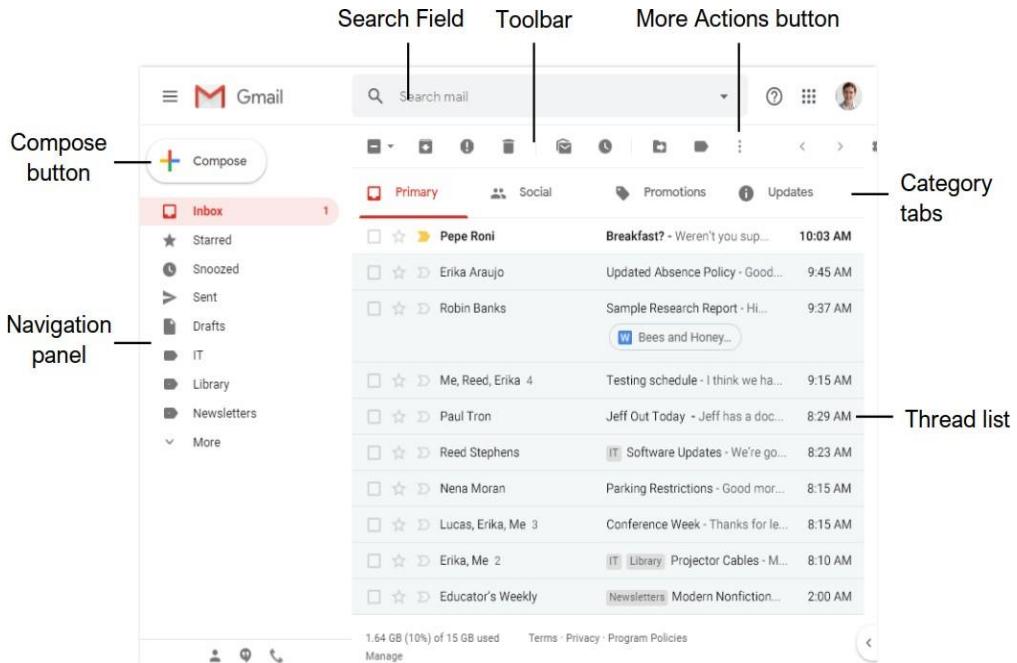


Figure 4 :Gmail interface

B. Messaging Apps:

Messaging apps provide real-time text communication, often with additional features such as voice and video calls, file sharing, and group chats. Instant messaging platforms enable real-time, informal communication. Messaging apps enable real-time communication, making it easier for web developers to discuss project details, solve issues quickly, and share resources. Examples include Telegram, which allows for organized conversations in channels, and WhatsApp, which supports group chats and multimedia sharing.

Applications:

- Quick exchanges for urgent matters.
- Group chats for team-wide discussions.
- Sharing files, links, and multimedia content.

Example Platforms: WhatsApp and Telegram



C. Video Conferencing:

Video conferencing tools enable face-to-face communication over the internet, facilitating virtual meetings, webinars, and online classes. Video conferencing tools are essential for remote web developers to conduct virtual meetings, collaborate with teams, and present work to clients. Tools like Zoom, which offers features like screen sharing and breakout rooms, and Microsoft Teams, which integrates with other Microsoft 365 applications, are widely used in the industry.

Applications:

- Conducting virtual meetings and brainstorming sessions.
- Hosting project presentations.
- Facilitating remote collaboration with a personal touch.

Example Platforms: Zoom, Microsoft Teams, Google Meet



D. Social Media Platforms:

Social media platforms allow users to connect, share content, and engage with others online. These platforms are used for networking, marketing, and staying informed about current events. Platforms for both professional and casual communication. Social media platforms help web developers stay updated on industry trends, network with other professionals, and showcase their work. LinkedIn is particularly useful for professional networking and finding job opportunities, while platforms like Twitter and Facebook are great for engaging with the developer community and sharing projects.

Applications:

- Engaging with a broader audience.
- Sharing industry updates and insights.
- Connecting with peers and professionals.

Example Platforms: LinkedIn, Twitter, Facebook and Instagram



Understanding the nuances of each communication tool empowers web development teams to choose the most suitable channels for different scenarios. Whether it's the formality of email, the immediacy of messaging apps, or the richness of video conferencing, leveraging these tools strategically enhances collaboration and contributes to the overall success of web development projects.

4.1.2. Using Email for Communication

Discussion



- How do you primarily use email?
- Are there specific purposes or objectives for which you rely on email?
- Are there any specific email etiquette guidelines or best practices you follow regularly?

Email is a fundamental tool for web developers, facilitating efficient and formal communication with clients, team members, and stakeholders. It provides a reliable platform for sharing project updates, discussing requirements, sending documents, and maintaining a written record of communications. Utilizing email effectively ensures clear and professional interactions, which is crucial for successful project management and client relations. Here are some key ways leaders can use email in their roles:

Key Ways Web Developers Can Use Email

1. Project Updates:

- Keep clients and team members informed about the progress of ongoing projects.
- Send regular status reports and milestones achieved.

2. Client Communication:

- Discuss project requirements, timelines, and deliverables.
- Address client queries and feedback promptly.

3. Team Collaboration:

- Coordinate tasks and share important information with team members.
- Use email to schedule meetings and set deadlines.

4. Document Sharing:

- Send design mockups, code snippets, and other project-related documents.
- Utilize attachments and links for easy access to shared files.

5. Networking:

- Connect with other professionals in the industry.
- Share your portfolio and seek collaboration opportunities.

4.1.2.1. Email etiquette

Email etiquette in a professional setting refers to the set of guidelines and principles that individuals should follow when using email for business or work-related communication. Adhering to proper email etiquette is crucial for maintaining a positive, respectful, and effective online communication environment. Here are some key elements of email etiquette in a professional setting:

- **Use a Professional Email Address:** Your email address should reflect your professionalism, typically using your name or a variation of it, and be associated with your organization if applicable.
- **Clear and Informative Subject Lines:** Use clear, concise subject lines that accurately describe the content of your email. This helps recipients understand the purpose of the message.
- **Greeting and Salutation:** Start your email with a polite greeting, such as "Dear [Name]" or "Hello [Name]." Use the recipient's name when possible.
- **Concise and Relevant Content:** Keep your emails clear and to the point. Address the primary purpose of the email in the first few lines. Avoid unnecessary jargon or overly long emails.
- **Professional Tone:** Maintain a professional and respectful tone in your emails. Avoid using slang, humor, or overly casual language unless it's appropriate for your specific workplace culture.
- **Grammar and Spelling:** Proofread your emails to ensure they are free from grammatical and spelling errors. A well-written email demonstrates attention to detail.
- **Use Cc and Bcc Judiciously:** Only use the "Cc" (carbon copy) and "Bcc" (blind carbon copy) fields when necessary. "Cc" is for recipients who should be aware of the email's content, while "Bcc" hides recipients' email addresses to protect their privacy.
- **Replying and Forwarding:** When replying to or forwarding emails, trim the previous message to include only relevant information. Avoid sending lengthy email chains.
- **Attachments:** Clearly label and attach files, and be cautious not to overload the recipient with large attachments. Mention the attached files in the email body.
- **Response Time:** Respond to emails promptly, especially if it's related to work tasks or time-sensitive matters. Even a brief acknowledgment can be helpful.
- **Respect Privacy:** Always respect the privacy of email recipients. Do not share their email addresses without permission, and avoid sharing sensitive information via email.
- **Signature:** Include a professional email signature with your name, title, contact information, and any relevant disclaimers or confidentiality notices.

- **Use Out-of-Office Messages:** If you'll be away from your email for an extended period, set up an out-of-office message to inform senders when they can expect a response.
- **Avoid Using Email for Emotional or Controversial Discussions:** Email is not the best medium for emotional or contentious discussions. Use other communication methods for such situations.
- **Follow Company Policies:** Adhere to your organization's specific email policies and guidelines, which may include rules for data security, email retention, and acceptable use.

Remember that email etiquette can vary slightly from one organization to another, so it's a good practice to be aware of your specific expectations regarding email communication. Ultimately, the goal of email etiquette in a professional setting is to facilitate clear, respectful, and efficient communication among colleagues, clients, and partners.

4.1.2.2. Key Email Skills

Developing strong email skills is essential for web developers to communicate effectively and professionally. Below are the key email skills students need to understand and utilize to communicate efficiently and professionally. Here are the important email skills you should have when you use Gmail and you are required to practice it step by step.

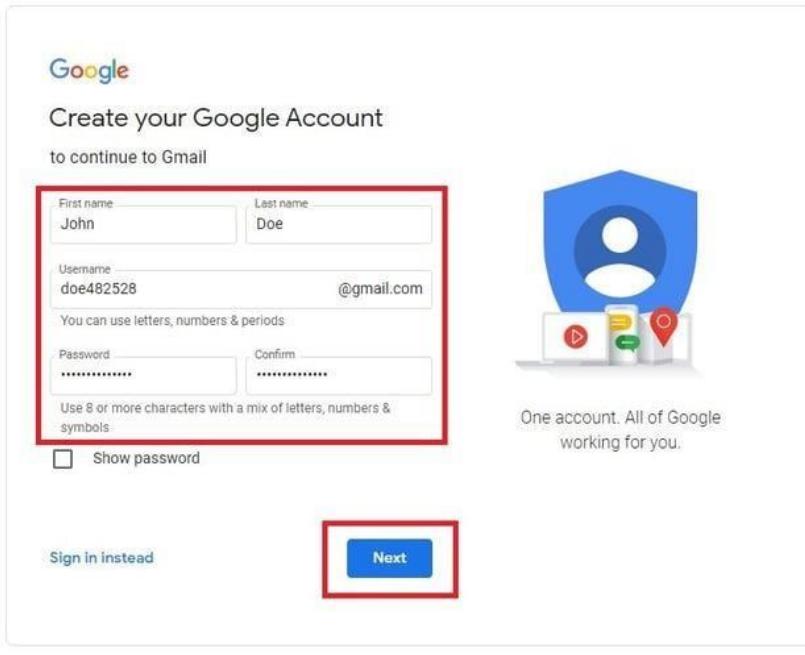
|  | Steps: Gmail usage |
|---|---|
| 1. Setting up a Gmail account: Creating an email account is the first step to using email services. This involves filling out necessary personal information, and setting up a secure password. Signing in involves entering your email address and password to access your inbox. | Steps: Step 1: Visit Gmail's website The first step to creating a Gmail account is to visit the https://www.google.com/gmail . Next, click on a blue button that says "Create an account" in the top-right corner of the page. |



Step 2: Fill in your information

You'll need to input your first and last name and the username you'd like for your email address. The username is what comes before "@gmail.com" in your email address (for instance, in "example@gmail.com," the username is "example").

You'll also need to set a strong password. Once you've done all of that, click on "Next."



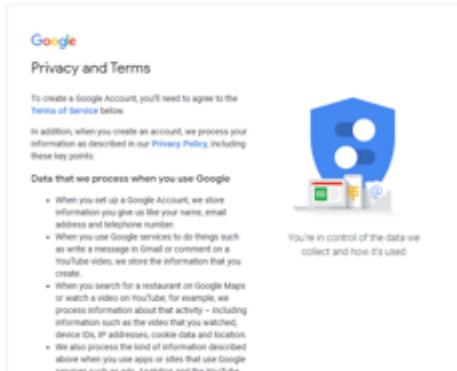
Step 3: Verify your phone number

Next, you must verify your phone number. You can choose to receive a verification code via text message or voice call. When you receive the code, type it into the verification box and click "Verify."



Step 4: Accept Google's terms of service and privacy policy

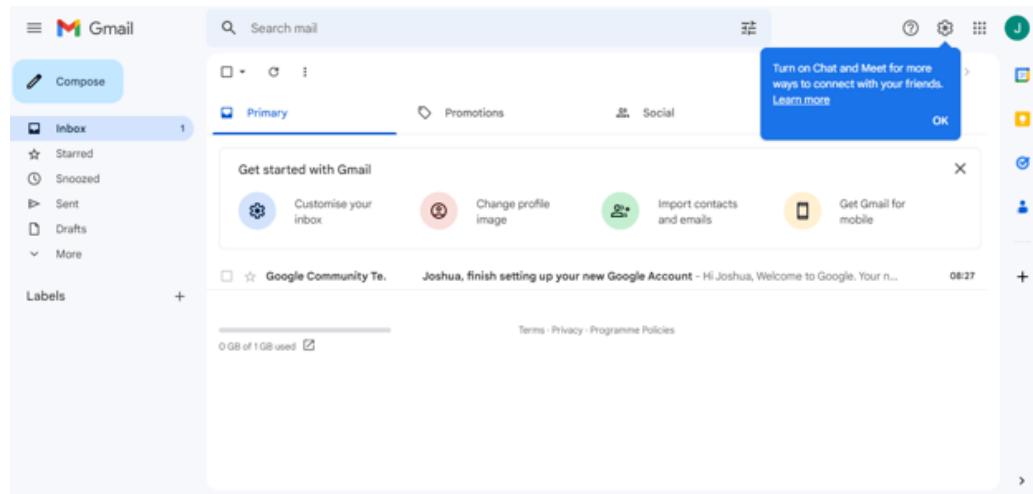
After verifying your phone number, the next step is to accept Google's terms of service and privacy policy.



Step 5: Personalize your account

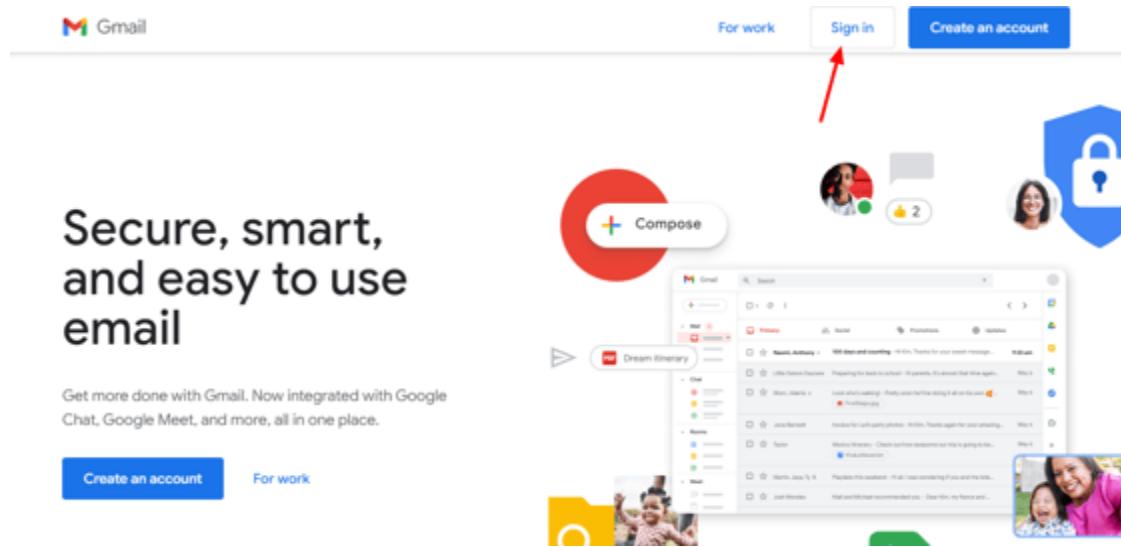
Your Gmail dashboard will be the next page you'll see. This is where you can view your emails, create and send new emails, and set up Gmail to forward your emails to another address if you'd like.

You can also personalize your account by adding a profile picture and filling in your personal information.



Step 6: Sign in to your new Gmail account

To sign in to your Gmail account, all you have to do is visit Gmail's website and click on the "Sign in" button in the top-right corner of the page, then enter your email and password.



2. Sending Email

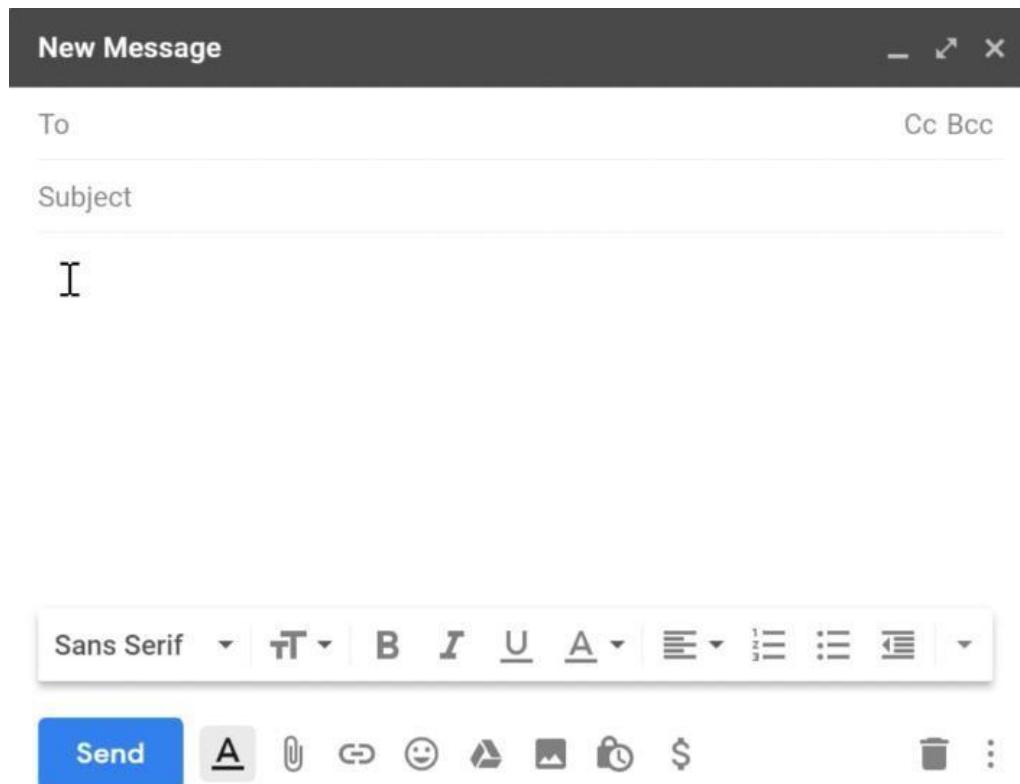
The screenshot shows a Gmail compose screen with several callout boxes explaining different parts of the interface:

- Subject**: The subject should say what the email is about. Keep the subject brief, but give the recipients a reasonable idea of what's in the message.
- To**: Olenna Mason, Julia Fillory, Henri Rousseau
- Recipients**: Recipients are the people you are sending the email to. You will need to type the **email address** for each recipient. Most of the time, you'll add recipients to the **To** field, but you can also add recipients to the **Cc** or **Bcc** fields.
- Cc and Bcc**: Cc stands for **carbon copy**. This is used when you want to send an email to someone who is not the main recipient. This helps to keep these people **in the loop** while letting them know that they probably don't need to reply to your message. Bcc stands for **blind carbon copy**. It works almost the same way as Cc, except that all of the email address in the Bcc fields are **hidden**, making it ideal when emailing a **large number** of recipients or when **privacy** is needed.
- Photography Studio Grand Opening!**
- Body**: The body is the actual text of the email. Generally, you'll write this just like a normal letter, with a greeting, one or more paragraphs, and a closing with your name.
- Hi Everyone,**
- I have exciting news! I am opening a new studio, EC Photography. We offer professional photography services for entertainment and events.**
- Hope to see you there!**
- Sans Serif**: Font dropdown menu.
- Send Button**: When you are satisfied with your message, click **Send** to send it to the recipients.
- Send**: The send button.
- Formatting Options**: Click the **Formatting** button to access **formatting**.
- Add Attachment**: An attachment is a file (like an image or a document) that is sent along with the email message. Gmail allows you to include multiple attachments.
- Formatting toolbar**: Includes options for bold, italic, underline, superscript, and other styling.
- Save**: Saved button.
- Compose**: Compose button.
- Inbox**: Inbox button with a count of 4.

Writing a professional email requires clarity, conciseness, and a formal tone. Important components include a clear subject line, a formal greeting, an organized message body, and a polite closing. This ensures that the message is understood and leaves a positive impression.

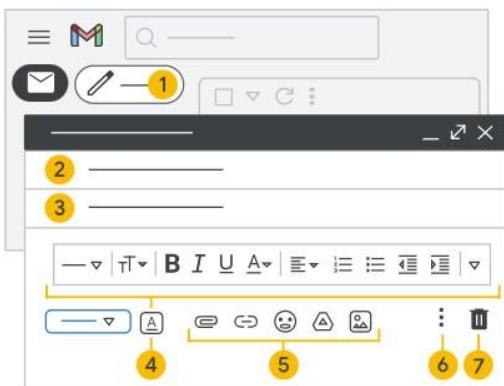
1. On your computer, go to Gmail.
2. At the top left, click **Compose**.
3. In the "To" field, add recipients. You can also add recipients in the "Cc" and "Bcc" fields.

When you compose a message, with a "+" sign" or "@mention" and the contact's name in the text field.



4. Add a subject.
5. Write your message.
6. At the bottom of the page, click Send.

Compose an email



- 1 Click **Compose** to create an email
- 2 Add recipients
- 3 Add a subject
- 4 Format text
- 5 Add files and links
- 6 Show more options, such as templates
- 7 Delete draft

3. Using Cc and Bcc:

The Cc (carbon copy) field is used to send a copy of the email to additional recipients whose responses are not required. The Bcc (blind carbon copy) field allows sending a copy to recipients without others knowing. These fields help manage communication transparency and privacy.

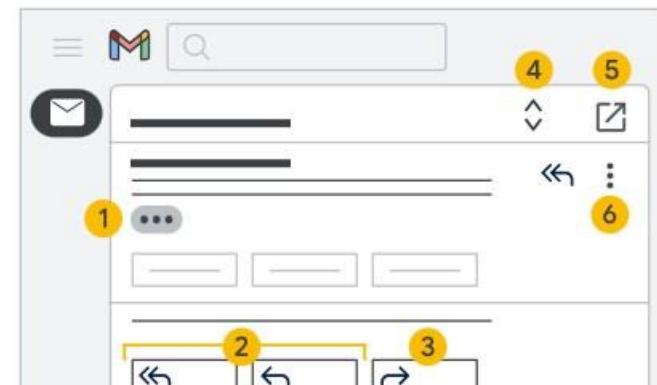
4. Adding Attachments:

Attachments are files sent along with an email. These can include documents, images, or other media. It is important to ensure attachments are relevant, appropriately named, and not excessively large to avoid delivery issues.

5. Replying and Forwarding:

Replying to an email involves responding to the sender's message, while forwarding allows you to send the original email to new recipients. Both actions should be done thoughtfully to maintain the context and relevance of the communication.

Reply to an email



- 1 See previous responses
- 2 Reply to email
- 3 Forward email
- 4 Expand all email
- 5 Open email in a new window
- 6 Show more options

The diagram illustrates a user interface for managing emails. At the top right are two buttons: 'Expand all email' and 'Open in new window'. Below them is a list of three emails, each with a small profile picture. To the right of the list are two buttons: 'Forward email' and a context menu icon with three dots. A blue line connects the 'Forward email' button to a larger box containing a 'Forward' button and several other smaller, illegible buttons. Another blue line connects the context menu icon to the same large 'Forward' box. At the bottom left, there are two buttons: 'Reply' and 'Forward'. A blue line connects the 'Reply' button to a box labeled 'See previous responses'. Another blue line connects the 'Forward' button in the large box to a box labeled 'Reply to email'.

Expand all email Open in new window

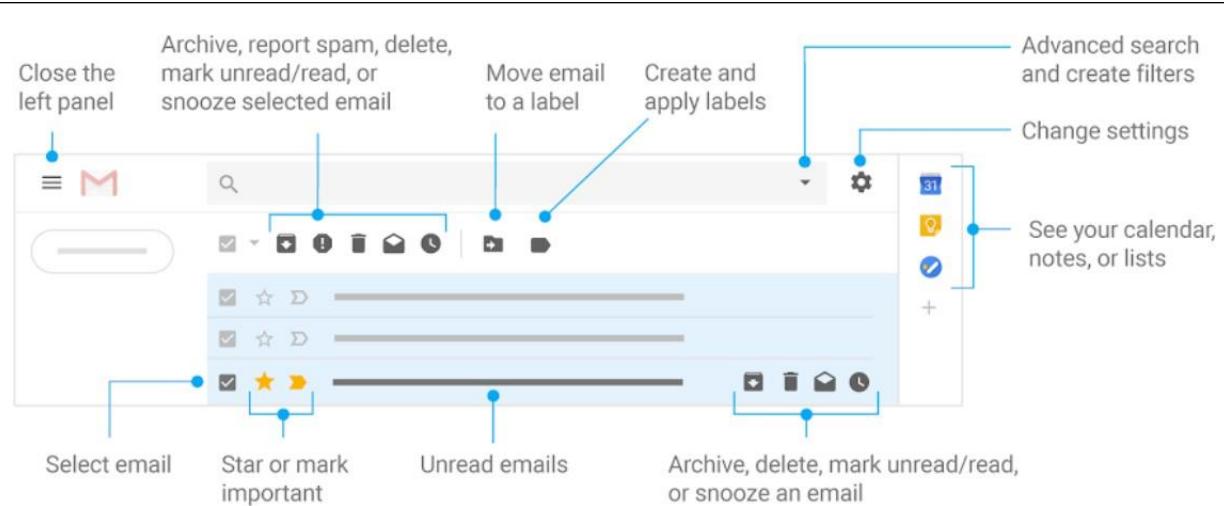
Forward email

See previous responses Reply to email

Reply Forward

6. Using Labels and Filters:

Labels help categorize emails for better organization, while filters automate the sorting process based on specific criteria such as sender, subject, or keywords. These tools are essential for managing a cluttered inbox and keeping track of important emails.

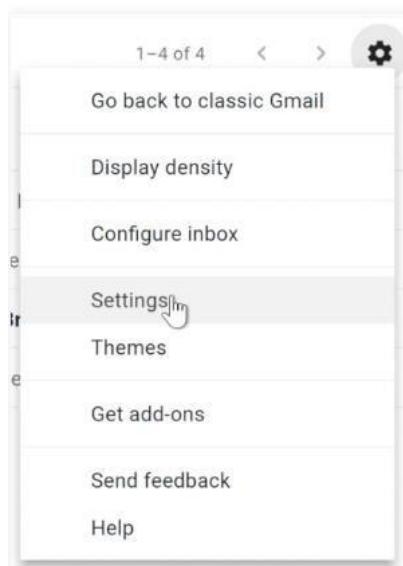


7. Setting Up an Email Signature:

An email signature provides a professional sign-off that typically includes your name, job title, contact information, and relevant links. It ensures consistency and professionalism in all email communications.

Steps

1. Open Gmail.
2. At the top right, click Settings and then See all settings.



3. In the "Signature" section, add your signature text in the box. If you want, you can format your message by adding an image or changing the text style.

Tip: Your image also counts toward the character limit. If you get an error, try to resize the image.



4. At the bottom of the page, click Save Changes.



8. Out-of-Office Autoreply(vacation Reply):

An out-of-office autoreply is a pre-written response that is automatically sent to anyone who emails you while you are unavailable. It informs senders of your absence and provides alternative contact information if necessary.

Steps

1. On your computer, open Gmail.
2. In the top right, click Settings Settings and then See all settings.
3. Scroll down to the 'Out of Office AutoReply' section.
4. Select Out of Office AutoReply on.
5. Fill in the date range, subject and message.
6. Under your message, tick the box if you only want your contacts to see your Out of Office reply.
7. At the bottom of the page, click Save changes.

Vacation responder:
(sends an automated reply to incoming messages. If a contact sends you several messages, this automated reply will be sent at most once every 4 days)
[Learn more](#)

Vacation responder off
 Vacation responder on

First day: June 15, 2021 Last day: June 22, 2021

Subject: Out of office

Message:

Sans Serif T I U A G S H I J K L M N P

[Plain Text](#)

I ran off to join the circus. Lucky for you, I'll be back in the office on June 23 and will reply to your email then.

Only send a response to people in my Contacts

groovyPost.com → Save Changes Cancel

Note: If you have a Gmail signature, it will be shown at the bottom of your out of office reply.

4.2. Online Collaboration

Collaboration refers to the joint effort of individuals or groups working together to achieve a common goal. In the context of web development, collaboration is essential due to the complexity of tasks involved, which often require diverse skill sets. Web developers collaborate to design, build, and maintain websites and web applications. Effective collaboration ensures seamless communication, improved problem-solving, and increased productivity in the development process.

4.2.1. Cloud-Based Collaboration Tools

Cloud-based collaboration tools have revolutionized the way we work together. Google Workspace (formerly G Suite) and Microsoft 365 are two widely used suites that provide a range of applications, including Google Docs, Google Sheets, Microsoft Word, and Microsoft Excel. These tools enable real-time collaboration, document sharing, and seamless communication.

Key concepts of cloud computing

Cloud computing is the use of computing resources that are delivered as a service over the Internet. End users access cloud-based applications through a web browser or a light-weight desktop / mobile app while the business software and user's data are stored on servers at a remote location.

Cloud computing facilitates online and mobile collaboration in 2 main ways:

- Provides storage of shared documents and files.
- Offers a range of online applications and tools.

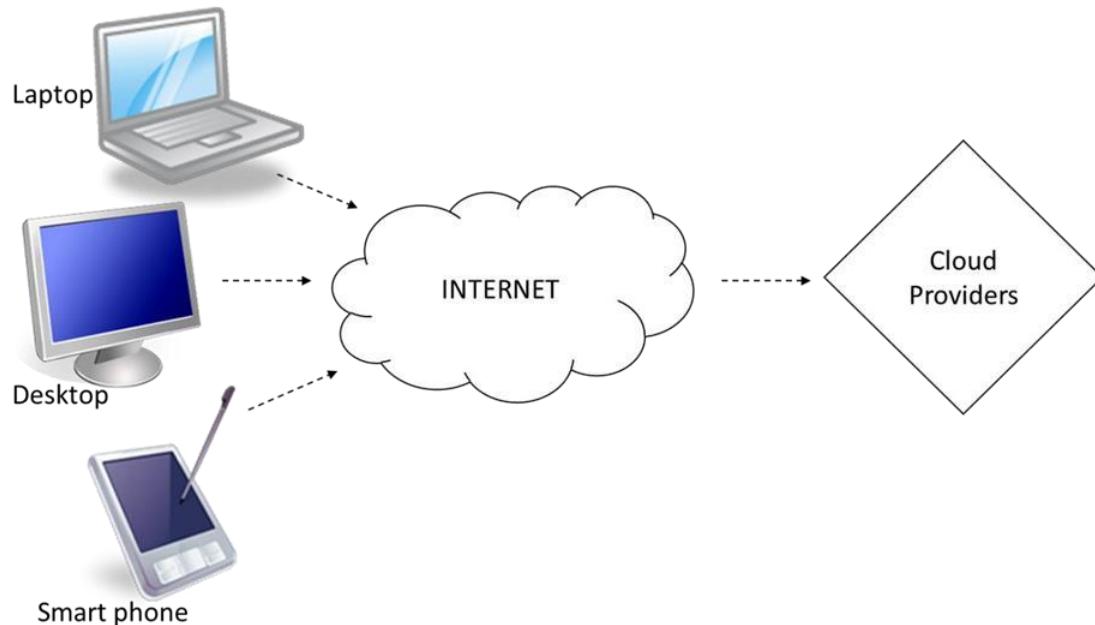


Figure Error! No text of specified style in document.-3 Cloud Computing Architecture

Benefits of Cloud Computing

- **Reduced cost:** Upfront infrastructure and travel costs can be avoided.
- **Enhanced mobility:** Cloud computing provides the opportunity to work and access information and services from virtually anywhere.
- **Scalability:** The impact on the physical facilities (such as work stations) is minimum when there are changes in the size of an online collaboration team. The team is hence highly scalable and only needs to consider the number of members needed in the team depending on the goals of the team.
- **Automatic Updates:** Online collaborative tools will usually allow users to set automatic updates, such as sending an out of office notice to incoming emails. The team can hence work more efficiently with reduced workload in performing manual updates to many tasks.

Functions and features of Collaboration Tools

The following are Key Characteristics of Online Collaborative Tools in common:

- **Multiple users:** Multiple users can access, work and contribute on the same document or projects.
- **Real time:** Responses, editing and updates are shown immediately, thus saving time for all involved.
- **Global reach:** Most online collaborative tools can be utilized from any location with the right set-up and Internet connection. Dispersal of information globally is easily achieved.
- **Concurrent access:** Concurrent multiple user access allows for faster editing and decision making on tasks.

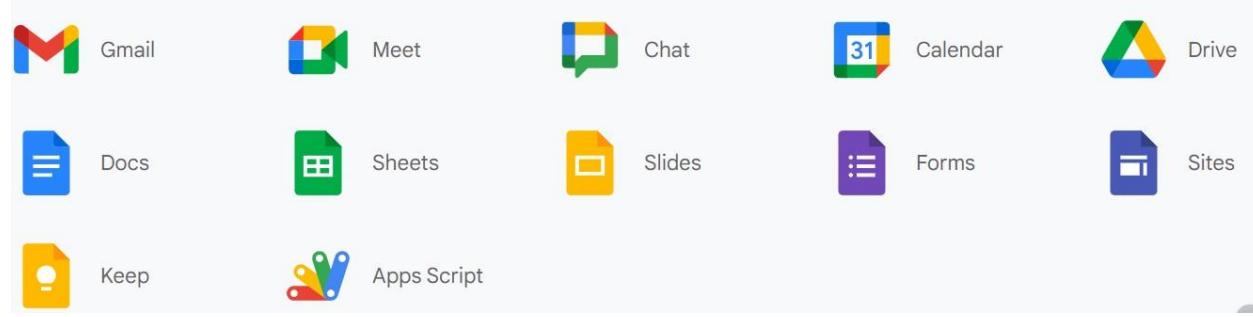
4.2.1.1.Common collaboration Tools

Collaboration tools are designed to enhance productivity and streamline teamwork by providing shared spaces for communication, document creation, and project management. For web developers, these tools facilitate real-time collaboration, version control, and seamless integration of workflows. This section introduces some of the most widely used collaboration tools, including Google Workspace, Microsoft 365, Slack, and GitHub.

A. Google Workspace:

Google Workspace is a suite of cloud-based productivity tools designed for collaboration and communication. It includes Gmail for email, Google Drive for file storage, Google Docs for document creation, Google Sheets for spreadsheets, and Google Meet for video conferencing. For example, multiple developers can simultaneously edit a Google Doc to collaborate on project specifications.

Included applications



B. Microsoft 365:

Microsoft 365 offers a comprehensive suite of productivity tools, including Outlook for email, OneDrive for file storage, Word for document creation, Excel for spreadsheets, and Teams for communication and collaboration. It provides robust integration, allowing web developers to collaborate on documents in real-time and hold virtual meetings. For instance, a team can use Microsoft Teams to chat and video conference while co-authoring a document in Word.



C. GitHub

GitHub is a web-based platform for version control and collaborative software development, using Git. It allows web developers to work on projects simultaneously, track changes, and manage code with features like pull requests and issue tracking. For instance, an open-source project on GitHub can have multiple contributors who submit pull requests to improve the code, which are then reviewed and merged by project maintainers. GitHub also serves as a portfolio for developers to showcase their work and collaborate with others in the community.



4.2.2. Real-Time Collaboration on Documents and Projects

Web-based productivity applications, also known as web apps, are software programs or tools that run in a web browser rather than being installed on a local computer or device. These applications are accessible from anywhere with an internet connection, making them convenient and versatile for a wide range of tasks. They are particularly valuable for collaborative work and remote access. Nowadays a variety of web apps such as word processors, spreadsheets, and presentation applications can be accessed using almost any browser.

Google Workspace (formerly G Suite) and Microsoft 365 (formerly Office 365) are two of the most popular and widely used cloud-based productivity and collaboration suites. They offer a range of applications and services that cater to individuals, businesses, and educational institutions. For this course this book will demonstrate you Google Workspace tools in various ways to enhance different work activities:

Online storage

Storing your work files online is the first step towards online collaboration. Online storage is a file hosting service that allows you to upload, store, and access your files online. It is also referred to as a cloud storage service, an online file storage provider, or cyberlocker. Google Docs, One Drive and Dropbox are examples of file storage and sharing services, aimed at allowing users to upload and share all types of files across all the devices they use.

| | | |
|---|---|--|
|  Google Drive |  OneDrive |  Dropbox |
| https://drive.google.com | https://onedrive.live.com | https://www.dropbox.com |

Google Drive

Google Drive serves as a cloud-based storage solution for documents, images, and other files. It is an option available for storing and sharing documents online. It provides several functionalities to help us manage our files and their access. It allows us to upload files from our devices and grant access to these files to others. Leaders can organize and share files with team members or students, ensuring access from any device with internet connectivity.

- **Document Storage:** Managers can use Google Drive to store and organize important documents, reports, and files securely in the cloud. This ensures easy access from any device and simplifies file management.
- **Collaboration:** Google Drive enables managers to collaborate on documents, spreadsheets, and presentations in real-time using Google Docs, Sheets, and Slides. Multiple team members can edit and comment on the same document simultaneously, streamlining the collaborative process.
- **File Sharing:** Managers can share files and folders with team members, clients, or partners, allowing for easy information dissemination and collaboration on projects.
- **Version Control:** Google Drive automatically saves versions of documents, making it easier for managers to track changes and revert to previous versions when necessary.

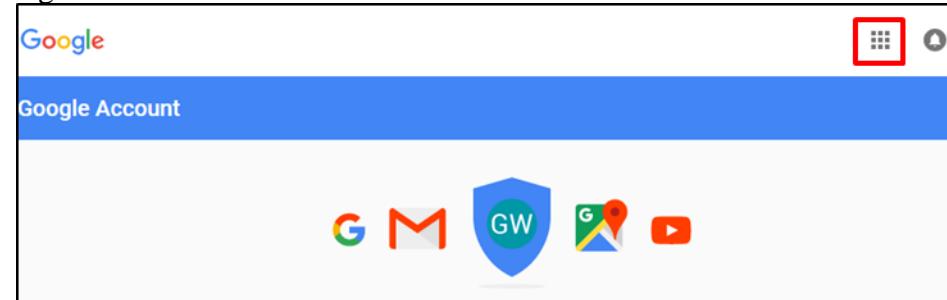


Steps: To open Google Drive:

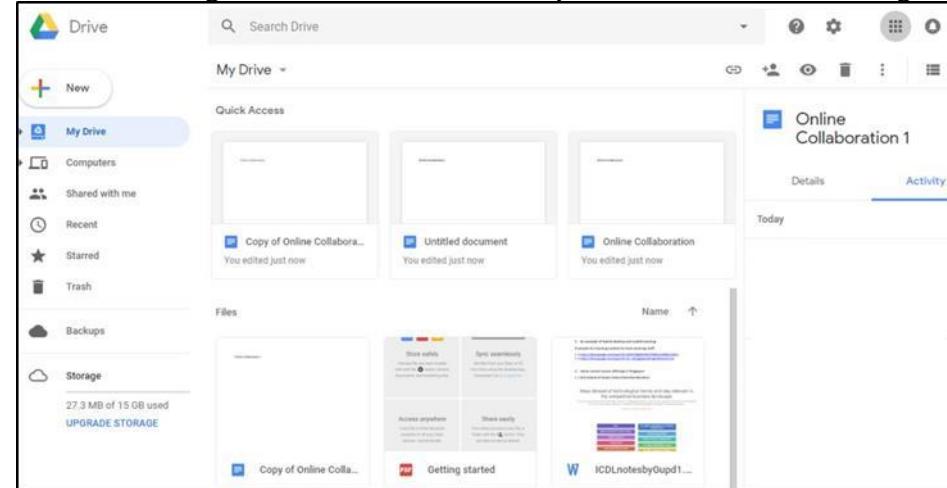
1. Go to the webpage <https://accounts.google.com> and click Sign in.
2. Enter your registered email or phone account identity and click NEXT.

The image shows the Google sign-in page. It has a text input field labeled "Email or phone" with a placeholder "Enter email". Below it is a "Forgot email?" link. A note says "Not your computer? Use Guest mode to sign in privately. [Learn more](#)". There are "Create account" and "NEXT" buttons at the bottom.

3. Enter your password.
4. Click on the tile icon at the right-hand corner of the opening page after successful sign-in:



5. Select the Google Drive icon from the drop-down menu and the Google



6. Drive screen will appear.

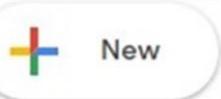


Steps: Managing Files and Folders

You can use the Google Drive account to create folders, upload and delete files.

Create Folder

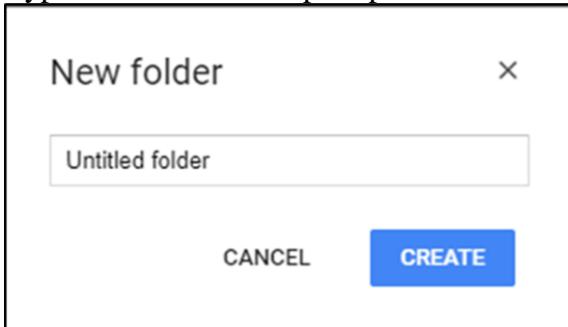
1. Click on New.



2. Click on the option **Folder**.



3. Type in a name in the space provided under New Folder.

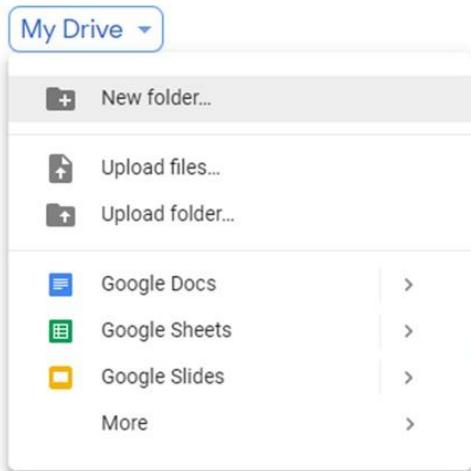


4. Click on **Create**



Steps: Upload a File

1. Click on the folder to add the file to.
2. Click on Upload files.



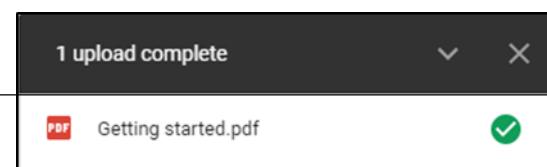
3. Locate the file you wish to upload from your computer or other storage device.
4. Click on the file name of the file you wish to upload.
5. Click Open or double-click on the file name using your mouse.
6. To upload the file quickly, reduce the size of active current window size by clicking on the following icon.



7. Drag the file to upload from the computer to the space under Files:

The screenshot shows the 'Files' page with several sections: 'Store safely', 'Sync seamlessly', 'Access anywhere', and 'Share easily'. At the bottom right, there is a 'Getting started' button with a red box drawn around it. The URL 'ICDLnotesbyGup...' is visible at the bottom right of the page.

8. A message appears in the bottom panel, displaying the upload progress.
9. The pop-up panel will display 1 upload complete upon successful upload.



Online Calendars

An online calendar allows you to keep track of your events and activities. You can let others see your calendar, and view schedules that others have shared with you. Google Calendar simplifies scheduling and event management. You can create and share calendars to coordinate meetings, events, and deadlines. The calendar can also integrate with other apps to send event reminders.

Google Calendar:

- **Schedule Management:** Users can use Google Calendar to create, manage, and organize their schedules. They can add and track meetings, appointments, and deadlines, helping them stay on top of their time management.
- **Event Coordination:** Google Calendar allows you to schedule meetings, set reminders, and invite participants. It streamlines event coordination and ensures that team members are on the same page.
- **Sharing Calendars:** Users can share their calendars with team members, helping everyone stay aware of each other's schedules and reducing scheduling conflicts.
- **Integration:** Google Calendar can be integrated with other tools like Gmail, making it easier for managers to create events directly from emails.

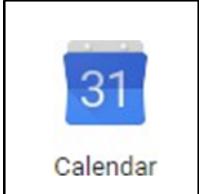


Steps: Sharing Calendar using Google Calendar

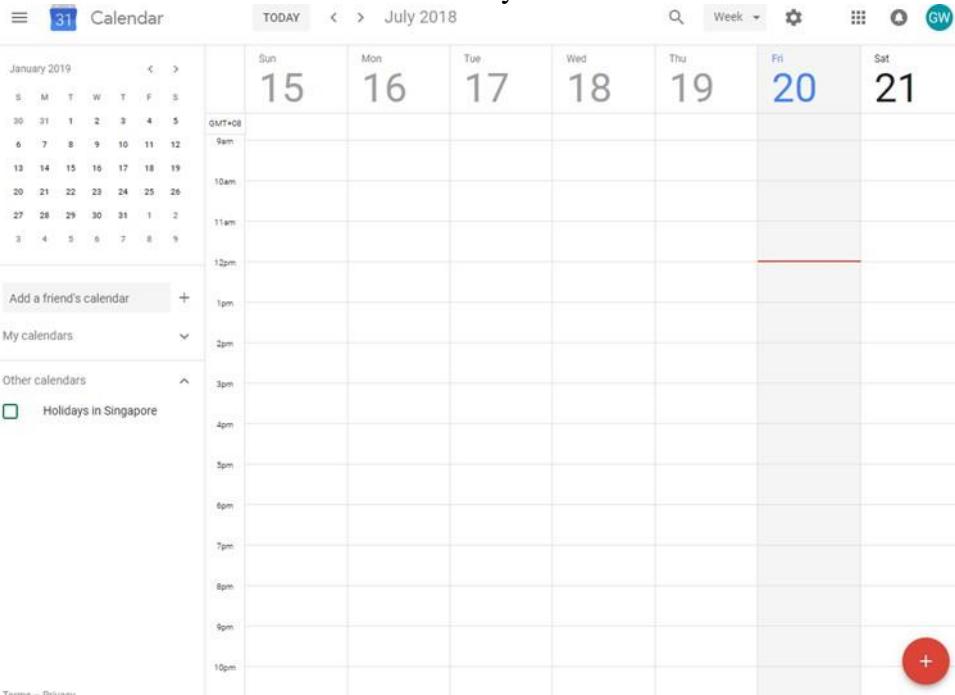
1. To Share a Calendar Go to the webpage <https://calendar.google.com> or click on the Google menu icon located on the right-hand corner of Google Chrome:



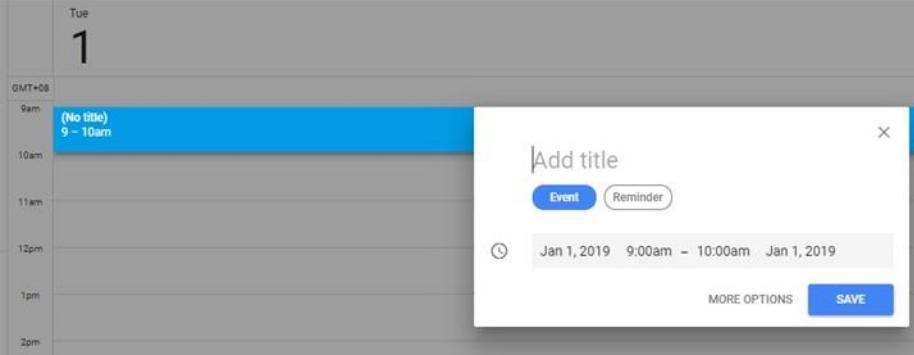
2. On the expanded menu drop-down window select **Calendar**.



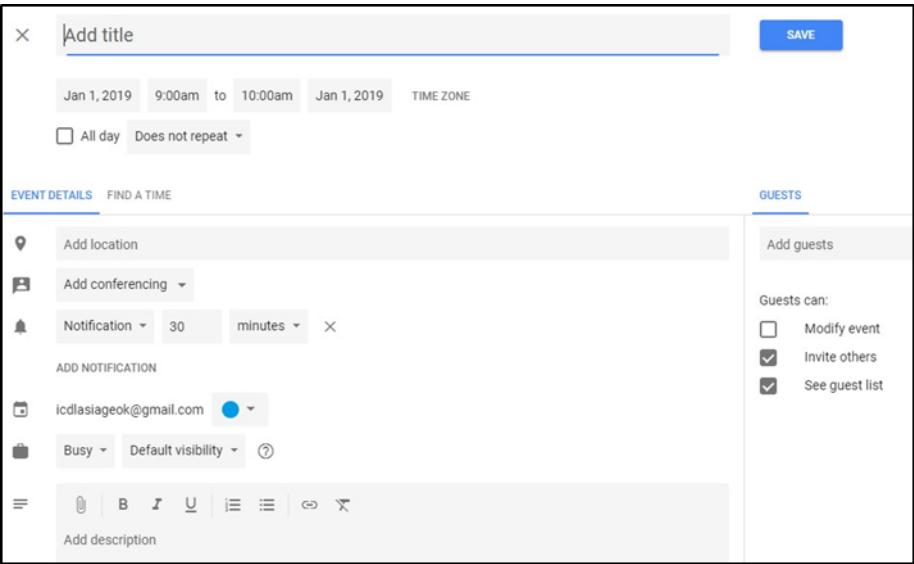
3. Select a date on the Calendar on which you wish to mark an event.



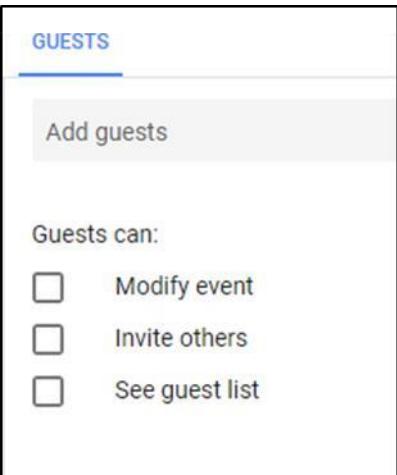
4. Select a time slot on the selected date and add a title.



5. To include more details for the marked event, click on MORE OPTIONS.
6. You may add details on the location, notification and add more descriptions about the event under the EVENT DETAILS.



7. To share your calendar, enter the email address of the person with whom you want to share the event and check the appropriate boxes against the options provided under Guests can.



Productivity applications

Google Docs, Sheets, and Slides

Google Docs, Google Sheets, and Google Slides are part of Google Workspace, which is a suite of productivity applications offered by Google. Each of these tools serves a distinct purpose:

1. Google Docs:

- Google Docs is a cloud-based word processing application. It allows users to create, edit, format, and collaborate on documents online. Features of Google Docs include:
 - **Real-time collaboration:** Multiple users can work on the same document simultaneously and see each other's edits in real-time.
 - **Access from any device:** Documents are stored in the cloud, making them accessible from any device with an internet connection.
 - **Version history:** Google Docs saves versions of documents, enabling users to track changes and revert to previous versions if needed.
 - **Template library:** Users can choose from a variety of templates for different types of documents, such as resumes, reports, and newsletters.

2. Google Sheets:

- Google Sheets is a web-based spreadsheet application. It allows users to create, edit, analyze, and collaborate on spreadsheets. Key features of Google Sheets include:
 - **Data analysis:** Users can perform data analysis, create charts and graphs, and use formulas to perform calculations on spreadsheet data.
 - **Real-time collaboration:** Like Google Docs, Google Sheets enables multiple users to collaborate on a single spreadsheet simultaneously.
 - **Cloud storage:** Spreadsheets are stored in the cloud, ensuring easy access from anywhere and automatic saving.
 - **Import and export data:** Google Sheets supports importing and exporting data in various formats, making it compatible with other spreadsheet software.

3. Google Slides:

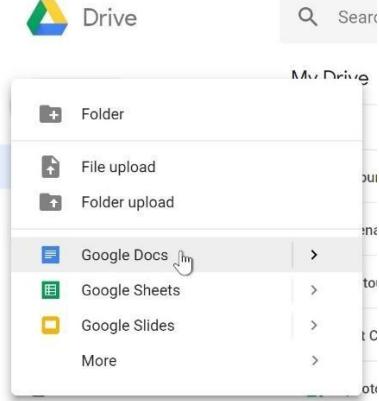
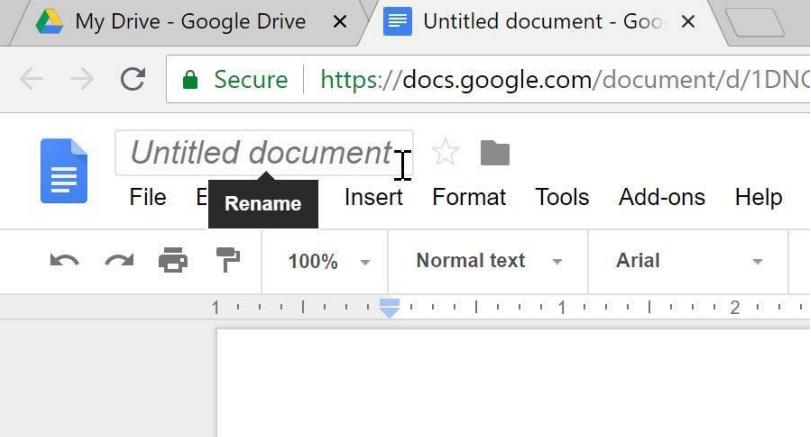
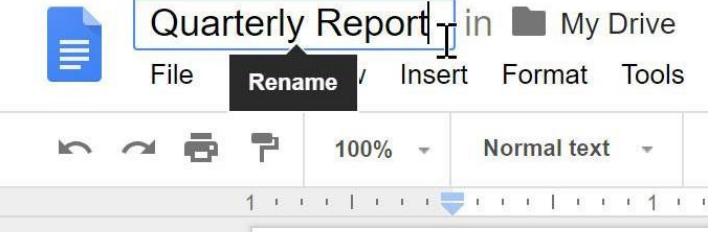
- Google Slides is a web-based presentation application, similar to Microsoft PowerPoint. Users can create, edit, and deliver presentations online. Key features of Google Slides include:
 - **Slide creation and editing:** Users can create slides with text, images, charts, and other multimedia elements.
 - **Presentation delivery:** Google Slides provides tools for delivering presentations, including speaker notes and presentation mode.
 - **Collaboration:** Multiple users can collaborate on a presentation in real-time, allowing for group input and feedback.
 - **Template gallery:** Users can choose from a variety of presentation templates to create professional-looking slideshows.

These three applications are accessible through a web browser and are designed for collaborative work. They are commonly used in both personal and professional settings for tasks such as

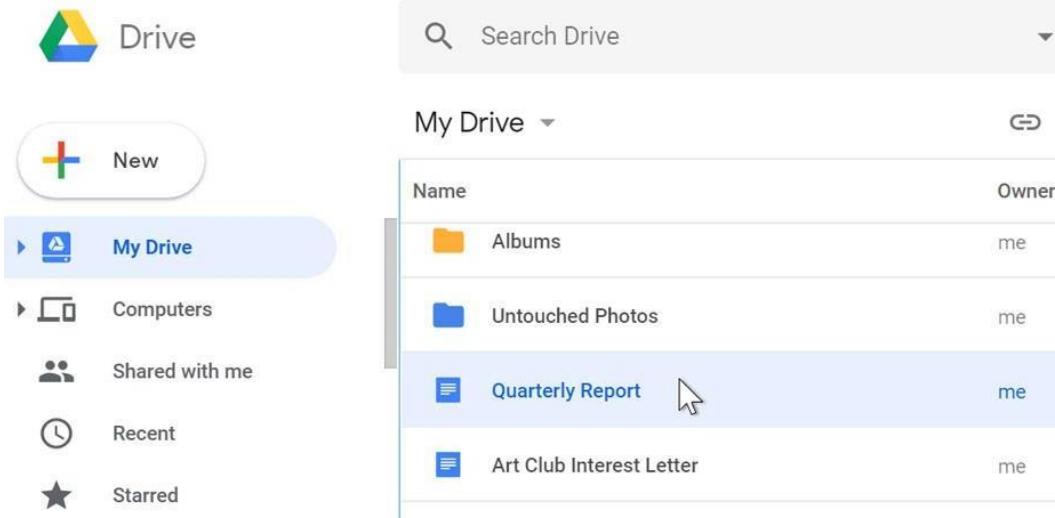
document creation, data analysis, and delivering presentations. Additionally, they integrate seamlessly with other Google Workspace applications and offer compatibility with common file formats used in productivity software.

Starting from the section below you will practically go through the different steps that are necessary in collaborating works using google docs.

 **Steps: To create a new file on Google Docs**

1. From Google Drive, locate and select the New button, then choose the type of file you want to create. In our example, we'll select **Google Docs** to create a new **document**.

2. Your **new file** will appear in a **new tab** on your browser. Locate and select **Untitled document** in the upper-left corner.

3. The **Rename** dialog box will appear. Type a **name** for your file, then click **OK**.


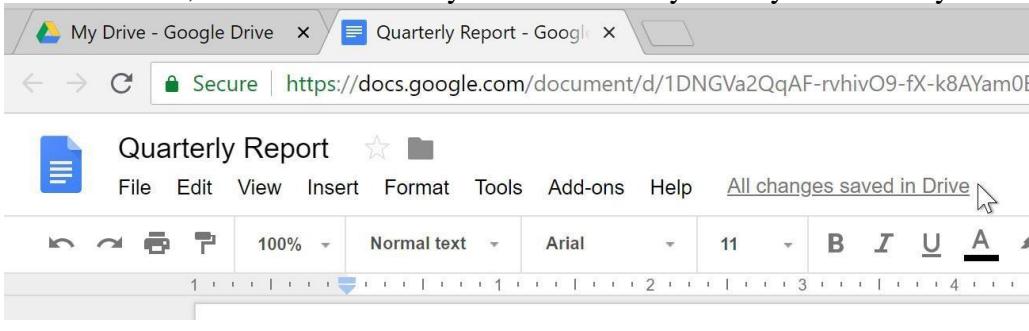
4. Your file will be **renamed**. You can access the file at any time from your Google Drive, where it will be **saved** automatically. Simply double-click to open the file again.



The screenshot shows the Google Drive interface. On the left, there's a sidebar with a 'New' button, 'My Drive' (which is selected and highlighted in blue), 'Computers', 'Shared with me', 'Recent', and 'Starred'. The main area is titled 'My Drive' and contains a table with five items:

| Name | Owner |
|--------------------------|-------|
| Albums | me |
| Untouched Photos | me |
| Quarterly Report | me |
| Art Club Interest Letter | me |

You may notice that there is no **Save** button for your files. This is because Google Drive uses **autosave**, which automatically and immediately saves your files as you edit them.



The screenshot shows a Google Document titled 'Quarterly Report'. The status bar at the top indicates the URL: <https://docs.google.com/document/d/1DNGVa2QqAF-rvhivO9-fX-k8AYam0E>. The status bar also shows 'Secure' and the text 'All changes saved in Drive' with a small arrow icon.



Steps: To share a file with a group of people

1. Locate and select the file you want to share, then click the **Share** button.

My Drive

| Name | Owner | Last modified |
|----------------------------|-------|---------------|
| Budget Proposals | me | Aug 7, 2018 |
| Travel Schedule - Montreal | me | Aug 7, 2018 |
| Elena's Files | me | Jan 14, 2013 |
| Albums | me | Jan 14, 2013 |

2. A dialog box will appear. In the **People** box, type the email addresses of the people you'd like to share the file with. If you want, you can add a **message** that will be emailed to the people you share the file with.

3. Click **Send**. Your file will be shared.

Share with others

Get shareable link

People

Julia Fillory Add more people...

Hi Julia,
Here's the schedule for our upcoming trip! I'm so excited!
Olenna

Send Cancel Advanced

For more control over your files, you can click the drop-down arrow to decide whether people can **edit**, **comment** on, or simply **view** the file.

Share with others Get shareable link 

People

Julia Fillory Add more people...  

Hi Julia,
Here's the schedule for our upcoming trip! I'm so ex
Olenna

Can organize, add, & edit Can view only 

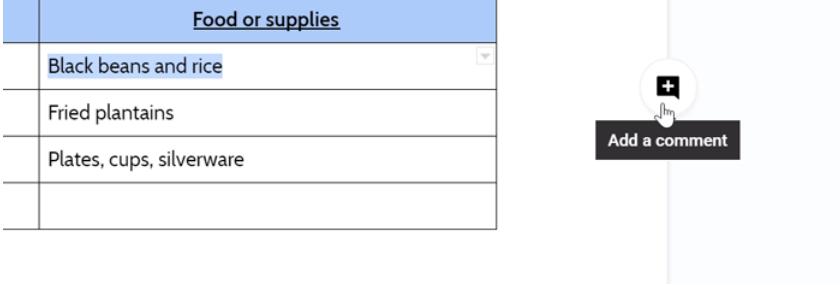
Send Cancel Advanced

Suggesting mode
Google Drive also has a feature called Suggesting mode, which is similar to the Track Changes feature in Microsoft Office. This allows each collaborator to make changes, while giving the other collaborators a chance to review the changes before making them permanent. This page has more information about how to make suggested edits and accept or reject other people's changes.

 **Steps: Adding and replying to comments**

❖ Comments are one of the features that allow you to collaborate in Google Docs. In this lesson, you'll learn how to create and reply to comments, tag collaborators, and more.

Anyone with **editor** or **commenter** access to the Google Doc can create a comment. There are several ways to add a comment, but the easiest way is to select the text that you want to comment on, then click the button that appears on the right side of the screen.



The comment box opens, and you can type your message.



Javier Flores

Could these be made vegetarian?

Comment

Cancel

If you want to make sure that a specific person sees your comment, you can tag them. To tag someone, type the **@ symbol** and start typing their email address. Then select them from your list of contacts.



Javier Flores

Could these be made vegetarian?

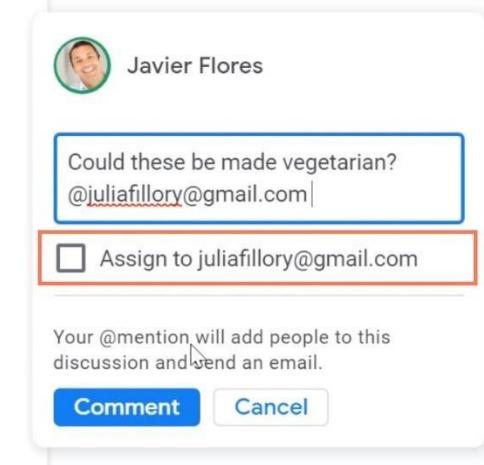
@juli



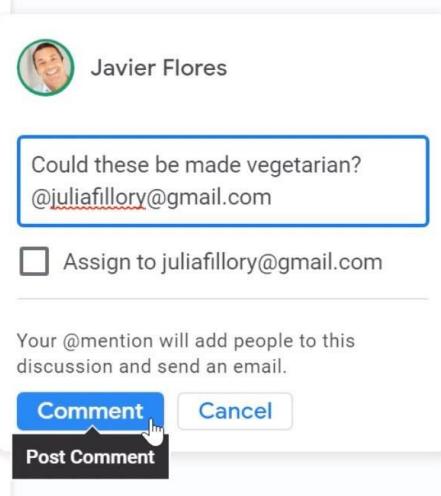
juliafillory@gmail.com



If you tag someone who doesn't already have access to the document, Google will ask you to share the document with them before posting your comment. When you tag someone, you have the option to assign the comment to them. Assigning a comment to someone means that they will be responsible for marking it as complete.

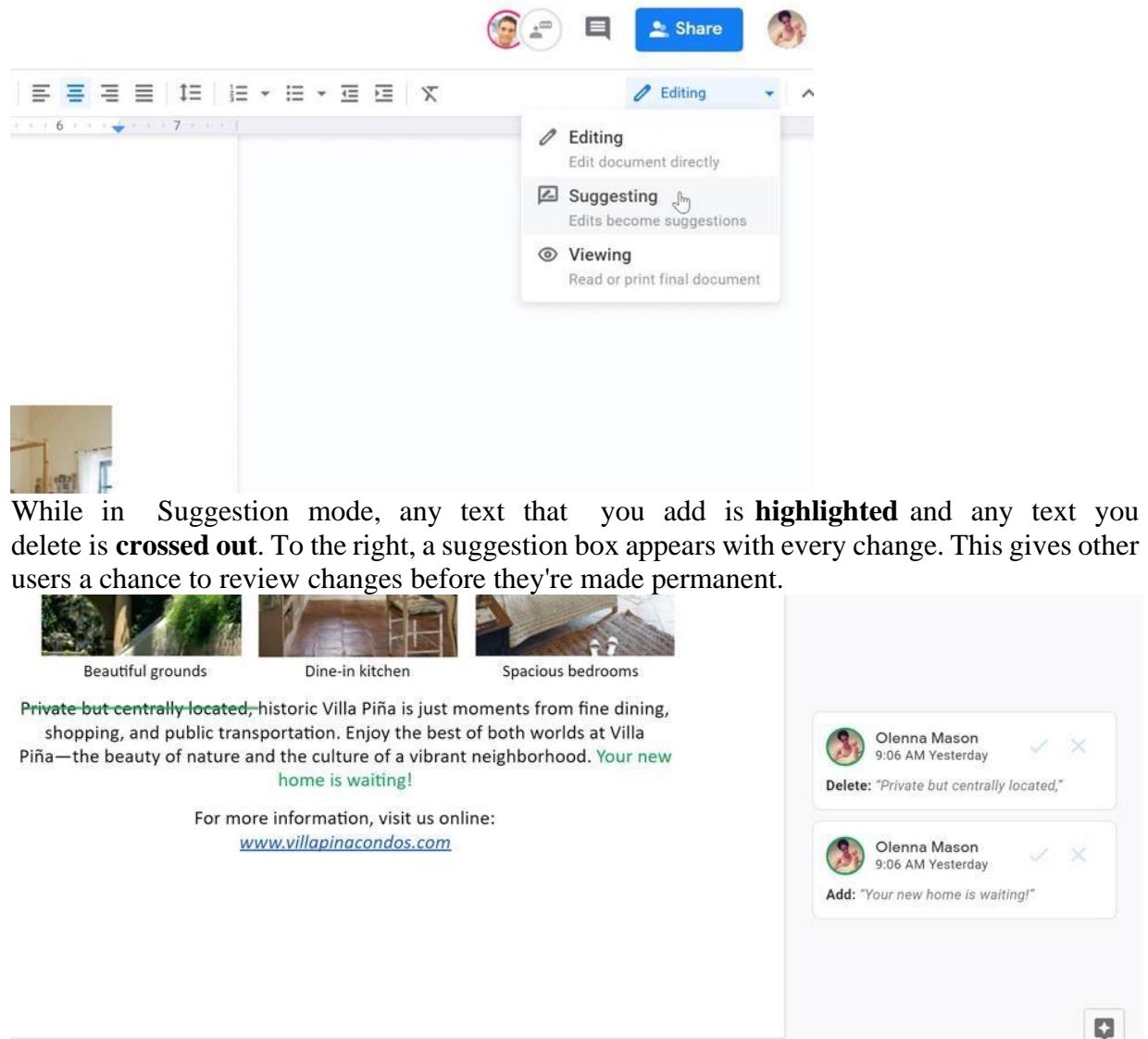


When you're finished typing your comment, click **Comment**.



Steps: Using Suggestion mode to track changes

When multiple people are working on a document, it's helpful to see how the document changes and grows over time. Google Docs lets you track any edits that are made, comment on them, and decide whether they should be added to the document. Google Docs calls this feature Suggestion mode, but it's very similar to the Track Changes feature in Microsoft Word. To track changes in Google Docs, you have to switch from **Editing mode** to **Suggestion mode**. Click the pencil icon underneath the Share button, and choose **Suggesting** from the drop-down menu.



The screenshot shows a web editor interface with a toolbar at the top. A dropdown menu is open, showing three options: "Editing" (selected), "Suggesting" (with a hand cursor icon over it), and "Viewing". Below the editor area, there are three images with captions: "Beautiful grounds", "Dine-in kitchen", and "Spacious bedrooms". A text block describes Villa Piña's location and features, with some text highlighted in green. To the right, a sidebar shows two suggestions from Olenna Mason, each with a delete button. At the bottom right of the editor area is a reply button.

While in Suggestion mode, any text that you add is **highlighted** and any text you delete is **crossed out**. To the right, a suggestion box appears with every change. This gives other users a chance to review changes before they're made permanent.

Private but centrally located, historic Villa Piña is just moments from fine dining, shopping, and public transportation. Enjoy the best of both worlds at Villa Piña—the beauty of nature and the culture of a vibrant neighborhood. **Your new home is waiting!**

For more information, visit us online:
www.villapinacondos.com

Reviewing changes

You can also see changes suggested by other people. If you have a question about a particular change, you can leave a comment in the suggestion box. Click the suggestion box, click the text field, write a response, and click **Reply**.

The screenshot shows a digital document interface with a light gray background. At the top, there's a header bar with a blue gradient. Below the header, the main content area has a white background.

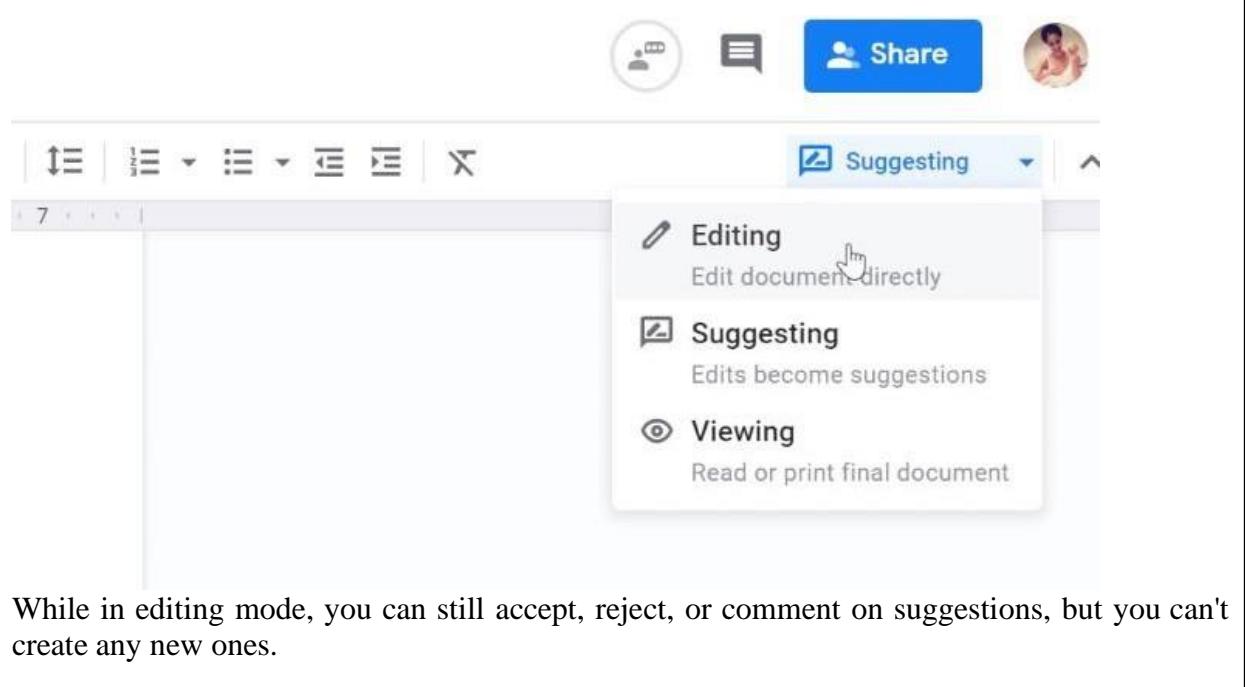
Suggestion 1: A comment from "Henri Rousseau" at 11:13 AM Today suggests adding the text "is the cost of rent". A blue callout box asks "Do we need to include this?" with "Reply" and "Cancel" buttons. A black button labeled "Reply to comment" is visible below the first comment.

Suggestion 2: Another comment from "Henri Rousseau" at 11:13 AM Today suggests replacing "June 1" with "May 15". A blue callout box asks "Do we need to include this?" with "Reply..." button. A black button labeled "Replace" is visible below the second comment.

Suggestion 3: A third comment from "Henri Rousseau" at 11:13 AM Today suggests replacing "ABOUT VILLA PIÑA" with "SAY HI". A blue callout box asks "Do we need to include this?" with "Reply..." button. A black button labeled "Replace" is visible below the third comment.

Text at the bottom: "To approve a change, click the **checkmark**. It will then become part of the document. To reject a change, click the **X**, and the suggestion will disappear."

Text at the bottom right: "To leave Suggestion Mode, click Suggesting in the top-right corner and select **Editing**."



The screenshot shows the Google Docs toolbar at the top with icons for file, edit, and share. A dropdown menu is open over the 'Suggesting' button, listing three modes: 'Editing', 'Suggesting', and 'Viewing'. The 'Suggesting' mode is currently selected. Below the toolbar, there's a large text area with some placeholder text and a small image of a person.

While in editing mode, you can still accept, reject, or comment on suggestions, but you can't create any new ones.

4.2.3. Conferencing and Virtual Meetings

Video conferencing tools have become integral in education, allowing educators and leaders to connect with students, colleagues, and stakeholders from anywhere. Platforms like Zoom, Microsoft Teams, and Google Meet enable virtual meetings, webinars, and interactive sessions. This topic will cover the use of these tools for effective online meetings.

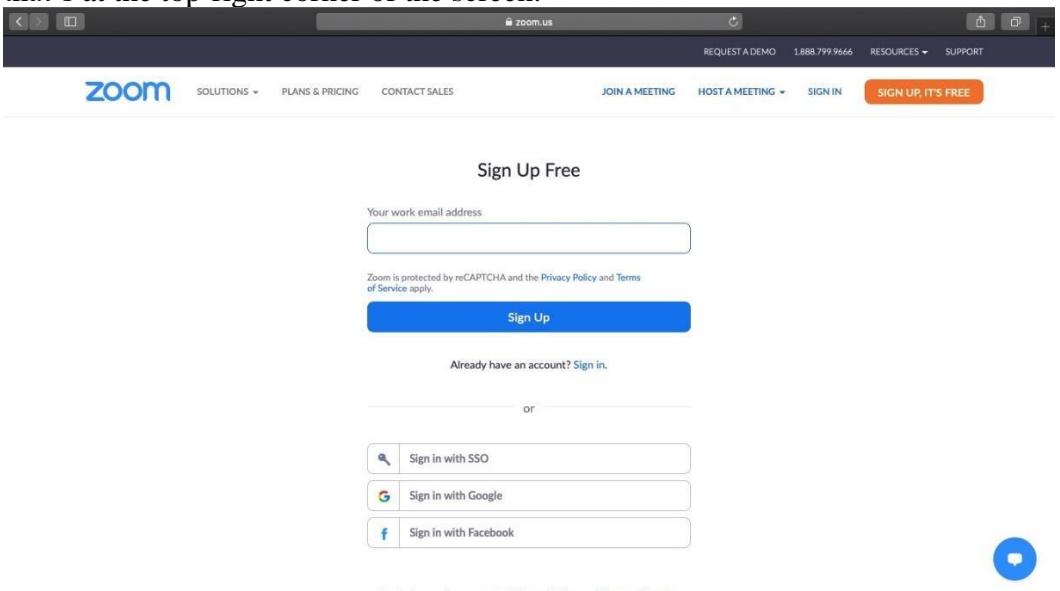
- **Google Meet:** Google Meet is a video conferencing tool, which is valuable for virtual meetings, lectures, and webinars. Educational leaders can host virtual discussions, engage with students or staff, and conduct training sessions.
- **Zoom Meeting App:** Zoom is a popular video conferencing and online meeting platform that provides a range of services for hosting and participating in virtual meetings, webinars, and collaborative sessions. Zoom has gained immense popularity for its user-friendly interface, robust features, and cross-platform compatibility.

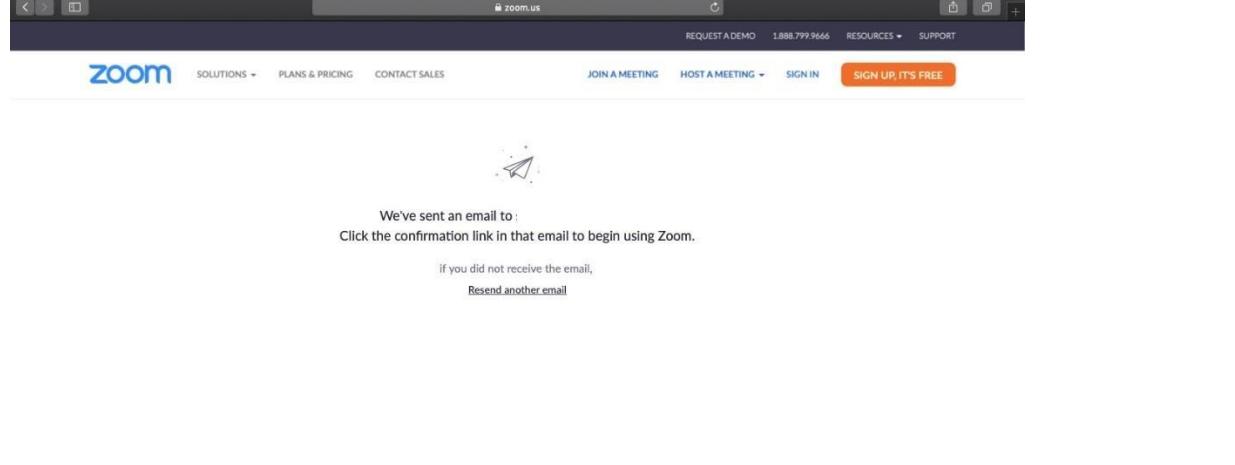
How to use Zoom meetings

Zoom makes it super easy for anyone to set up and conduct a virtual face-to-face meeting — but if you've never used the tool before, this can still be tricky. To help you out, here's a step-by-step guide to using Zoom Meetings the right way. We'll cover the steps on the desktop platforms on windows:

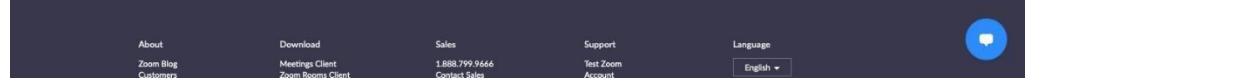
- a. How to Get Started with Zoom
- b. How to Set Up a Zoom Meeting
- c. How to Join a Zoom Meeting

- d. How to Schedule Meetings
- e. How to Record Meetings
- f. How to share screen

| | |
|--|--|
|  | Steps: Setting up online meeting session using Zoom meeting |
| <p>A. How to get started with Zoom</p> <p>Step 1: To get started with Zoom, head to their website, and click on the “SIGN UP” button that’s at the top-right corner of the screen.</p>  <p>The screenshot shows the Zoom sign-up page. At the top, there's a navigation bar with links for 'REQUEST A DEMO', '1.888.799.9666', 'RESOURCES', and 'SUPPORT'. Below that is a main heading 'Sign Up Free'. A large input field is labeled 'Your work email address'. Below it, a note says 'Zoom is protected by reCAPTCHA and the Privacy Policy and Terms of Service apply.' A blue 'Sign Up' button is centered. Below the button, a link says 'Already have an account? Sign in.' followed by an 'or' separator. There are three social login buttons: 'Sign in with SSO' (with a magnifying glass icon), 'Sign in with Google' (with a Google icon), and 'Sign in with Facebook' (with a Facebook icon). At the bottom left, a note says 'By signing up, I agree to the Privacy Policy and Terms of Service'. On the right side, there's a small blue circular icon with a white speech mark inside.</p> <p>Step 2: You have two options when it comes to creating a Zoom account. You can either:</p> <ul style="list-style-type: none">▪ Create a new account using your work email address.▪ Sign in using SSO (Single Sign-On) or your Google or Facebook account. <p>If you’re using Zoom for a company meeting or other work purposes, it’s best to sign up using your work email address.</p> <p>Step 3: Zoom will now send you an email with a confirmation link. Click on that link to go to Zoom’s Sign Up Assistant and sign in using your credentials.</p> | |



We've sent an email to: [redacted]
 Click the confirmation link in that email to begin using Zoom.
 If you did not receive the email,
[Resend another email](#)



About
[Zoom Blog](#)
[Careers](#)

Download
[Meetings Client](#)
[Zoom Rooms Client](#)

Sales
 1.888.799.9666
[Contact Sales](#)

Support
[Test Zoom Account](#)

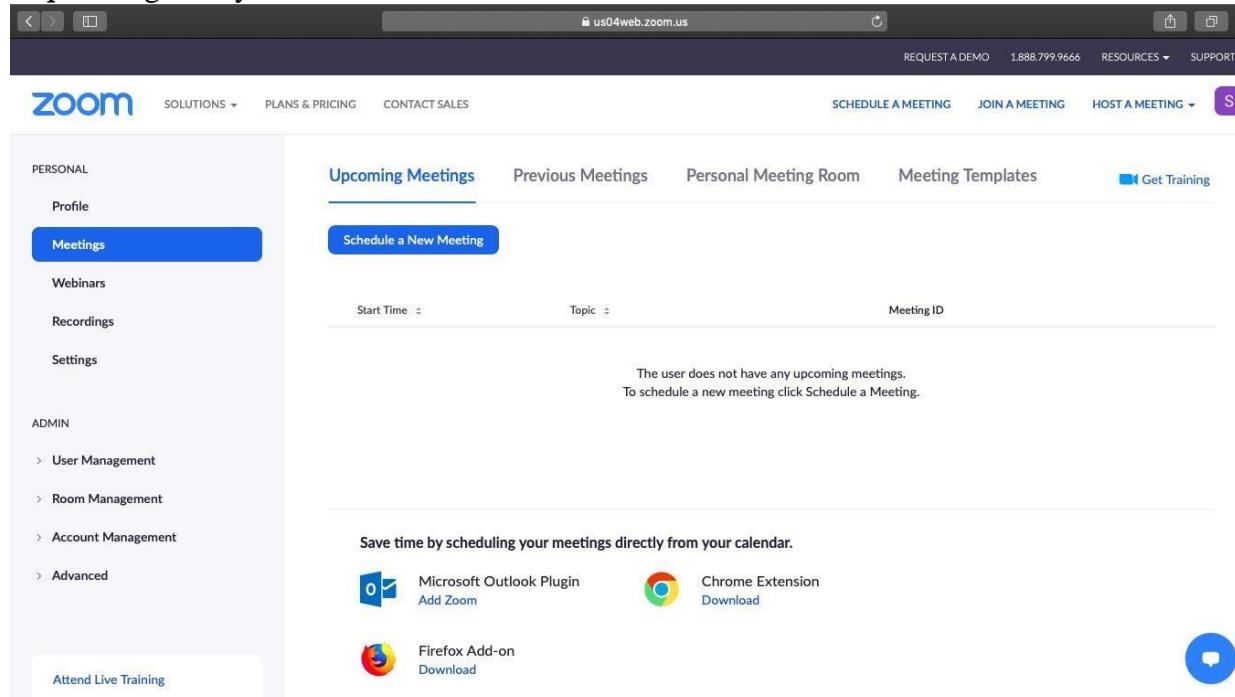
Language
 English ▾

[SIGN UP, IT'S FREE](#)

Step 4: Download the desktop app/Zoom client from the Zoom website for easy access.

B. How to set up a Zoom meeting
 Here's a step-by-step guide to set up a Zoom meeting easily:

A. Starting a Zoom meeting
Step 1: Log in to your Zoom account.



SCHEDULE A MEETING JOIN A MEETING HOST A MEETING S

PERSONAL

- Profile
- Meetings**
- Webinars
- Recordings
- Settings

ADMIN

- > User Management
- > Room Management
- > Account Management
- > Advanced

Upcoming Meetings Previous Meetings Personal Meeting Room Meeting Templates [Get Training](#)

[Schedule a New Meeting](#)

Start Time ▾ Topic ▾ Meeting ID

The user does not have any upcoming meetings.
 To schedule a new meeting click Schedule a Meeting.

Save time by scheduling your meetings directly from your calendar.

 Microsoft Outlook Plugin Add Zoom

 Chrome Extension Download

 Firefox Add-on Download

[Attend Live Training](#)

Step 2: Hover your cursor over the “HOST A MEETING” link at the top-right corner of the screen, and select one of the following options:

- With Video On
- With Video Off
- Screen Share Only

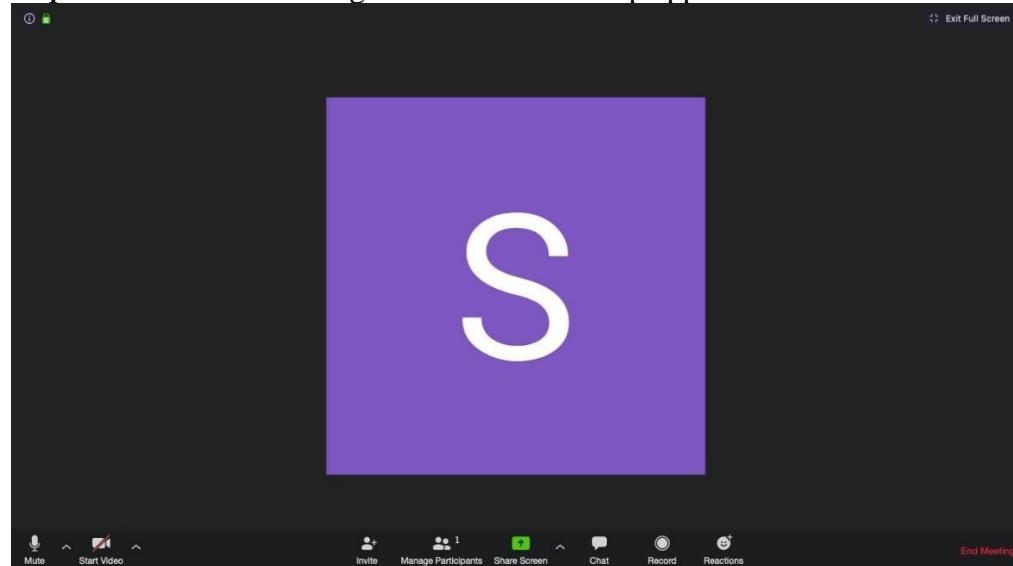
Step 3: The website will redirect you to the Zoom app and start a meeting. Here, you can edit meeting settings or copy the “Invitation URL” that you send to the attendees.



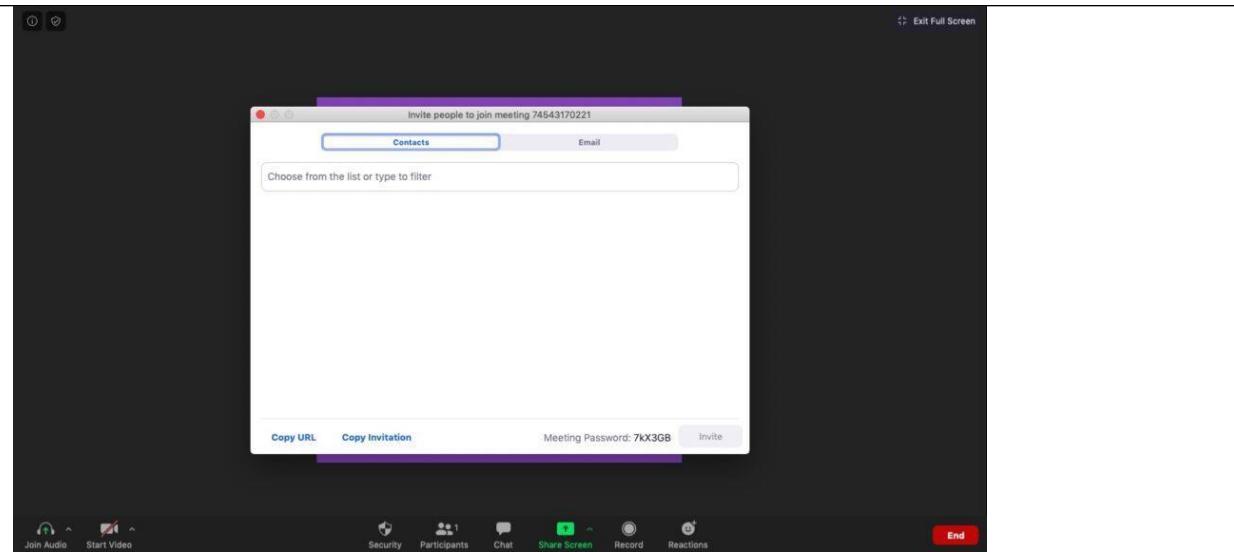
Note: You can also start a meeting quickly through the desktop app by following the instructions we list for mobile devices later on.

B. Adding participants

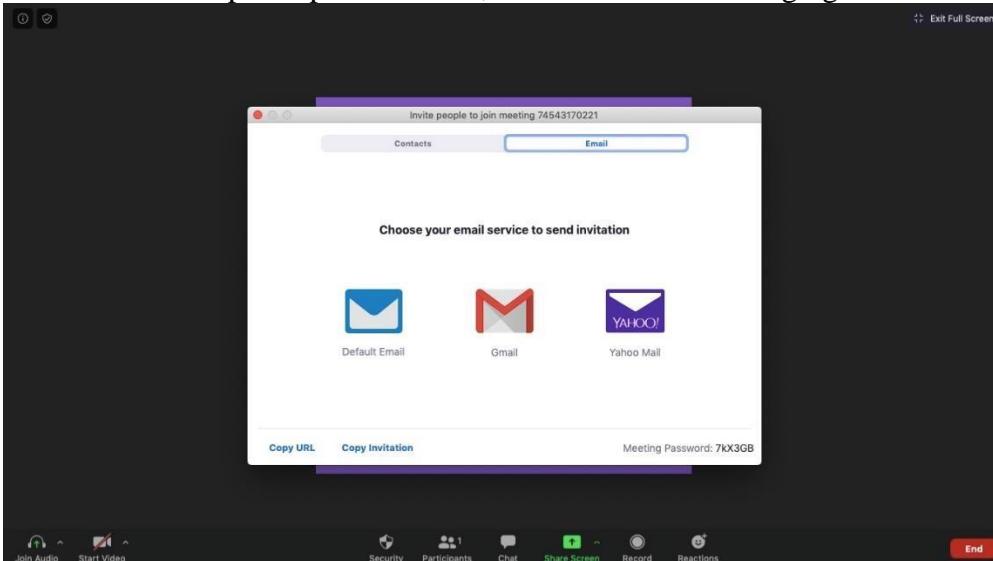
Step 1: Start a new meeting on the Zoom desktop app.



Step 2: In the new meeting screen, click on the “Invite” button in the toolbar at the bottom.



Step 3: Here, Zoom will give you the options to either “Copy URL” or “Copy Invitation”. You can send these to participants via text, email or instant messaging.



Step 4: You can also directly email the meeting details through your preferred email client via the Zoom app itself.

C. How to join a Zoom meeting

Here's a step-by-step guide to join a Zoom meeting quickly:

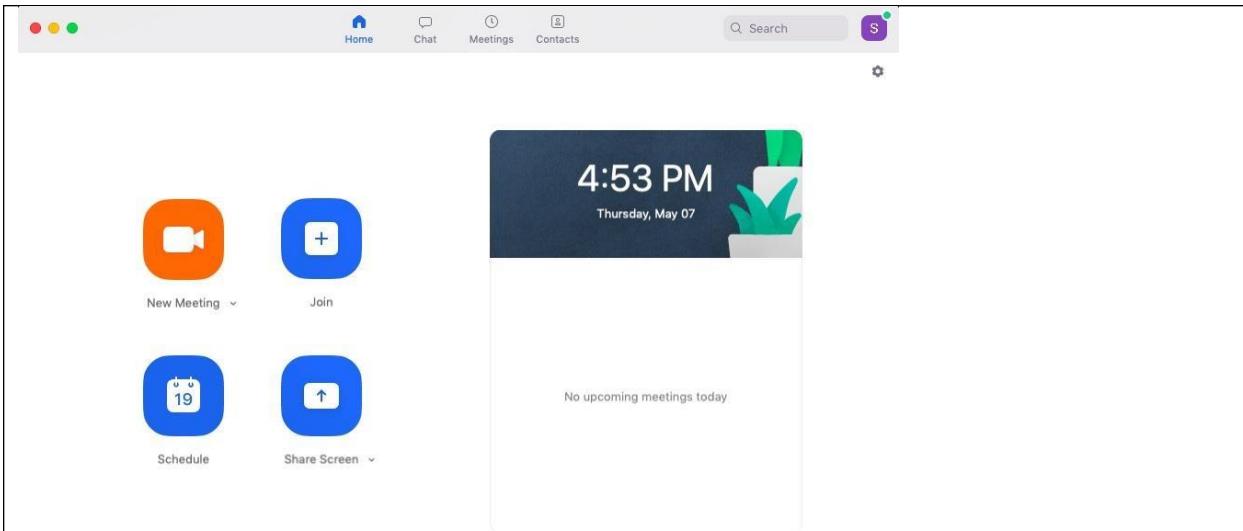
Note: The same steps apply to both your desktop and your phone.

A. Join using a Meeting Link

If you have a join link for a meeting, just click on it or paste it into your web browser to join the meeting.

B. Join using a meeting ID

Step 1: Open the Zoom app and click on the “Join” icon.



Step 2: Paste the Meeting ID in the box provided, add your display name for the meeting and click on the “Join” button.

Join Meeting

Meeting ID or Personal Link Name

Don't connect to audio

Turn off my video

You're now all set to communicate with your team members!

D. How to schedule meetings

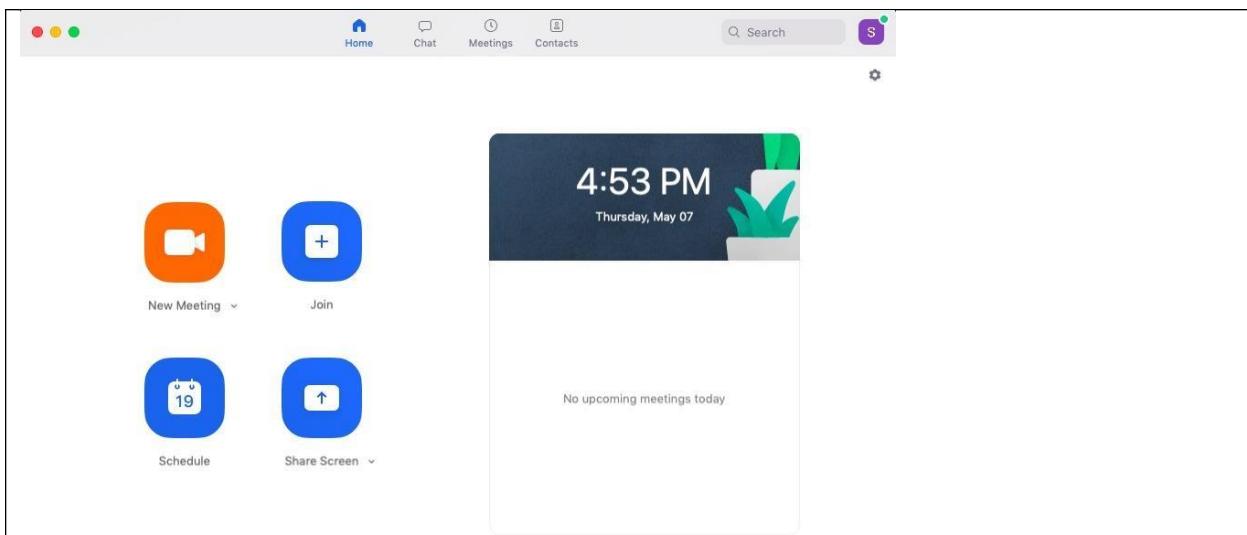
With a busy schedule, you can easily forget about appointments or upcoming business meetings. Luckily, Zoom lets you schedule meetings in advance to avoid this!

You can schedule a meeting by setting:

- Its date and time
- Meeting ID
- Whether it requires a password to join or not
- And more!

Here's a step-by-step guide to schedule meetings in Zoom easily:

Step 1: To schedule a meeting, head to the Zoom app and click on the blue “Schedule” button (looks like a calendar icon).



Step 2: Enter meeting details in the Schedule Meeting pop-up window that appears. You can set its date and time, privacy and access settings. You can also select your preferred calendar (between iCal, Google Calendar or others) to schedule the event in your calendar.

Schedule Meeting

Topic

Date
7/ 5/2020 5:00 PM to 7/ 5/2020 5:30 PM
 Recurring meeting Time Zone: Mumbai, Kolkata, New Delhi

Meeting ID
 Generate Automatically Personal Meeting ID 573-299-7884

Password
 Require meeting password 8xXS1P [?](#)

Video
Host On Off Participants On Off

Audio
 Telephone Computer Audio Telephone and Computer Audio
Dial in from United States [Edit](#)

[Cancel](#) [Schedule](#)

Quick Tip: Setting a meeting password can help avoid Zoombombing, which happens when someone who hasn't been invited to the meeting joins and disrupts it.

Step 3: Once you've adjusted preferences, click on the “Schedule” button at the bottom right of the screen.

E. How to record Zoom meetings

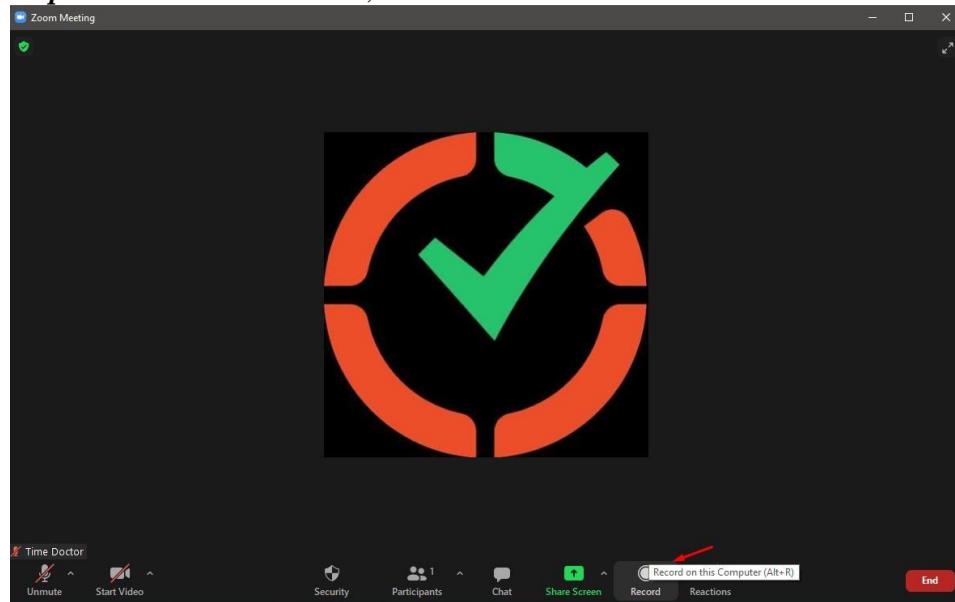
Recording a meeting lets you easily use it as a reference to document everything that was discussed. This is especially important for remote teams who use Zoom video conferencing as their key mode of communication.

Zoom allows you to record meetings easily and save them either to your local device or the Zoom cloud. By saving it to the Zoom cloud, your team members can access it across multiple platforms easily.

Here's how to record Zoom meetings:

Step 1: Start a meeting.

Step 2: In the Zoom toolbar, click on the “Record” icon.



Step 3: Choose between “Record on this Computer” or “Record to the Cloud.” This starts the recording, and all meeting members will see the word “Recording” in red at the top of the screen.

Step 4: Click on “Pause/Stop Recording” to stop recording the meeting. Alternatively, you can also end a meeting to stop recording it.

Step 5: After you end the meeting, Zoom converts the recording to MP4 format and stores it in your preferred location. You can now easily access your recorded sessions any time you want!

F. Screen sharing

Zoom lets you share your screen with other meeting participants easily.

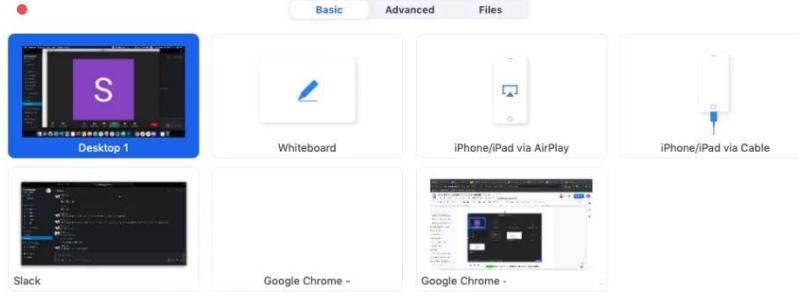
This lets you:

- Host virtual presentations and workshops.
- Explain processes in detail.
- Review work and project documents together with your team.

To share your screen, just click on the “Share Screen” icon in the toolbar at the bottom.

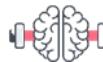
This lets you share:

- A specific app or window.
- A whiteboard.
- Apple iPhone / iPad screen (if your device supports this).



If you want more screen sharing options, click on the “Advanced” tab at the top of your screen. Here, you can choose to share:

- A part of your screen or the entire screen.
- Only your computer’s sound or your microphone’s sound as well.
- Content from a second camera or content only from your screen.



Self-check 4-1:

Dear Students! So far, We have discussed Online communication and collaboration and Fundamental components the platforms. Now it is your turn to answer the following self-check questions

1. What is the primary purpose of using email in web development?
2. List two examples of messaging apps and their uses in web development.
3. How can video conferencing tools benefit remote web development teams?
4. Name one social media platform useful for professional networking and describe its benefit.
5. Why is effective communication important in web development projects?

Dear Students! Have you answered the self-check questions? If yes, let us move on to work on a collaborative project and summarize this unit.

Unit Summary

In this unit, we explored the fundamental aspects of online communication and collaboration, highlighting their significance in web development. You learned about various communication tools, including email, messaging apps, video conferencing, and social media platforms, each playing a crucial role in facilitating effective communication among team members, clients, and stakeholders. These tools help streamline workflows, enhance problem-solving, and promote innovation through diverse perspectives.

We also delved into common collaboration tools like Google Workspace, Microsoft 365, Slack, and GitHub. These tools are indispensable for web developers, enabling real-time collaboration, efficient project management, and seamless integration of workflows. Google Workspace and Microsoft 365 provide comprehensive suites of productivity applications, while Slack and GitHub focus on communication and version control, respectively. Understanding these tools ensures you can effectively contribute to and manage collaborative web development projects.

Unit Review Questions

Part I: Multiple Choice Questions

1. Which tool is primarily used for professional communication and project updates?
 - A. Messaging Apps
 - B. Email
 - C. Social Media Platforms
 - D. Video Conferencing
2. Which platform is best for real-time document collaboration?
 - A. Slack
 - B. GitHub
 - C. Google Docs
 - D. Facebook
3. Which feature in email helps manage communication transparency?
 - A. Cc
 - B. Bcc
 - C. Subject line
 - D. Signature

4. What is the main purpose of an out-of-office autoreply?
 - A. To notify senders of your absence
 - B. To block spam emails
 - C. To send automated marketing emails
 - D. To forward emails to another address

Part II: Short Answer Questions

Instruction: Give brief answer for the following review questions:

1. What are the primary benefits of using video conferencing tools for web developers?
2. How can GitHub be used by web developers for collaboration?
3. What strategies can be employed to prevent cyberbullying and online harassment?
4. Describe the role of email in project management for web developers.

Answer key for Self-check questions

Self-Check 4-1: Online Communication and Collaboration

1. The primary purpose of using email in web development is to facilitate formal communication with clients, team members, and stakeholders, share project updates, send documents, and maintain a written record of communications.
2. Examples include WhatsApp, used for quick exchanges and group chats for team discussions, and Telegram, which supports organized conversations in channels and file sharing.
3. Video conferencing tools like Zoom and Microsoft Teams enable virtual meetings, allowing teams to collaborate in real-time, conduct project presentations, and engage in face-to-face communication, essential for remote collaboration.
4. LinkedIn is useful for professional networking as it helps web developers connect with other professionals, find job opportunities, and showcase their work.
5. Effective communication ensures that all team members share a common understanding of project goals, facilitates quick problem-solving, streamlines workflows, and fosters innovation through diverse perspectives.

UNIT 5

SAFE ONLINE ENVIRONMENT

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 5.1. Overview of Security and Privacy
- 5.2. Potential threats to computers, devices, and data.
- 5.3. Ways to protect computers, devices, and data

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Create a safe online environment

Unit Overview

In this unit, we delve into the essential aspects of creating and maintaining a safe online environment. As the digital world continues to expand, understanding and implementing effective security and privacy measures has become crucial. This unit will guide you through the complexities of digital security, emphasizing the importance of protecting personal information and data from various threats. You will explore potential risks to computers, devices, and data, and learn practical strategies to mitigate these risks. By the end of this unit, you will be equipped with the knowledge and skills necessary to secure your digital presence and safeguard your data, ensuring a safer and more trustworthy online experience.

Key Terms: *Security, Privacy, Digital Identity, Digital Footprint, Threats, Malware, Cybercrime, Social Engineering, Identity Theft*

5.1. Overview of Security and Privacy

Discussion



Dear students: Before we proceed to topics lets discuss in a small group on the following points

- ❖ What is Digital Identity?
- ❖ What are the Reasons for Protecting Personal Information?

Security refers to the measures and practices that protect data, systems, and resources from unauthorized access, damage, disruption, or theft. It encompasses strategies and tools to ensure the confidentiality, integrity, and availability of information and resources, often involving authentication, encryption, firewalls, and other protective mechanisms.

Privacy is the right and practice of controlling personal information and data, determining who has access to it, and how it is used. It involves protecting individuals from unwarranted intrusion into their personal lives, ensuring that data collection and processing are transparent, and obtaining consent for the collection and use of personal information.

5.1.1. Reasons for Protecting Personal Information

Nowadays, more and more people are using the Internet and mobile devices for online shopping, banking, business, communication and other activities. Some companies rely on various cloud services and other web-based services to run their day-to-day business. Making information easier to access through the Internet also exposes businesses to some security issues. Hackers are able to take advantage of vulnerabilities in the transmission of data online to gain unauthorized access to systems and networks. There have been many reports of data breaches and identity theft in the past few years around the world. Cybercriminals often steal personal information such as banking records, credit card details, usernames and passwords for financial gain.

Personal Information is most often used by companies to identify and authorize users who transact business on their websites. For example, an online shopping site may have a record of a user's name, address, credit card details, etc . Hackers may steal this information in order to impersonate a user and then conduct fraudulent and unauthorized transactions and other fraudulent activities. Without adequate security and protection of personal information, users are exposed to Internet based crimes such as identity theft and fraud and loss of privacy. Companies which do not protect their user's personal information may lose customers' trust - and their business.

In summary, security is a broader concept that encompasses measures to protect data and systems from various threats, while privacy is more focused on the individual's control over their personal information and how organizations handle and safeguard that data. Both are crucial in our increasingly digital and interconnected world, especially in the context of data breaches, cyberattacks, and the growing concerns about the use of personal data by companies and governments.

5.1.2. Overview of Digital Identity

In today's digital age, individuals have a digital identity that is a reflection of their online presence and activities. Digital identity comprises various components that together form a person's or an entity's virtual representation. Understanding these components is crucial for educational leaders to navigate the digital landscape effectively and protect their online identity. Here are the key components of digital identity:

- 1. Personal Information:** This component includes basic information such as your name, date of birth, contact details, and other personally identifiable information (PII). Educational leaders should be cautious about sharing sensitive PII online and be aware of potential privacy risks.
- 2. Usernames and Account Credentials:** Usernames and passwords are essential for accessing online services and platforms. They are a part of your digital identity. Using strong, unique passwords and implementing two-factor authentication (2FA) is crucial for safeguarding your online accounts.
- 3. Online Profiles:** Your presence on social media platforms, professional networks, and educational websites contributes to your digital identity. Academic leaders should manage and curate their online profiles, ensuring that the information presented aligns with their professional image.
- 4. Digital Footprint:** The digital footprint consists of data generated by your online activities, including social media posts, comments, online searches, and website visits. Educational leaders should be mindful of their digital footprint, as it can influence their reputation and credibility.
- 5. Digital Signatures and Certificates:** Digital signatures and certificates are cryptographic methods used for verifying the authenticity and integrity of digital documents and communications. In the context of academic leadership, digital signatures can be used to authenticate important documents and communications.
- 6. Interactions and Communications:** Interactions with others, including email communications, messages on social media, and comments on educational forums, contribute to your digital identity. The tone, language, and professionalism of these interactions reflect on your online presence.
- 7. Digital Skills and Achievements:** Your digital identity can be enhanced by showcasing your digital skills, certifications, and achievements online. Academic leaders can use online platforms to highlight their expertise and professional growth.

Understanding and managing these components of digital identity is crucial for leaders to maintain a positive and professional online presence. They should be mindful of the information they share and be proactive in protecting their digital identity from potential threats and privacy risks. Additionally, promoting responsible digital citizenship and privacy practices within educational institutions and industries is essential for fostering a safe online environment.



Self-check 5-1:

Dear Students! So far, We have discussed what Security and Privacy are and the Reasons for Protecting Personal Information. Now it is your turn to answer the following self-check questions

1. Why is regular software updating important for security?
2. Define privacy in the context of online data.
3. What are two key components of digital identity?

Dear Students! Have you answered the self-check questions? If yes, let us move on to work on a collaborative project and summarize this unit.

5.2. Potential threats To Computers, Devices and Data

Maintaining data security is vital for individuals, small businesses and large corporations. Ensuring that data is kept secure is essential in avoiding disaster, both personally and professionally, but unfortunately it can be a difficult task due to malicious or unintentional behavior. There are many potential threats to computers, devices and data, including malware, the risk of unauthorized access, theft, and accidental damage. It is important to understand these threats so that you can plan appropriately to reduce the likelihood of them affecting you.

5.2.1. Malware

This is malicious software typically designed to install itself on a computer or device without consent in order to damage files or steal information. Malware is an umbrella term used to refer to viruses, worms, Trojans, spyware, adware, and other forms of malicious code. Installed without the user's consent, malware is designed to disrupt computer operation, collect sensitive information, or gain unauthorized access to a computer, device, or network.

The main types of malwares include:

- **Virus:** An intrusive program that infects computer files and may cause damage.
- **Worm:** Self-replicating malware that uses a computer network to send copies of itself to other computers.
- **Trojan:** Destructive program that masquerades as an application.

Spyware: Malware that collects information on user browser habits without their consent.

5.2.2. Unauthorized access

Accessing computers, devices or data without permission, whether the intention is malicious or not, may have negative consequences such as the alteration, loss or theft of data.

5.2.3. Accidental damage

Damage from accidents such as breakages, fires or floods may result in the destruction, alteration or loss of computers, devices and data.

5.2.4. Cybercrime

An offence that involves using the Internet or a computer to carry out illegal activities, often for financial or personal gain. Examples include **social engineering** and **identity theft**.

A. Social Engineering

Social engineering is a way to manipulate or influence people with the goal to illegally obtain sensitive data (for example, passwords or credit card information). Social engineers research and learn about the personal environment of their target and fake their identity to obtain confidential information from the victim. In most cases, they infiltrate third-party computer systems to spy on sensitive data.

Methods of Social Engineering

- **Phone calls:** One of the most common methods social engineers use in their attacks is conducted via the phone. The attacker may impersonate a person of authority, a person representing a person of authority or a service provider to extract information from an unsuspecting user. For example, a person claiming to be the CEO of the company calls someone on the helpdesk, requesting for his password, which he claims to have forgotten.
- **Phishing:** A type of social engineering attack wherein the perpetrator sends an e-mail that appears to come from a legitimate source (for example, a bank). The e-mail usually requests for verification of information, sometimes warning of dire consequences if the recipient fails to comply. A phishing e-mail usually includes links to fraudulent web pages which are made to look very similar to legitimate web pages, including logos and content.
- **Pharming** In a pharming scam, a victim's computer or server is infected with malicious code that re-directs them to bogus websites. It is similar to Phishing in that it uses fake or spoofed websites to collect confidential data. However, in Pharming, the victim is re-directed to a bogus site even if they have entered the correct web

address.

One way a pharming attack is done is by using DNS poisoning. In a DNS poisoning attack, the domain name system table in a server is modified so that users are automatically redirected to fraudulent sites.

Shoulder Surfing: This includes direct observation techniques, such as looking over someone's shoulder, to get information. It is commonly used to obtain passwords, ATM PINs and security codes.

Phishing Examples :

1. Email Phishing:

Example: You receive an email that appears to be from your bank, stating that there's been suspicious activity on your account. The email includes a link to a login page that looks like your bank's website.

Case: In 2017, a widespread phishing campaign targeted Gmail users. Attackers sent seemingly legitimate Google Docs invitations, leading recipients to a fake login page. Many individuals fell victim to this scheme.

2. Spear Phishing:

Example: An attacker researches a specific individual, perhaps an academic leader at a university, and sends an email pretending to be from a trusted colleague. The email requests confidential information.

Case: The 2016 spear-phishing attack on the Democratic National Committee (DNC) is a notable case. Hackers sent emails posing as colleagues to DNC members, leading to the compromise of sensitive political data.

3. Vishing (Voice Phishing):

Example: You receive a phone call from a purported tech support agent who claims that your computer is infected. They ask you to install remote access software to "fix" the issue, granting them control of your device.

Case: The "Microsoft Tech Support" scam is a common example of vishing. Scammers impersonate Microsoft technicians, attempting to gain access to victims' computers or steal personal and financial information.

Brown, L. A., & Wilson, S. P. (2019). Social engineering techniques and countermeasures in the digital age. Computers & Security, 18(3), 125-140.

B. Identify Theft

Identity theft is when someone deliberately impersonates and uses another person's identity. This is usually done for financial gain or to obtain credit and/or other benefits using someone else's name: for example, when someone uses another person's identity to obtain a driver's license. This type of fraud could have a devastating effect on the person whose identity has been assumed.

An initial implication of identity theft is the amount of time and money needed to re-establish your identity and credit history and to clear your name.

Methods of Identity Theft

- **Information Diving:** Also known as Dumpster Diving, it is a method of obtaining personal or private information by digging through a dumpster or trash bin for discarded documents or material such as utility bills or credit card statements.
- **Skimming:** Identity thieves use skimming as a method of capturing a victim's personal data by using a small electronic device. A skimmer is a device that is usually attached to an ATM machine's card slot. A victim may unwittingly slide his card into the skimmer, which then reads and stores all the information from the card's magnetic strip (Davis & Garcia, 2018).
- **Pretexting:** This involves creating and using an invented scenario (the pretext) to engage a targeted victim. The pretext increases the chance the victim will reveal information or perform actions that would be unlikely in ordinary circumstances – for example, someone pretending to be from a company that provides you with a service might persuade you to share your bank account details with them.



Discussion

Dear students Discuss Different kinds of attacks or threats you have experienced locally in your environment or privately.



Self-check 4-1:

Dear Students! So far, we have explored the different Threats to Computers, Devices and Data.

Now it is your turn to answer the following self-check questions.

1. What is malware and how can it affect your computer?
2. Describe the difference between a virus and a worm.
3. What is phishing and how can you recognize it?
4. What is identity theft and what are its potential consequences?

Dear Students! Have you answered the self-check questions? If yes, let us move and explore Protecting Mechanisms of computers, devices and data

5.3. Ways to protect computers, devices and data

Discussion



- How we can protect our devices and data from authorized access?
- List different mechanisms you use to protect yourself online?

There are many actions you can take to protect computers, devices and data from threats:

5.3.1. Use anti-virus software

Anti-virus software is used to scan files to identify and eliminate malware. There are numerous free and paid anti-virus software packages available. Norton, McAfee, windows defender, Kaspersky, AVG and Avast Antivirus are the popular antivirus. But the popularity and effectiveness of antivirus software can change over time, It is important to research the available options . This are some tips when you use Anti- virus software:

- Use anti-virus software to detect viruses and stop them from being installed or removing them.
- Regularly update anti-virus, application and operating system software to detect new viruses, fix known problems and security risks.
- You can use antivirus software to scan external drives and downloaded software.

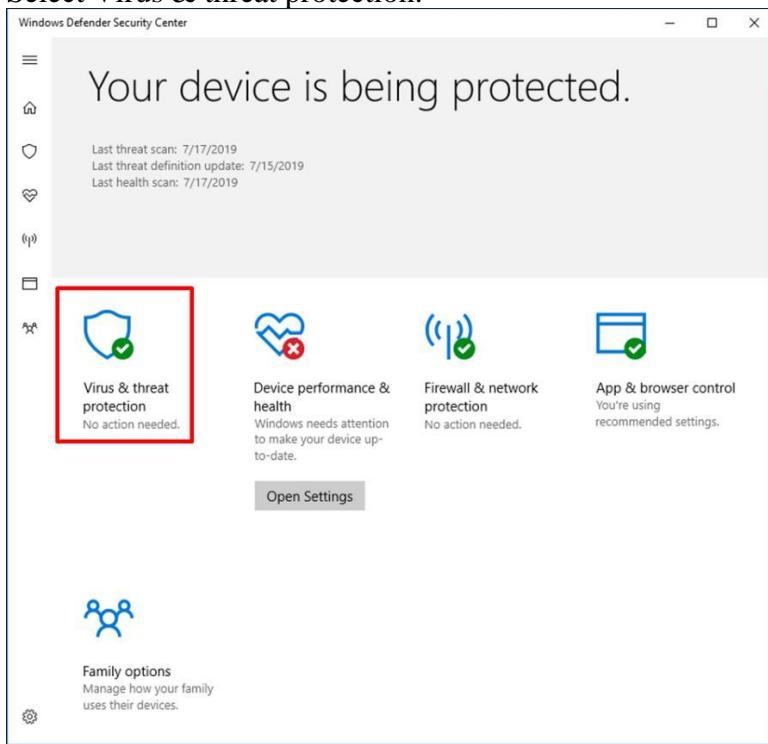
An example of anti-virus software is Windows Defender. Windows Defender software is provided with Windows 10. Windows Defender Security Center is a central location to view status and manage security features, such as firewall and antivirus. You can choose to perform different types of scans:

- **Quick scan** – This searches for viruses in locations where they are most likely to occur.
- **Full scan** – This searches the entire computer for viruses but will be slower and may affect your computer's performance more when it is running.
- **Custom Scan** – This allows you to choose which files and location you want to check.

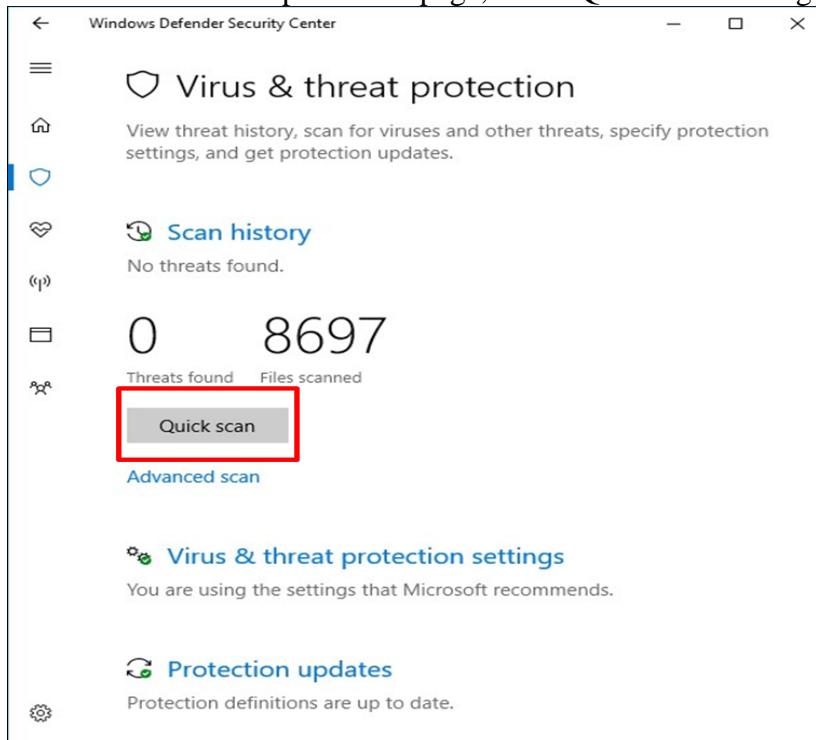


Steps: To perform a quick scan in Windows Defender Security Center:

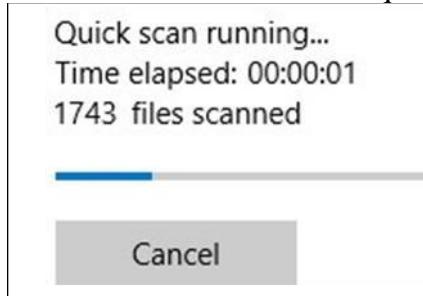
1. Open Windows Defender by clicking the search tool on the taskbar and entering Windows Defender.
2. Select Windows Defender Security Center.
3. Select Virus & threat protection.



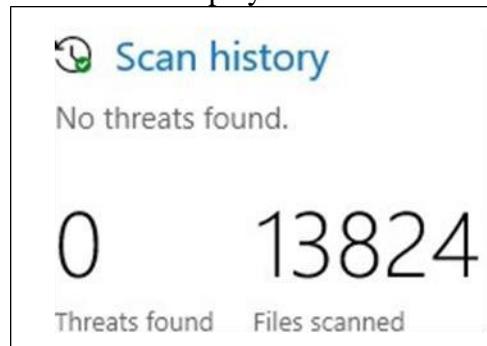
4. In the Virus & threat protection page, select Quick Scan to begin a quick scan.



5. Windows Defender runs the quick scan.



The results are displayed when the scan is finished.



5.3.2. Regularly update Software

Software developers often release software updates to address problems or fix security vulnerabilities in their products. These updates help prevent or fix problems or enhance and

improve how your computer works. Therefore, it is important to regularly check for updates to your software, such as your anti-virus, application and operating system software.

Issues associated with failing to keep your software up to date may include.

- Vulnerability to viruses, spyware and other malware.
- Software crashing, freezing, or performing poorly.

As well as resolving security issues, software updates not only improve security but may also contain improvements and new features.

5.3.3. Don't use Unknown sources

Do not download programs from unknown sources

Do not download programs, open attachments or links from unknown sources to minimize the threat of malware being installed.

If you download software from the Internet, make sure it is from a reliable source. Scan downloads with antivirus software upon completion. During the installation process, read all prompts about what the program is putting on your computer. USB drives, or other drives, may be infected with a virus. You can use antivirus software to scan external drives.

Do not open email attachments or links from unknown sources

Your computer can become infected if you open an email attachment, that contains malicious code. Even if the message is from someone you know, always use caution before opening a link or downloading an attachment. In general, do not open email attachments you were not expecting to receive.

5.3.4. Use Strong passwords



Discussion

- How strong is your password?
- Do you think using the same password for every platform a good practice?

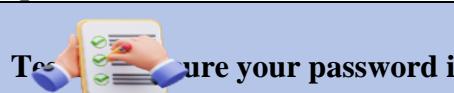
Use strong passwords to prevent unauthorized access. Passwords should be of adequate length and character mix, kept private and changed regularly. Password policies are guidelines or requirements on the creation and use of passwords. These passwords can be required for access to

computer systems, applications, shared files, networks, online forums, online shopping accounts and services, cloud storage, social media and email accounts etc.

The following are some guidelines for password policies.

- **Create passwords with adequate length:** While the word adequate may seem quite vague in this context, it is important to generate a password that is usually over 8 characters to reduce the chances of someone figuring it out. Many websites have minimum password length requirements.
- **Create passwords with adequate character mix:** Passwords that use a mix of caps characters and numbers are stronger than passwords that only use lowercase letters. Like password length, having a password that has character mix, like starting with a capital letter and including numbers, can be a requirement put in place by certain websites. For example, a password such as “fireworks” can be made stronger by varying the character style. If you capitalize the first letter, and swap the i and o with the numbers 1 and 0 you get “F1rew0rks” which is less likely to be solved.
- **Avoid using passwords that include your personal information,** such as your name, birthdate, or spouse name.
- **Avoid words found in the dictionary.**
- **Do not share passwords:** This may seem an obvious point, but disclosing your password to other people, even if they are trusted colleagues, friends or family, is not recommended. If you need to share a password, make sure it is with someone who will not take advantage of your trust or end up revealing it to a third party.
- **Change regularly:** It is recommended that you change your passwords from time-to-time. This can help prevent someone who has gained knowledge of your password from monitoring your activities over an extended period of time.

Operation Sheet 5-1:



Title: Secure password testing
Objective: To test how our passwords are strong
Instructions:

- Go to <http://howsecureismypassword.net/> to test how secure your password is:
- Tested it with different sample passwords of your own or write “password”.

HOW SECURE IS MY PASSWORD?



Common Password: In The Top 10 Most Used Passwords

Your password is very commonly used. It would be cracked almost instantly.

Possibly A Word

Your password looks like it could be a dictionary word or a name. If it's a name with personal significance it might be easy to guess. If it's a dictionary word it could be cracked very quickly.

Character Variety: Just Letters

Your password only contains letters. Adding numbers and symbols can make your password more secure.

5.3.5. Use Encryption

Encryption is a process that encodes data or information so that only authorized users can read the information. In the event that the data is intercepted by an unauthorized user, the interceptor will need to decrypt the data first before he is able to read it. Encryption is used to protect data as it's passed over a network. It is often used to prevent illegal access to or reproduction of information.

Decryption is the process of changing encrypted data back into its original form, so that it can be read. Decryption uses a key, or algorithm, to “unlock” the encrypted data. A common use of encryption is when data is sent through a secure web connection.



Steps: To encrypt Microsoft Word document with password

1. On Microsoft word Click the FILE tab.
2. Click Info.
3. Click Protect Document.

Info



Convert

Compatibility Mode

Some new features are disabled to prevent problems when working with previous versions of Office. Converting this file will enable these features, but may result in layout changes.



Protect Document

Protect Document

Control what types of changes people can make to this document.



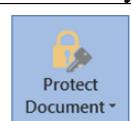
Check for Issues

Inspect Document

Before publishing this file, be aware that it contains:

- Document properties and author's name

4. Click Encrypt with Password.



Protect Document

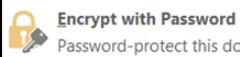
Protect Document

Control what types of changes people can make to this document.



Mark as Final

Let readers know the document is final
and make it read-only.



Encrypt with Password

Password-protect this document



Restrict Editing

Control the types of changes others
can make



Restrict Access

Grant people access while removing
their ability to edit, copy, or print.



Add a Digital Signature

Ensure the integrity of the document
by adding an invisible digital signature

5. In the Encrypt Document dialog box, type a password in the Password box.



6. Click OK.
7. In the Confirm Password dialog box, type the password again in the Reenter password box, and then click OK.



8. Save the file.

5.3.6. Use safe browser settings

There are some important web browser settings to consider. Customizing these settings according to your preferences and security requirements can help you optimize your browsing experience while safeguarding your privacy and data. Important web browser settings can enhance your online experience, improve security, and protect your privacy. Here are some key web browser settings to consider:

1. Privacy Settings:

- **Cookies:** Configure cookie settings to manage how websites store and access data on your device. You can choose to block all cookies, allow cookies from visited sites, or create exceptions.
- **Clear Browsing Data:** Regularly clear your browsing history, cookies, cache, and site data to maintain privacy and free up storage space.

2. Security Settings:

- **Pop-ups:** Block pop-up windows to prevent unwanted ads and potential security threats.
- **Safe Browsing:** Enable safe browsing to receive warnings when visiting potentially harmful or deceptive websites.

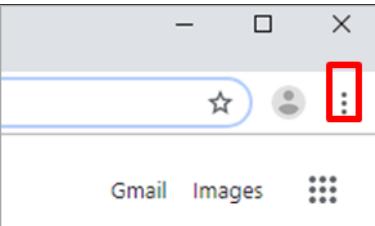
3. Password and Autocomplete Settings:

- **Password Manager:** Use a built-in or third-party password manager to securely store and autocomplete your login credentials.
- **Autocomplete:** Configure autocomplete settings for forms and payment information while ensuring the security of your data.

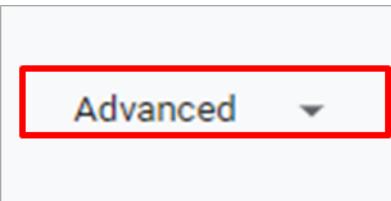
Managing Cookies

A cookie is a text file stored on a user's computer that monitors their web habits. Cookies are small pieces of data that are sent by websites to your browser when you are browsing. They have different functions, including recording of browsing activity. Their use raises privacy concerns, but they can also make navigating the Web easier. Note that allowing sites to save and read cookie data is the recommended setting in Google Chrome and is the default setting.

|  | Steps: To allow / block cookies: |
|---|---|
| | <ol style="list-style-type: none">1. Click the Google Chrome icon  on the taskbar to open the web browser.2. Click the More button |



3. Select **Settings**
4. Scroll to the end of the Settings page and click **Advanced**.



5. Under **Privacy and Security**, click **Site Settings**.
6. Click **Cookies** in the Site Settings window.
7. To block Cookies, turn the Allow sites to save and read cookie data (recommended) slider to Off.



8. To allow cookies, turn the Blocked slider to On. Allow sites to save and read cookie data (recommended) is displayed.

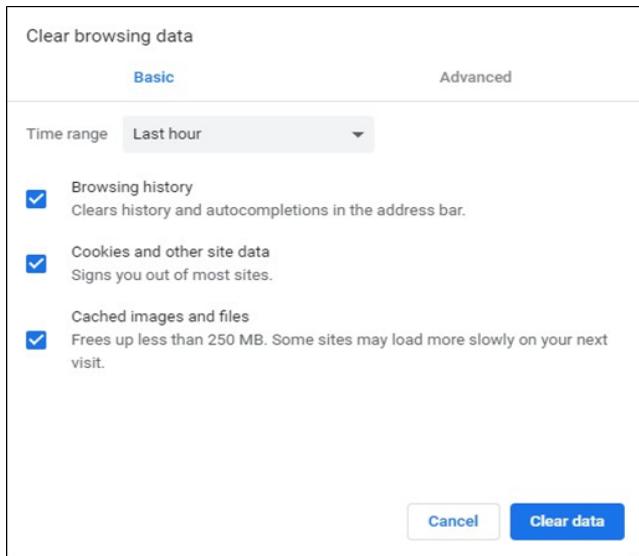


9. Click X to close the Settings tab.

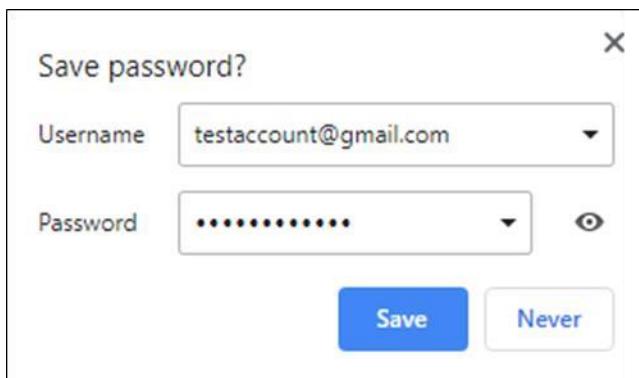
Managing Browsing Data

Most web browsers automatically remember visited websites. The list of visited websites is stored in your browser history. To view your history in Google Chrome, open the more icon and mouse over History, a list of the website visited by the browser will display in a sub menu.

In addition to history, your web browser also automatically saves temporary Internet files. This data may help your browser to load previously visited webpages more quickly. However, if there is a lot of saved data this may slow down the performance of your browser.



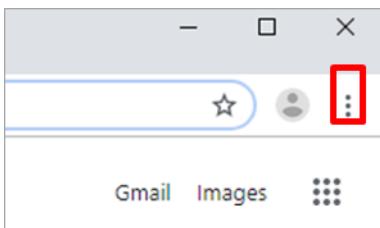
Web browsers can remember passwords and personal information such as email addresses, phone numbers etc. This can be very useful as it can speed up the process of completing forms and signing in to online accounts. Google Chrome asks if you want to save passwords, however by default, it remembers data when you fill out forms. You may not want your web browser to remember your personal information, especially on a shared computer.



Steps: To delete your browser history, temporary Internet files, saved form data and saved passwords:

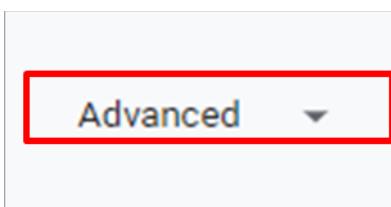
1. Click the **Google Chrome** icon  on the taskbar to open the web browser.

2. Click the More button



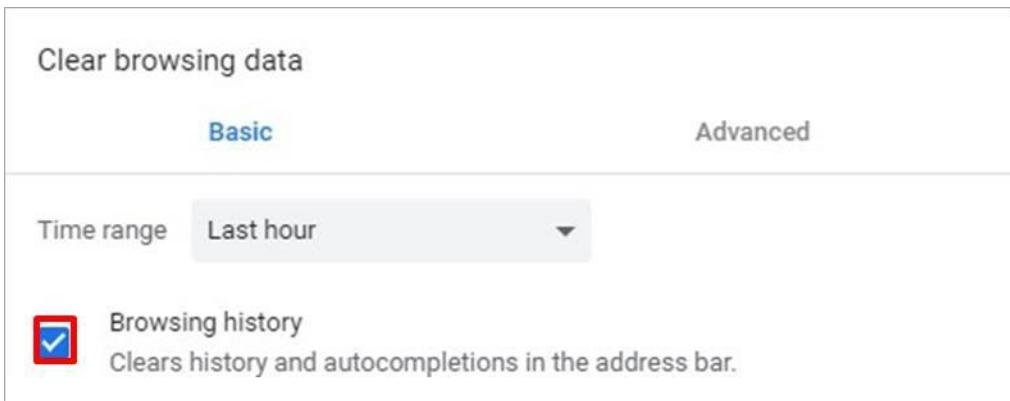
3. Select **Settings**

4. Scroll to the end of the Settings page and click **Advanced**.

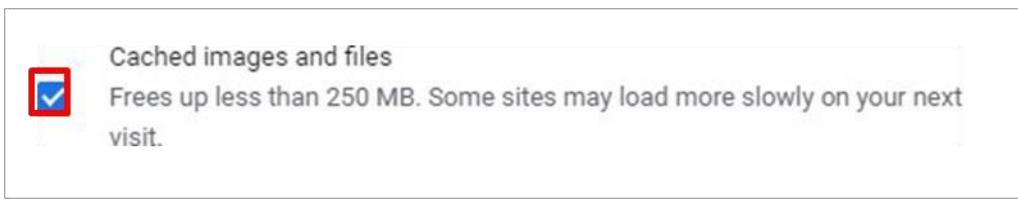


5. Under **Privacy and Security**, click **Clear Browsing Data**.

6. To delete browsing history; if necessary, check the Browsing History checkbox in the Clear browsing data dialog box.



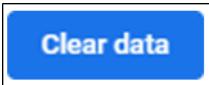
7. To delete temporary Internet files, if necessary, check the Cached images and files checkbox.



8. Select the time range from the Time range drop-down list. For example, select All Time.



Advanced

9. To delete saved form data and saved passwords, click the **Advanced tab**, near the top of the dialog box.
10. To delete saved form data, **check the Auto-fill form data checkbox**.
11. To delete saved passwords, check the **Passwords** and other sign in data checkbox. Select the time range from the Time range drop-down list. For example, select All Time to fully clear passwords and data.
12. Click the **Clear data button**.
- 
- Clear data
13. The browsing history, temporary internet files, all saved passwords and form data history are cleared for the time range set.
14. Click **X** to close the Settings tab.

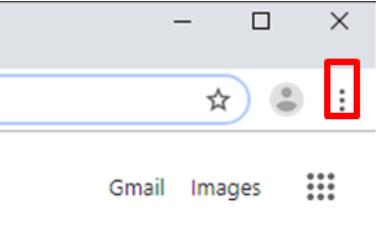
•Managing Pop-Ups

Pop-ups are new unrequested browser windows that open automatically over the current web page, they usually contain advertisements. Such pop-ups may relate to gaming or explicit content and are generally both unnecessary and unwanted. Some pop-ups may be dangerous to your computer; others are not harmful at all. Blocking pop-ups is recommended. In the Google Chrome browser, pop-ups are blocked by default. You may, however, add and remove specific web pages to a list of allowed or blocked websites.

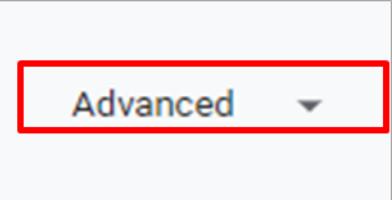


Steps: To delete your browser history, temporary Internet files, saved form data and saved passwords:

- 
1. Click the **Google Chrome** icon on the taskbar to open the web browser.
 2. Click the More button



3. Select **Settings**
4. Scroll to the end of the Settings page and click **Advanced**.



5. Under **Privacy and Security**, click **Site Settings**.
6. In the Site Settings window, under **Permissions**, click **Pop-ups and redirects**.
7. To block Pop-ups, turn the **Blocked slider** to **Off** in the Pop-ups and redirects options.



8. To allow Pop-ups, turn the Blocked slider to On.



1. Click X to close the Settings tab

5.3.7. Privacy Control in social media accounts



Discussion

- Do you know what your current privacy settings are on each social media Platform?
- Do you think everything what you will do on those platforms should be public?

Privacy settings refer to the configurable options within various digital platforms, applications, devices, or services that allow users to control and manage the visibility and accessibility of their

personal information, data, and online activities. These settings enable users to make informed decisions about who can access their data, what information is shared, and how it is used. An important way of protecting yourself when using a social network is to apply appropriate privacy settings. These settings will typically allow you to control who sees your profile and who you interact with. For example, Facebook allows us to manage the privacy of our postings and links to apps.



Steps: To Adjust your privacy settings on Facebook.

1. Click your profile picture in the top right of Facebook.
2. Select Settings & privacy, then click Settings.
3. Click Privacy in the left column.

Operation Sheet 5-2:

In the following activity you will Review Facebook privacy policies and manage your privacy settings in the Privacy Center.



Review and manage your privacy settings on Facebook:

Title: Review and manage your privacy settings on Facebook

Objective: To review the deferent privacy settings on Facebook

Instructions:

- Walk through Privacy Checkup to make sure you're sharing your information with who you want.
- Customize your Ad preferences to see ads that are more relevant to you.
- Manage some of the content you share on Facebook, like posts, photos and videos, from the Manage Activity section of your activity log.
- Learn how to manage what others can see about you.

5.3.8. Two-Factor Authentication (2FA):

Two-Factor Authentication (2FA) is a security process that requires users to provide two different authentication factors to verify their identity. These factors typically include something the user knows (e.g., a password) and something the user has (e.g., a mobile device or security token). 2FA adds an extra layer of security to prevent unauthorized access even if the password is compromised. Different social media platforms offer 2FA to enhance the security of your account. Here are the steps to set up 2FA using Telegram:

Operation Sheet 5-3:



Two-Factor Authentication Setup with Telegram:

Activity Title: Two-Factor Authentication Setup with Telegram

Objective: To guide users through the process of setting up Two-Factor Authentication (2FA) on their Telegram accounts for enhanced security.

Activity Steps:

1. Accessing Account Settings:

- open the Telegram app on their mobile devices.
- Navigate to the menu by tapping on the three horizontal lines in the top-left corner.

2. Opening Account Settings:

- In the menu, select "Settings."

3. Locating Privacy and Security:

- Within "Settings," find and tap on "Privacy and Security."

4. Initiating Two-Step Verification:

- In "Privacy and Security," locate and tap on "Two-Step Verification."

5. Setting a Password:

- tap on "Set Additional Password" to create 2FA password.
- Emphasize the importance of choosing a strong and memorable password.

6. Adding a Hint (Optional):

- add a hint for their 2FA password. However, not make it too explicit.

7. Adding a Recovery Email (Optional):

- add a recovery email for account recovery if you forget your 2FA password.

8. Verifying the Password:

- enter the newly created password to verify it.

9. Confirmation and Completion:

- Once the password is verified, confirm that 2FA is successfully set up on their Telegram account.

10. Testing the Two-Step Verification:

- log out of your Telegram account and attempt to log back in. They should experience the 2FA process by entering the password they just set.

5.3.9. Regularly Back Up Data to Remote Location:

Regularly back up data to a remote location to keep it safe in the event of problems such as network, hardware or software problems, theft or accidental damage (Jones & Patel, 2018). It is important to make regular backups as your files can be lost or destroyed accidentally; the hard disk may also develop problems. Thus, backups must be made to an external device such as CDs, DVDs, external hard disks or USB drives.

Another option is to back up data on the Internet, using a “cloud” storage solution. By frequently backing up to a Web-based location, the risk of data loss because of fire, theft, file corruption, or other disaster is virtually eliminated. Viewed through a Web browser, remote files and folders appear as if they are saved on an external local hard drive.

5.3.10. Setting up security policies

ICT policies are usually implemented in a workplace to ensure safe and appropriate use of Internet services and connections. A company may issue a document to be signed by employees to comply with their regulations. When using ICT in an educational environment it is good practice to set some rules for how it can be used by staff and students in order to protect everyone from potential risks. A good starting point is to create a document called an acceptable use policy (AUP).

An acceptable use policy (AUP) is a document that outlines what the school determines is acceptable in terms of behavior when using computers and the Internet . It typically applies to

students and staff. You should check if there is an existing AUP for your school system that you can use. If not, there are many examples of AUPs available online that you can use as a starting point for creating your own AUP.

In general, AUPs typically cover the appropriate use of e-mail, Internet, social media, network equipment, data, computers and mobile devices. They can also cover a wide range of ICT issues, including:

- Internet safety
- Data security
- Data protection
- Accessibility
- Health and safety

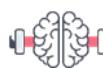
Your AUP should be created with input from parents, students, teachers, and school management. It should be communicated to all students, staff and parents, and you should continually review and update your AUP to take account of continuous advances in technology.

5.3.11. Securing equipment

storing all equipment in a secure location. Some tips for keeping devices secure include:

- Using laptop or tablet trolleys.
- Locking them securely in safe storage facilities.
- Using password controls.
- Setting up remote tracking, locking, and wiping to be used in the event of loss or theft.

Mobile devices such as tablets and laptops are particularly easy to steal in a school environment because they are easy to carry and hide, and because devices are used by a number of different people it can be difficult to keep track of them.



Self-check 5-3:

Dear Students! So far, we have explored the different Threats to Computers, Devices And Data.

Now it is your turn to answer the following self-check questions

1. Why is it important to use strong passwords and change them regularly?
 2. What are the benefits of using two-factor authentication (2FA)?
 3. Explain the role of regular data backups in data protection.
 4. What are some key practices for protecting personal information on social media?
-

Dear Students! Have you answered the self-check questions? If yes, let us summarize the unit.

Unit Summary

In Unit Five, you delved into the critical aspects of maintaining a safe online environment. The unit began with an overview of security and privacy, highlighting the significance of protecting personal and organizational data from unauthorized access, breaches, and cyber threats. You learned about the importance of security measures such as encryption, antivirus software, and regular software updates, which collectively help safeguard your devices and information.

The unit also covered potential threats to computers, devices, and data, including malware, unauthorized access, accidental damage, cybercrime, and social engineering. Understanding these threats is crucial for implementing effective protection strategies. Additionally, you explored the concept of digital identity and the importance of managing and protecting your online presence.

Finally, the unit provided practical guidance on protecting computers, devices, and data. This included using strong passwords, managing browser settings, performing regular backups, and setting up two-factor authentication. These practices are essential for creating a secure digital environment and mitigating risks associated with cyber threats.

Unit Review Questions

Part I: Multiple Choice Questions

1. Which of the following best describes encryption?
 - A. A method to delete data permanently
 - B. A process that encodes data so that only authorized users can read it
 - C. A software update process
 - D. A type of malware
2. What is malware?
 - A. A protective software for computers
 - B. Malicious software designed to damage or disrupt systems
 - C. A type of firewall
 - D. An encryption technique
3. Which of the following is a type of social engineering attack?
 - A. Phishing
 - B. Encryption
 - C. Firewall
 - D. Antivirus
4. Why is it important to regularly update software?
 - A. To increase the size of the software
 - B. To fix security vulnerabilities and improve functionality
 - C. To delete unnecessary files
 - D. To install new fonts
5. What does two-factor authentication (2FA) involve?
 - A. Using two different passwords for the same account
 - B. Verifying identity through two different methods
 - C. Using encryption and decryption simultaneously
 - D. Installing two antivirus programs

Part II: Short Answer Questions

Instruction: Give brief answer for the following review questions:

1. Explain the importance of using strong passwords and give an example of a strong password.
2. What is phishing and how can it be identified?
3. Describe the role of antivirus software in protecting your computer.
4. Why is managing browser settings important for online security?
5. What steps can you take to secure your digital identity online?
6. Explain the importance of using strong passwords and give an example of a strong password.
7. What is phishing and how can it be identified?
8. Describe the role of antivirus software in protecting your computer.
9. What steps can you take to secure your digital identity online?
10. How does two-factor authentication (2FA) enhance security?
11. What is social engineering, and how can you protect yourself from it?
12. Describe the importance of regular data backups.
13. What are cookies, and how can managing them improve your online privacy?

Answer key for Self-check questions

Self-Check 5-1: Overview of Security and Privacy

1. Answer: Regular updates fix security vulnerabilities, improve software performance, and prevent potential exploits by cyber threats.
2. Answer: Privacy involves controlling personal information and data, determining who has access to it, and how it is used.
3. Answer: Key components include personal information (e.g., name, contact details) and usernames and account credentials.

Self-Check 5-2: Threats to Computers, Devices, and Data

1. Answer: Malware is malicious software designed to damage files or steal information. It can disrupt computer operations, collect sensitive information, or gain unauthorized access to systems.
2. Answer: A virus infects computer files and requires user action to spread, while a worm is self-replicating and spreads automatically through networks.
3. Answer: Phishing involves fraudulent emails that appear to come from legitimate sources, requesting sensitive information. Recognize it by checking for suspicious links, spelling errors, and unexpected requests for personal data.
4. Answer: Identity theft is when someone uses another person's identity for fraudulent purposes. Consequences include financial loss, damaged credit, and significant time and effort to restore one's identity.

Self-Check 5-3: Protecting Computers, Devices, and Data

1. Answer: Strong passwords prevent unauthorized access, and changing them regularly reduces the risk of someone gaining prolonged access if a password is compromised.
2. Answer: 2FA adds an extra layer of security, requiring two forms of verification, making it harder for unauthorized users to access accounts even if they have the password.
3. Answer: Regular backups ensure that data can be recovered in case of hardware failure, accidental deletion, or cyberattacks, minimizing data loss.
4. Answer: Key practices include using privacy settings to control who can see your information, being cautious about what you share, and regularly reviewing your account settings.

Project 1: Collaborative Academic Project Using Google Drive

In the following activity you will learn how to collaborate on a group project using Google doc.



Collaborative Academic Project Using Google Drive

Title: Collaborative Academic Project Using Google Drive

Objective: To demonstrate effective collaboration on an academic project using Google Drive and google Doc.

Case Scenario: You are a group of four students working on a group project for your history class. Your project is about creating a presentation on "The Industrial Revolution." Each team member has specific tasks to complete, and you need to collaborate using Google Drive to create a cohesive and well-organized presentation.

Step-by-Step Activity:

Step 1: Set Up the Project Folder

Create a new Google Drive folder for your project and name it "Industrial Revolution History Project."

Step 2: Share the Folder

Share the folder with your team members:

- Click on the folder.
- Click the "Share" button in the upper-right corner.
- Enter your team members' email addresses and set their permissions to "Can edit."

Step 3: Create Subfolders

Inside the project folder, create subfolders for each major section of your project, such as "Research," "Images," "References," and "Presentation."

Step 4: Organize Tasks

In the "Research" subfolder, create a **Google Docs document** for each team member to write their respective sections of the project. Assign each team member a specific section, like "Causes," "Effects," "Innovations," and "Impact."

Step 5: Collaborative Document Editing

Each team member should:

- Open their assigned Google Docs document.

- Add their research and content to their document.
- Comment and suggest edits in each other's documents to ensure consistency and accuracy.

Step 6: Collecting Visuals

In the "Images" subfolder, each team member should upload relevant images, graphs, and charts that complement their research. Ensure images are properly named for easy identification.

Step 7: Creating References

In the "References" subfolder, create a Google Sheets document for collecting and citing your sources. Each team member should add their sources and properly format citations.

Step 8: Presentation

In the "Presentation" subfolder, create a Google Slides presentation. Each team member can work on a specific section of the presentation and incorporate the content from their Google Docs documents.

Step 9: Regular Updates and Communication

Use the comment feature in Google Docs, Sheets, and Slides to communicate with team members, ask questions, and provide feedback.

References

1. Berners-Lee, T. (2000). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. HarperSanFrancisco.
2. Castells, M. (2010). *The Rise of the Network Society: The Information Age: Economy, Society, and Culture* (Vol. 1). Wiley-Blackwell.
3. Kahn, R. E., & Cerf, V. G. (1999). What is the Internet (And What Makes It Work). *Internet Policy Institute*.
4. Zakon, R. H. (2020). Hobbes' Internet Timeline 25. *Internet Society*. Retrieved from <https://www.zakon.org/robert/internet/timeline/>
5. Chao, L. (2021). Mastering Web Browser Development: Chrome and Firefox Developer Tools. Apress.
6. Google. (2022). Google Chrome Help Center. Retrieved from <https://support.google.com/chrome/>
7. Mozilla Foundation. (2021). The Mozilla Firefox Web Browser. Retrieved from <https://www.mozilla.org/en-US/firefox/>
8. Pogue, D. (2021). Windows 10: The Missing Manual (3rd ed.). O'Reilly Media. Badke, W. B. (2021). Research Strategies: Finding Your Way Through the Information Fog (6th ed.). iUniverse.
9. Cohen, D. J., & Rosenzweig, R. (2005). Digital History: A Guide to Gathering, Preserving, and Presenting the Past on the Web. University of Pennsylvania Press.
10. Morville, P., & Rosenfeld, L. (2015). Information Architecture: For the Web and Beyond (4th ed.). O'Reilly Media.
11. Rumsey, S. (2008). How to Find Information: A Guide for Researchers. Open University Press.
12. Sullivan, D. (2019). Search Engine Land's Guide To SEO. Third Door Media. Retrieved from <https://searchengineland.com/guide/what-is-seo>
13. Cain, J. (2018). Email Essentials: How to Write Effective Emails and Build Relationships That Last. Wiley.
14. Cole, G. M., & Smith, C. (2019). Using Microsoft Teams for Collaboration and Communication. Microsoft Press.

15. Ferris, M. (2020). *The Beginner's Guide to Slack: Communication and Collaboration in Teams*. Wiley.
16. Granneman, S. (2021). *Collaborative Workspaces and Cloud Computing: Google Workspace and Office 365*. O'Reilly Media.
17. Kroski, E. (2009). *The Social Media Guide for Social Good: Communication and Collaboration*. ALA TechSource.
18. Andress, J. (2019). *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (2nd ed.). Syngress.
19. Easttom, C. (2018). *Computer Security Fundamentals* (4th ed.). Pearson IT Certification.
20. Granneman, S. (2019). *Privacy and Security on the Internet: A Practical Guide for Secure Computing*. Apress.
21. Stanger, J. (2021). *CompTIA Security+ Study Guide: Exam SY0-601* (8th ed.). Sybex.
22. Whitman, M. E., & Mattord, H. J. (2017). *Principles of Information Security* (6th ed.). Cengage Learning.

Module III

Professional Ethics

Contents

| No | Contents | page |
|----------|--|------|
| | Module Description | 261 |
| 1 | Unit 1: Developing Professions and Professionalism | 262 |
| 1.1 | Concept of Profession and Professionalism | 262 |
| 1.2 | Profession and Professionalism | 263 |
| 1.3 | Characteristics of professionalism | 263 |
| 1.4 | Values of professionalism | 267 |
| | Unit Summary | 267 |
| | Unit Review Questions | 268 |
| 2 | Unit 2: Describing the Concept of Ethics and Professional Ethics | 269 |
| 2.1 | Introduction | 269 |
| 2.2 | Fundamentals of Professional Ethics | 269 |
| 2.3 | Common Principles of Professional Ethics | 270 |
| 2.4 | The importance of professional ethics | 272 |
| 2.5 | Professional Ethics Required from a Worker | 273 |
| 2.6 | Ethiopian ethical service delivery | 273 |
| 2.7 | Some Characteristics of Work Ethics | 275 |
| | Unit Summary | 276 |
| | Unit Review Questions | 276 |
| 3 | Unit 3: Reflecting and Evaluating Ethical Practices in the Workplace | 277 |
| 3.1 | Introduction | 277 |
| 3.2 | Ethical practices in the workplace | 277 |
| 3.3 | Unethical Behavior in the Workplace | 279 |
| 3.4 | Factors that Affect the Practice of Ethical Behavior | 281 |
| 3.5 | Advantages and Implications of respecting Workplace Ethics | 282 |
| | Unit Summary | 283 |
| | Unit Review Questions | 283 |
| 4 | Unit 4: Applying Codes of Ethics, Conduct, and Standards of Professional Practice | 284 |
| 4.1 | Introduction | 284 |
| 4.2 | The Purpose of Professional Codes of Conduct and Practice | 284 |

| | | |
|----------|--|------------|
| 4.3 | Types of Codes of Ethics | 285 |
| 4.4 | Code of Ethics among Professionals | 286 |
| 4.5 | The Difference between a Code of Ethics and a Code of Conduct | 287 |
| 4.6 | Applying Code of Conduct | 287 |
| 4.7 | Standards of Professional Practice | 287 |
| | Unit Summary | 288 |
| | Unit Review Questions | 289 |
| 5 | Unit 5: Identifying Mechanisms of Professional Ethics Decision Making | 290 |
| 5.1 | Introduction | 290 |
| 5.2 | Ethical Decision Making | 291 |
| 5.3 | Rationalization | 293 |
| 5.4 | Some Steps to Ethical Thinking and Ethical Behaving | 293 |
| 5.5 | Important principles for making Ethical decision | 294 |
| 5.6 | Dress codes and code of conduct | 296 |
| 5.7 | Workplace policies and procedures | 296 |
| 5.8 | Industry Compliance and Regulations | 297 |
| | Unit Summary | 298 |
| | Unit Review Questions | 298 |
| 6 | Unit 6: Ethical Standards, Practice, and Employability Skills | 299 |
| 6.1 | Introduction | 299 |
| 6.2 | Promoting Ethical Standards and Practice | 300 |
| 6.3 | Ways to Improve an Organization's Ethical Climate | 305 |
| | Unit Summary | 306 |
| | Unit Review Questions | 208 |
| | References | 312 |

Module Description

This module describes the performance outcomes, skills and knowledge required to:

- Provide rules on how a person should act towards other people and institutions in the environment.
- Respect for persons, which means showing concern for the well-being of others & self.
- Correctly interpret and apply code of ethics, conduct and standards of professional practice to industry and organizational decision making.
- Receive, respond and act on oral and written organizational communication (express and support one's views to others)
- Promote rational thinking and unbiased judgment among students.
 - ✓ Improvement of the cognitive skills (skills of the intellect in thinking clearly)
 - ✓ Learn the value of teamwork in the world of work

Module Instruction:

Learning Instructions: How to use this Module

For effective use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what are being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

UNIT 1

Professions and Professionalism

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding Professional Ethics

Unit Learning Outcomes

This unit will also assist you in attaining the following learning outcome. Specifically, upon completion of this unit, you will be able to:

- Explain the profession and professionalism.
- Describe the characteristics of professionalism.
- Identify the values of professionalism.

Key Terms: Occupation, Profession, professionalism, skill, values, Honest, Ethics

1.1 Concept of Profession and Professionalism

This unit focuses on two main conceptual issues. The first is the basic concepts of professions, followed by professionalism, which will help you get a better understanding of workplace ethics. In general, we might say being a professional means taking care, being courteous, and following conventional business norms. Professionalism serves a gatekeeping function. Individuals judge messages, and in turn the sender of those messages, by their professionalism

Professionalism is the conduct, behavior and attitude of someone in a work or business environment. A person doesn't have to work in a specific profession to demonstrate the important qualities and characteristics of a professional.

A person can be called as a professional based on the standards of education, training, specific knowledge and skill she/he possesses in order to fulfill the requirements of a particular task

assigned to him. Professional workplace behavior is necessary for the long-term success of a business, whether it's a big corporation or small business.

- What is a profession?
- What is professionalism?
- Reflect on the value of professional ethics in our daily lives.

1.2 Profession and Professionalism

The word profession can be defined as a vocation or occupation requiring, especially, usually advanced education, knowledge and skill. Thus, the term profession refers to the knowledge, skill and ability which can be obtained through formal education and training. A person who attained a certain specialized skill is known as a professional. It should also be acceptable by society. On the other hand, professionalism is defined as “the skill, good judgment, and polite behavior that is expected from a person who is trained to do a job well”

Although each profession requires people with specific skills and talents, every efficient employee should have general professional qualities. These qualities can help you show employers that you are a well-rounded individual. Those who are highly respected and admired in their roles exude professionalism. Some qualities of a professional include

- Knowing your staff.
- Standing for something/job/.
- Keeping your word/confidentiality/.
- Being honest.

1.3 Characteristics of professionalism

The art of professionalism can be understood as the practice of doing the right thing, not because of how one feels but regardless of how one feels. Professionals make a profession of the specific kind of activity and conduct to which they commit themselves and to which they can be expected to conform. Professional workers take responsibility for their own behavior and work effectively with others.

Professional characteristics refer to the qualities a person exemplifies in a business environment. Professionalism includes standards for behavior and the employee's ability to embody the company's values and do what their employer expects of them. Professionalism is necessary for the long-term success of any business, large or small. It ensures that customer relationships are maintained, employee interactions are positive and that a company meets its goals and objectives.

1.3.1 Characteristics of professionals in the workplace

There are different characteristics of professionalism:

Professional appearance

Professionals should always strive for a professional appearance, including appropriate attire and proper hygiene and grooming. Clothing should always be clean and ironed properly. Pants, dresses, formal skirts, crisp white shirts and leather shoes are all appropriate for a professional's wardrobe.

Reliable

Professionals are dependable and keep their commitments. They do what they say they will do and don't overpromise. Professionals respond to colleagues and customers promptly and follow through on their commitments in a timely manner. Punctuality is a key aspect of this professional characteristic. It's always important to clarify any areas of uncertainty when dealing with customers or members of your team to ensure there are no mistaken assumptions or surprises.

Ethical behavior

Embodying professionalism also means to be committed to doing the right thing. Honesty, open disclosure and sincerity are all characteristics of ethical behavior. Many organizations include a commitment to ethical behavior in their code of conduct. Professionals can adopt a personal code of conduct and make the same commitment on an individual basis.

Organized

A professional keeps their workspace neat and organized so that they can easily find items when they need them. All files and paperwork should be in place and, if they have to deliver a presentation, all materials should be ready well in advance so there are no unexpected delays.

Accountable

Just as a professional accepts credit for having completed a task or achieved a goal, they also are accountable for their actions when they fail. They take responsibility for any mistakes that they make and take whatever steps necessary to resolve any consequences from mistakes. They are accountable and expect accountability from others.

Professional language

People who behave with professionalism monitor every area of their behavior, including how they talk. They minimize the use of slang and avoid using inappropriate language in the workplace. They even are conscientious of the language they use in informal settings.

Separates personal and professional

Professionals understand the importance of separating their personal lives from their professional lives. While professionals may experience the same challenges in their personal lives as others, they maintain a clear separation between their professional lives and workplace demeanor.

Positive attitude

Part of being a professional means maintaining a positive, can-do attitude while working. A positive attitude will improve a professional's overall performance and increase the likelihood of a positive outcome. It will also impact the behavior and performance of others, improving employee morale in the office.

Emotional control

Emotional control is another key characteristic of professionalism. Professionals understand the importance of maintaining their composure and staying calm in all situations. By remaining calm, even during challenging moments, others can rely on them to be rational and of sound judgment.

Effective time management

An employee who knows how to manage their time well is viewed by their peers as a professional. Some characteristics of time management abilities include showing up at the office on time in the morning, being on time for meetings and letting someone in the office know if they suspect that they might be late.

Focused

A professional is clear about their goals and understands what they need to accomplish to achieve them. They know how to stay focused on their work to maintain their productivity. Professionals recognize the importance of maintaining focus to improve the quality of their work and be as efficient as possible.

Poised

Professionals should demonstrate poise, a calm and confident state of being. Being poised means maintaining a straight posture, making eye contact when communicating and helping establish a friendly and professional presence. Being poised means also staying calm during times of heightened pressure.

Respectful of others

Professionals always treat others with respect. They understand that though humor is appropriate in the workplace, they should always use it with respect to others. The only time that professionals engage in conversations about other people is if they are evaluating their performance and looking for constructive ways to improve their performance in the workplace.

Strong communicator

A professional must have strong communication skills. This means that they not only can effectively and efficiently convey messages to others but also that they can actively listen to and understand what others are telling them. By engaging in open and constructive communication with others, professionals can collaborate more effectively and accomplish a lot.

Possesses soft skills

Soft skills are personal attributes that allow someone to interact effectively with others. Soft skills include things like leadership, critical thinking, teamwork and people skills. Soft skills help professionals to behave courteously when addressing colleagues and managers, use the right language when communicating and respect the opinions of others.

1.4 Values of professionalism

Professional workplace behavior is necessary for the long-term success of a business, whether it's a big corporation or small business. Someone who displays professional values will:

- Portray a professional image through reliability, consistency and honesty.
- Dress and act appropriately.
- Deliver work outcomes to agreed quality standards and timescales.
- Be accountable for their actions

Unit Summary

The concepts of ethics and professional ethics place special emphasis on the part that professionals play in shaping the ethical climate within their companies. Professional ethics are the set of guidelines that professionals follow when interacting with clients, colleagues, and the general public. They enable specialists to distinguish between **good and bad, right and wrong** by using morality as a standard of evaluation.

All professions, regardless of their particular area of expertise, must adhere to fundamental standards of professional ethics. These values include the significance of having a strong sense

of identity, being on time, being honest and ethical, using resources properly, being loyal, and having self-efficacy.

Ethics is the critical study and assessment of what constitutes good, wicked, right, and wrong behavior in human beings.

Unit Review Questions

Direction. Write a short answer.

1. What is a profession and what is professionalism?
2. List out some characteristics of professionalism.
3. Mention some values of professionalism.

UNIT 2

Describing the Concept of Ethics and Professional Ethics

2.1. Introduction

This unit focuses on two major conceptual issues. The first is the basic concept of ethics, and professional ethics is another concept, that helps you understand the set of values and standards of work discipline in charge of changing the culture of ethics in their respective organizations as role models in accepting and implementing their professional code of ethics.

This will create an attractive and conducive working environment for both consumers and the public as a whole, which will enable them to be served effectively and efficiently without discrimination or in accordance with ethical standards.

Learning outcomes: After training this unit, you will be able to:

- Describe the concepts of ethics and professional ethics.
- Identify some characteristics of work ethics.
- Explain the Principles of Common Professional Ethics

Key Terms and Concepts

- Professional ethics, spirit, corruption, benchmark, moral, code of ethics, identity.

2.2. Fundamentals of Professional Ethics

What is the significance of professional ethics?

Professional workers are in charge of changing the culture of ethics in their respective organization. They are expected to be role models in accepting and implementing their professional code of ethics. As a result, a fertile and conducive work environment will be created and effective, efficient, just, and ethical services be delivered to the customers and the public at large.

Ethics is: the critical examination and evaluation of what is good, evil, right and wrong in human conduct. A specific set of principles, values and guidelines for a particular group or organization.

Ethics examines the rational justification for our moral judgments; it studies what is morally right or wrong, just or unjust. In a broader sense, ethics reflects on human beings and their interaction with nature and with other humans, on freedom, on responsibility and on justice

Professional Ethics: Professional ethics can generally be defined as an established set of principles that govern how a professional relates with clients, fellow professionals, and the public at large. Most professional specialty fields today have their own rules of professional conduct, which is monitored by the memberships' leaders.

Professional ethics refers to ethics that enables professionals to distinguish what is right from what is wrong using morality as standard of evaluation. Professional ethics can be conceived as a parameter by which actions and behaviors of a professional can be determined as right or wrong. In absolute terms all professions do not have the same set of values and standards such as accountants, managers, lawyers, trainers, medical doctors, engineers and technicians etc.

2.3. Common Principles of Professional Ethics

There are fundamental principles of professional ethics that apply to all professions, regardless of their specific specialization. They are:

- A. **Punctuality:** Punctuality refers to the state of being strictly observant of an appointed or regular time across all professions. A worker should be punctual not only when he/she meets deadlines and when he/she is always available during working hours.
- B. **Honesty and integrity:** honesty basically refers to the art of telling the truth. Employers and institutions expect their workers to be honest and hence professionals should be honest, and at the same time work in good behavior and integrity. Integrity is the quality of being honest and showing a consistent and uncompromising adherence to strong moral and ethical principles and values

- C. **Proper utilization of resources:** A worker in any profession should handle the resources in a way that she/he can use the resources for a longer time. In other words, avoiding wastage in any, be it material or financial resources.
- D. **Loyalty and self-efficacy:** Any worker, in a given profession must stand for, and not against the employer or the institution. Moreover, an ethical professional must develop self-efficacy that is an appraisal or evaluation of a professional about his /her professional and personal competence to succeed in a particular task.
- E. **Working in cooperation with colleagues:** A worker of a given profession should make sure that he/she has a good relationship with other workers. The main rationale behind these codes of professional ethics is to create a fertile and good atmosphere for cooperation and better productivity. He/she should serve hand in hand with other professionals for better productivity and effective provision of service.
- F. **Anti-corruption spirit:** One way of fighting corruption is by making professionals comply with the laws, rules, and regulations of the state.
- G. **Confidentiality:** A worker of a given profession needs to keep some information that should be kept secret. For instance, in case of a physician or nurse he/she has to keep all the information regarding the patient contained in a patient's chart. Another example is that a trainer should keep all information about a student's achievements confidential.
- H. **Commitment:** A professional should help his /her clients and the whole community to maintain and satisfy societal or public demands.
- I. **Persistent tolerance and democratic culture:** Any worker or professional have to develop the ability and willingness to accommodate differences in ideas, outlooks or views is indispensable
- J. **Respecting the dignity of people:** A professional or worker should respect the dignity of all personnel serving them in accordance with their basic needs, irrespective of their sex, personal status, and religion or by supernal factors involved.
- K. **Impartiality/Non-partiality:** A professional should be impartial he/she should treat all equally in his/her service delivery without any discrimination. For example, a trainer should treat his/her students equally on the basis of their academic performance or achievement.

- L. **Accountability and responsibility:** Any worker in a given profession has professional responsibilities or duties. As part of a given organization, everyone is expected to be answerable for his/her actions.
- M. **Transparency:** That is, his/her work must be open to the public to whom he/she delivers public services. Confident professionals who work for the people effectively and efficiently are usually transparent.
- N. **Responsiveness:** Responsiveness refers to the extent that a professional satisfies the needs, preferences, or values of his/her connections and professional relationship with his clients or public, he/she has to be able to reply or respond to the people's demand.

2.4. The importance of professional ethics

The importance of professional ethics cannot be overstated. It is essential for building trust and credibility with clients, colleagues, and the wider community. It also helps to maintain the integrity and reputation of the profession and ensures that professionals are held accountable for their actions. The prime objectives of professional ethics are as follows:

1. Moral awareness (proficiency in recognizing moral problems in engineering like plagiarism and patenting)
2. Convincing moral reasoning (comprehending and assessing different views)
3. Moral coherence (forming consistent viewpoints based on facts)

A professional code of ethics is designed to ensure employees are behaving in a manner that is socially acceptable and respectful of one another. It establishes the rules for behavior and sends a message to every employee that universal compliance is expected. Similarly, the other purposes of professional ethics

- Advance the quality of service. the quality of service that professionals could render
- Evaluate the performance of professionals in each profession.
- Distinguish between acceptable and non-acceptable characters or codes of conduct.
- Serve as a foundation for professional identity.

Being professional can ensure a positive first impression, successful interpersonal relationships, and a lasting reputation within your organization and industry.

2.5. Professional Ethics Required from a Worker

Work is an activity that a person engages in regularly to earn a livelihood. Ethical requirements address the fact that all personnel on engagements should maintain independence in mind and in appearance, and perform all professional responsibilities with integrity.

Ethical standards serve as a benchmark for professionals to evaluate their activities and code of conduct accordingly. In the context of web development, these ethical standards are crucial to maintaining the integrity of the profession and ensuring trust with clients and users.

Example in ICT and Web Development:

A web developer should not intentionally write inefficient code or include hidden vulnerabilities that can cause a website to malfunction later. It would be unethical to advertise a web application as having certain security features if it does not actually possess them. Additionally, inflating the capabilities of a product to secure a contract would be considered unethical.

Professional organizations in the ICT and web development sector may define their ethical approach in terms of several key components, including: honesty, trustworthiness, transparency, accountability, confidentiality, objectivity, respect, obedience to the law, and loyalty.

2.6. Ethiopian ethical service delivery

In the Ethiopian context, twelve principles of ethical service delivery are identified as the basis for ethical service delivery in the working areas and the public at large. Based on these principles, each profession is expected to develop its own professional codes that address the particular nature of the profession. The values are stated as principles of ethical service delivery and are 12 in number. These are:

- i. Integrity: Organizations and personnel demonstrate integrity through a consistency between actions and words that inspires trust and credibility. Integrity also means

- keeping promises, honoring commitments, meeting deadlines and refusing to participate in unscrupulous activities or business dealings.
- ii. Loyalty: is proven by never disclosing information learned in confidence and by remaining faithful to coworkers, clients, business partners and suppliers. Loyal employees avoid conflicts of interest, help build and protect the good reputation of their company and help boost the morale of their coworkers.
 - iii. Transparency: embodies honesty and open communication because to be transparent someone must be willing to share information when it is uncomfortable to do so. Transparency is an individual being honest with him about the actions he is taking.
 - iv. Confidentiality: involves a set of rules or a promise usually executed through confidentiality agreements that limits the access to or places restrictions on distribution of certain types of information.
 - v. Honesty: All personnel must be committed to telling the truth in all forms of communication and in all actions. This includes never purposely telling partial truths, selectively omitting information, making misrepresentations or overstatements. Honesty also means reliably sharing both good and bad news with equal candor
 - vi. Accountability: Accountability requires a total commitment to the ethical quality of all decisions, actions and relationships. High expectations for ethical behavior drive business practices when an organization and its personnel are held accountable to fellow employees, consumers, the local community and the wider public in general
 - vii. Serving the public interest: the term 'public interest' means matters concerning welfare of the people. Public Interest is anything that affects the rights, health, or finance of the public at large. It is a common concern among citizens in the management and affairs of local, state, and national government.
 - viii. Exercising legitimate authority: A legitimate authority is one which is entitled to have its decisions and rules accepted and followed by others. In the case of law, people feel a personal responsibility to comply voluntarily with those laws that are created and enforced by legitimate legal authorities.
 - ix. Impartiality: Impartiality (also called evenhandedness or fair-mindedness) is a principle of justice holding that decisions should be based on objective criteria, rather than on the

- basis of bias, prejudice, or preferring the benefit to one person over another for improper reasons.
- x. Respecting the law: Respect is demonstrated by a full commitment to the human rights, dignity, autonomy, interests and privacy of all personnel. It means recognizing that everyone deserves equal respect and support for sharing ideas and opinions, without fear of any penalty or form of discrimination
 - xi. Responsiveness: Responsiveness means "being able to react quickly," like a sports car whose responsiveness makes it fun to drive, or a "responding with emotion," like the responsiveness of an audience at the concert of their all-time favorite singer.
 - xii. Leadership: Demonstrated by a conscious effort to set a positive example of ethical behavior, leadership is a commitment to excellence through ethical decision-making. Businesses and business executives maintain their leads by constantly improving operational efficiency, worker satisfaction and customer approval.

2.7. Some Characteristics of Work Ethics

Work is an activity that a person engages in regularly to earn a livelihood.

A professional job is a career that requires a specific amount of advanced training and education. These jobs often require some level of post-secondary education. Example, doctor.

A work ethic is a personal set of values that determines how any employee approaches their work. Employees with strong work ethics are highly motivated and produce consistently high-quality results. A good work ethic can be taught as long as more productive behaviors are clearly demonstrated to your employees. Among the many characteristics of work ethics, the following are some to be mentioned:

1. Appearance: Displays proper dress, grooming, hygiene, and manners.
2. Attendance: Attends class, arrives and leaves on time, tells instructor in advance of planned absences, and makes up assignments promptly.
3. Attitude: Shows a positive attitude, appears confident and has true hopes of self.
4. Character: Displays loyalty, honesty, dependability, reliability, initiative, and self-control.

5. Communication: Displays proper verbal and non-verbal skills and listens.
6. Cooperation: Displays leadership skills; properly handles criticism, conflicts, and stress; maintains proper relationships with peers and follows chain of command.
7. Organizational Skill: Shows skills in management, prioritizing, and dealing with change.
8. Productivity: Follows safety practices, conserves resources, and follows instructions.
9. Respect: Deals properly with diversity, shows understanding and tolerance.
10. Teamwork: Respects rights of others, is a team worker, is helpful, is confident, displays a customer service attitude, and seeks continuous learning.

Unit Summary

The concepts of ethics and professional ethics place special emphasis on the part that professionals play in shaping the ethical climate within their companies. Professional ethics are the set of guidelines that professionals follow when interacting with clients, colleagues, and the general public. They enable specialists to distinguish between good and bad, right and wrong by using morality as a standard of evaluation.

All professions, regardless of their particular area of expertise, must adhere to fundamental standards of professional ethics. These values include the significance of having a strong sense of identity, being on time, being honest and ethical, using resources properly, being loyal, and having self-efficacy.

Ethics is the critical study and assessment of what constitutes good, wicked, right, and wrong behavior in human beings.

Unit Review Questions

Direction: Write a short answer to the following questions:

1. What is the importance of professional ethics?
2. List out some characteristics of work ethics.
3. Identify the common principles of professional ethics.

UNIT 3

Reflecting and Evaluating Ethical Practices in the Workplace

3.1. Introduction

This unit discusses the fundamentals of ethical behavior and how organizations can establish ethical standards for their employees based on them. Ethics in the workplace sets the standard for right and wrong behaviors and helps build morally upright employees. It guides employees whenever they face an ethical dilemma. Ethics also plays a significant role in creating a positive ambiance, ensuring happiness, and improving cooperation and trust among team members. To implement workplace ethics, you need to start by understanding the basic concepts of ethical behavior in the workplace.

Learning outcomes: After training this unit, you will be able to:-

- Explain and evaluate the practices of ethics in the workplace.
- Identify ethical and unethical behaviors in the workplace.
- Explain factors that affect the practice of ethical behavior.

Key Terms and Concepts

- Nepotism, Undue Pressure, Embezzlement, Abuse, Trust, culture Coaching, Apologize, Accountability

3.2. Ethical practices in the workplace

- What are ethical and unethical behaviors in the workplace?
- Identify some of them.

Ethical practices of workers or professionals can be evaluated using ethical standards of workers. Ethical behaviors ensure maximum productivity output at work and could be pivotal for career growth. The following are the examples of ethical behavior:

- | | |
|--|--|
| A. Obey the company's rules and regulations. | F. Accountability |
| B. Communicate Effectively | G. Uphold Trust |
| C. Develop professional relationships. | H. Show initiative without being told. |
| D. Taking Responsibility | I. Respect your colleagues. |
| E. Professionalism/Standards | J. Work Smarter |

Trust and Mutual Respect for Colleagues at Work

These examples of ethical behaviors ensure maximum productivity output at work. Encouraging mutual respect will help to: Reduce workplace stress, conflict and problems. An increase in workplace respect will help to improve communication between colleagues, increase teamwork and reduce stress as peace in the workplace soars. Increase productivity, knowledge and understanding. Every worker should try their best to get respect and trust from colleagues in the workplace. The ways to get respect from your coworkers are:

- Follow the rules.
- Work Hard.
- Talk less, listen more.
- Assume the best about people.
- Apologize and admit mistakes.
- Take criticism and learn from it.
- Stand up for yourself.
- Help other people succeed.

To build trust in the workplace also considers the following:

- Listen more than you speak.
- Solicit and act on feedback.
- Show appreciation every day.
- Empower your team by trusting them first.
- Encourage coaching.

- Practice consistency.
- Focus on nonverbal communication soft skills and
- Create an inclusive culture

3.3. Unethical Behavior in the Workplace

Workplace ethics are a dynamic set of values that vary with people and their definition of a workplace. For some, it is a physical office they go to every day, while others, their home office. No matter whether you work from home or commute to work every day, workplace ethics is required to build a successful career. Organizations are known to embrace ethical practices and behaviors to increase productivity and uphold integrity—while setting a penalty for workers who default workplace ethics. Workplace ethics are the set of values, moral principles, and standards that need to be followed by both employers and employees in the workplace. It is the set of rules and regulations that need to be followed by all staff of the workplace.

These ethics are implemented by employers to foster both employee-employee relationship and employee-customer relationships. An organization may decide to put these ethics into writing or not—they are however meant to be followed. Unethical workplace behaviors:

- A. Lies/Dishonesties
- B. Taking credit for others hard work
- C. Verbal Harassment/Abuse
- D. Violence/Forcefulness
- E. Non-office-related work
- F. Extended Breaks
- G. Theft/Embezzlement
- H. Sexual Harassment
- I. Corrupt Practices
- J. Management/Employers Unethical Behaviors
- K. Sex for Job/Promotion
- L. Verbal Harassment
- M. Undue Pressure
- N. Nepotism

- O. Unfriendly work environment
- P. Unrealistic Expectations

How to Solve Unethical Issues at the Workplace

i. Have Rules

Organizations need to have predefined rules and regulations regarding workplace ethics. These rules and regulations should be given to new employees together with their employment contracts. Also having the rules written at strategic places at the workplace will also help remind people about the rules. People tend to unconsciously imbibe things they see every day.

ii. Accept Feedback/Complaint

Make it easy for employees to send feedback or complaints in case of harassment, abuse, or any other unethical activities going on in the workplace.

iii. List Consequences for Unethical Behaviors

Consequences for unethical behaviors should also be placed alongside the rules at strategic places in the organization. That way, if anyone wants to ignore the rules despite seeing them, the fear of getting punished will stop him or her from going ahead.

iv. Swift Justice/Disciplinary Action

Some companies often cover up issues of rape, sexual harassment, etc. when the perpetrator is a high-ranking member of the organization. Things like this should not be accommodated. Irrespective of who breaks the rule, there should be swift disciplinary action by the organization.

v. Conduct on-job ethical training.

On-job training and development is vital to workplace ethical practices.

3.4. Factors that Affect the Practice of Ethical Behavior

There are different factors that can affect the Practice of ethical behavior are individual and social Factors.

1. Individual Factors

Many individual factors affect a person's ethical behavior at work, such as knowledge, values, personal goals, morals and personality. The more information that you have about a subject, the better chance you will make an informed, ethical decision. For example, what if you had to decide whether to approve building a new company store? What if you did not have the knowledge that the store would disturb an endangered species nest? Without the appropriate knowledge, you could be choosing an unethical path.

Values are an individual's judgment or standard of behavior. They are another individual factor that affects ethical behavior. To some people, acting in an improper way is just a part of doing business. Would you feel that it is ethical to make up lies about your competitor just to win a contract? Some people's standard of behavior will feel that lying for a business financial win is not unethical.

Morals are another individual characteristic that can affect an individual's ethics. Morals are the rules people develop as a result of cultural norms and values and are, traditionally, what employees learn from their childhood, culture, education, religion, etc. They are usually described as good or bad behavior. Would you have good morals if you pushed a product on a customer that you knew was not going to help solve a problem?

Many ethical work situations will also be affected by a person's goals. Which characteristics do you feel are worthy to aspire to? Is financial gain ranked ahead of good character or integrity? If your personal goals are about acquiring wealth no matter what the consequence, then you might act unethical in the future.

Lastly, an employee's personality plays an important factor in determining ethical behavior. Do you enjoy risk or do you prefer the safe route? Individuals who prefer to take risks tend to have a higher chance of unethical conduct at work. For example, if you are willing to risk dumping

chemicals into a nearby water supply to launch a profitable drug, then your riskiness could end up creating health issues in local citizens for the sake of financial gain.

2. Social Factors

Cultural norms, the Internet and friends and family are three social factors that can affect ethical behavior. Different cultures have norms that vary from place to place in the business world. For example, you might have to face a request for a bribe in order to conduct business in certain countries. This might be unethical to you but considered an acceptable norm in their workplace.

3.5. Advantages and Implications of respecting Workplace Ethics

Ethics is a way of thinking that encourages people to consider the impact of their actions on others and to act in a way that is good for the greater good. It helps us to determine what is right and wrong and to make decisions that are based on moral values.

Advantages of Ethics in the Workplace:

It can stimulate positive employee behavior and create a positive ambiance in the workplace

- A. Ensures management guides and mentors their employees in a healthy environment
- B. A workplace with good ethics usually strengthens the bond employees have with their superior
- C. It boosts productivity through employee performance and job satisfaction which in turn increases company growth.
- D. Bad workplace ethics can cause a strain in the relationship with company stakeholders
- E. When it leaks (which it most likely will) poor behavior can be recorded and propelled into unsavory headlines online. This can lead to reputational damage to the brand name.

The ethical implication of an action implies or suggests what we believe about right and wrong. For example most of us believe it is wrong to buy or sell another human being. The success is

debatable, but regulations addressing surrogate pregnancies attempt to ensure that people are not commercially produced and sold.

Unit Summary

Ethical practices in the workplace are crucial for maximum productivity output and career growth. Trust and mutual respect are essential for reducing workplace stress, conflict, and problems, improving communication, teamwork, and fostering peace. Workplace ethics are a dynamic set of values that vary by individual and organizational definitions. Employers and employees must follow these ethics to foster employee-employer and employee-customer relationships.

Unethical behavior in the workplace can be addressed through on-job training and development. Factors that affect ethical behavior include individual factors such as knowledge, values, personal goals, morals, and personality. Knowledge helps individuals make informed decisions, while values are judgments or standards of behavior. Morals are rules developed from cultural norms and values, and personal goals can also impact ethical work situations. Personality plays an important role in determining whether an individual's goals prioritize good character or integrity. Overall, ethical practices in the workplace are essential for a successful career and a healthy work environment.

Unit Review Questions

Direction: Write a short answer to the following questions:

1. Explain the practice of ethics in the workplace.
2. Evaluate the practice of ethics in your workplace.
3. Identify ethical behaviors in the workplace.
4. Identify unethical behaviors in the workplace.
5. Explain factors that affect the practice of ethical behavior.

UNIT 4

Applying Codes of Ethics, Conduct, and Standards of Professional Practice

4.1. Introduction

This unit discusses the fundamentals of codes of ethics, conduct, and standards of professional practice. A code of ethics and professional conduct outlines the ethical principles that govern decisions and behavior at a company or organization. They give general outlines of how employees should behave, as well as specific guidance for handling issues like harassment, safety, and conflicts of interest.

A code of ethics is broader, providing a set of principles that affect employee mindset and decision-making. A code of conduct offers principles defining the ethics of a business, but it also contains specific rules for employee actions and behavior

Learning outcomes: After training this unit, you will be able to:-

- Explain the purpose of professional codes of conduct and practice.
- Identify the main types of codes of ethics.
- Explain the difference between a code of ethics and a code of conduct.

Key Terms and Concepts

- Management, Assessment, objectivity, truthfulness, conflict, compliance, code of conduct, ethical code, integrity, harassment

4.2. The Purpose of Professional Codes of Conduct and Practice

- **Explain the drive of Professional Codes of Conduct in your school**

A code of ethics is a guide of principles designed to help professionals conduct business honestly and with integrity. A code of ethics, also referred to as an "ethical code," may encompass areas such as business ethics, a code of professional practice, and an employee code of conduct. A well-written code of conduct clarifies an organization's mission, values and principles, linking

them with standards of professional conduct. The code articulates the values the organization wishes to foster in leaders and employees and, in doing so, defines desired behavior.

4.3. Types of Codes of Ethics

A code of ethics can take a variety of forms, but the general goal is to ensure that a professional and its employees are following the laws, conducting themselves with an ideal that can be exemplary, and ensuring that the business being conducted is beneficial for all stakeholders. The following are three types of codes of ethics found in occupation.

1. Compliance-Based Code of Ethics

For all industries, laws regulate issues such as hiring and safety standards. Compliance-based codes of ethics not only set guidelines for conduct but also determine penalties for violations. In some industries, including banking, specific laws govern business conduct. These industries formulate compliance-based codes of ethics to enforce laws and regulations. Employees usually undergo formal training to learn the rules of conduct. Because noncompliance can create legal issues for the company as a whole, individual workers within a firm may face penalties for failing to follow guidelines. To ensure that the aims and principles of the code of ethics are followed, some companies appoint a compliance officer. This individual is tasked with keeping up to date on changes in regulation codes and monitoring employee conduct to encourage conformity.

This type of code of ethics is based on clear-cut rules and well-defined consequences rather than individual monitoring of personal behavior. Despite strict adherence to the law, some compliance-based codes of conduct do not thus promote a climate of moral responsibility within the company.

2. Value-Based Code of Ethics

A value-based code of ethics addresses a company's core value system. It may outline standards of responsible conduct as they relate to the larger public good and the environment. Value-based ethical codes may require a greater degree of self-regulation than compliance-based codes.

Some codes of conduct contain language that addresses both compliance and values. For example, a grocery store chain might create a code of conduct that espouses the company's commitment to health and safety regulations above financial gain. That grocery chain might also include a statement about refusing to contract with suppliers that feed hormones to livestock or raise animals in inhumane living conditions.

4.4. Code of Ethics among Professionals

Advisers must be registered and certified or state regulators are bound by a code of ethics. This is a legal requirement and also a code of loyalty that requires them to act in the best interest of their clients. Certified public accountants, which are not typically considered fiduciaries to their clients, still are expected to follow similar ethical standards, such as integrity, objectivity, truthfulness, and avoidance of conflicts of interest. Professionals should obey by:

- Act with integrity, competence, diligence, respect, and in an ethical manner with the public, clients, prospective clients, employers, employees, colleagues in the investment profession, and other participants in the global capital markets.
- Place the integrity of the investment profession and the interests of clients above their own personal interests.
- Use reasonable care and exercise independent professional judgment when conducting investment analysis, making investment recommendations, taking investment actions, and engaging in other professional activities.
- Practice and encourage others to practice professionally and ethically that will reflect credit on themselves and the profession.
- Promote the integrity and viability of the global capital markets for the ultimate benefit of society.
- Maintain and improve their professional competence and strive to maintain and improve the competence of other investment professionals.

All companies will have a different code of ethics with different areas of interest, based on the industry they are involved in, but the areas that companies typically focus on include: integrity, objectivity, professional competence, confidentiality, and professional behavior.

A code of ethics in business is a set of guiding principles intended to ensure a business and its employees act with honesty and integrity in all facets of its day-to-day operations and to only engage in acts that promote a benefit to society.

4.5. The Difference between a Code of Ethics and a Code of Conduct

A code of ethics is broader in its nature, outlining what is acceptable for the company in terms of integrity and how it operates. A code of conduct is more focused in nature and instructs how a business' employees should act daily and in specific situations. A code of ethics is a guiding set of principles intended to instruct professionals to act in a manner that is honest and that is beneficial to all stakeholders involved. A code of ethics is drafted by a business and tailored to the specific industry at hand, requiring all employees of that business to adhere to the code.

4.6. Applying Code of Conduct

A code of conduct applies to everyone within an organization, with each company having their own code of conduct that employees need to follow. A code of conduct is created by the employer to let their staff members know what is expected of them in terms of behavior in the workplace. A code of conduct is the most common policy within an organization. This policy lays out the company's principles, standards, and the moral and ethical expectations that employees and third parties are held to as they interact with the organization.

Some Codes of Conduct sets the values and principles that we as employees follow in our interactions with each other and with our stakeholders such as customers and other business partners, our shareholders and the regulatory authorities. It forms the basis for our behavior and for the public image.

4.7. Standards of Professional Practice

The Standards of Professional Practice are an agreed upon set of ethical and professional standards. Members of the professions use these standards in developing their own codes and guide them daily as they continue their work. These include the following:

- A. Professional Services and Agreement with Institutional Mission and Goals
- B. Employment Relationships and Management of Institutional Resources
- C. Conflict of Interest and Legal Authority
- D. Equal Consideration and Treatment of Others
- E. Professional Behavior and Integrity of Information and Research
- F. Confidentiality and Research Involving Human Subjects
- G. Representation of Professional Competence and References
- H. Selection and Promotion Practices
- I. Job description and performance evaluation
- J. Campus Community and Professional Development
- K. Assessment

Unit Summary

Codes of ethics and conduct outline ethical principles that govern decisions and behavior at a company or organization. They provide general guidelines for employees to behave and provide guidance for handling issues like harassment, safety, and conflicts of interest. A code of conduct is broader, encompassing principles defining the ethics of a business and specific rules for employee actions and behavior.

There are three types of codes of ethics: compliance-based, value-based, and non-compliance-based. Compliance-based codes set guidelines for conduct and determine penalties for violations, while value-based codes address a company's core value system. Non-compliance-based codes may not promote a climate of moral responsibility within the company. Non-compliance-based codes may require more self-regulation than compliance-based codes.

In conclusion, codes of ethics and conduct are essential tools for organizations to ensure ethical behavior and compliance. Understanding the purpose, types, and differences between codes of ethics and conduct can help organizations better manage their ethical practices and ensure the well-being of their employees.

Unit Review Questions

Direction: Write Short Answer for the Following Questions

1. Explain the difference between a code of ethics and a code of conduct.
2. List out the main types of codes of ethics.
3. Explain the purpose of professional codes of conduct and practice.

UNIT 5

Identifying Mechanisms of Professional Ethics Decision Making

5.1. Introduction

This unit discusses the fundamental factors of ethical decision-making frameworks and theories. Ethical decision-making is the process by which you aim to make your decisions in line with a code of ethics. To do so, you must seek out resources such as professional guidelines and organizational policies and rule out any unethical solutions to your problem. Making ethical decisions is easier said than done.

Ethical decision-making is often guided by ethical frameworks or theories, such as utilitarianism, deontology, virtue ethics, or the principle of respect for persons. It requires critical thinking, empathy, and consideration of various perspectives to arrive at a well-reasoned and morally justifiable decision. Ethical decision-making refers to the process of evaluating and choosing among alternatives in a manner consistent with ethical principles. In making ethical decisions, it is necessary to perceive and eliminate unethical options and select the best ethical alternative. Integrity, respect, responsibility, fairness, compassion, courage, and wisdom are the seven principles of ethical decision-making

Learning outcomes: After training this unit, you will be able to:

- Identify mechanisms for ethical decision-making.
- Explain the important principles for making ethical decisions.
- Mention some steps to ethical thinking and ethical behavior.
- Identify the factors that impact professional and ethical decision-making.
- List out industry compliance and regulations.

Key Terms and Concepts

- Rationalization, philosophy, induction, deduction, monitoring, value of nature, empathy, procedure, respect, recruitment, safety, and compliance

5.2. Ethical Decision Making

An ethical decision is one that engenders trust, and thus indicates responsibility, fairness and caring to an individual. Ethical decision-making requires a review of different options, eliminating those with an unethical standpoint, and then choosing the best ethical alternative. Ethical decisions generate and sustain trust; demonstrate respect, responsibility, fairness and caring; and are consistent with good citizenship. These behaviors provide a foundation for making better decisions by setting the ground rules for our behavior.

Decision-Making is necessary in order to deal with conflicting duties, loyalties or interests create moral dilemmas requiring decisions to be made and ethical decision-making involves the ability to distinguish right from wrong along with the commitment to do what is right. Workers are expected to make ethical and sound decisions in the workplace. Decisions should be made on the basis of established rules, regulations, procedures and practices.

Making good ethical decisions requires a trained sensitivity to ethical issues and a practiced method for exploring the ethical aspects of a decision and weighing the considerations that should impact our choice of a course of action. Having a method for ethical decision-making is essential. When practiced regularly, the method becomes so familiar that we work through it automatically without consulting the specific steps. Ethical decision-making is often guided by ethical frameworks or theories, such as utilitarianism, deontology, and virtue ethics.

The word utility is used to mean general well-being or happiness. Utility is the consequence of a good action. Utility, within the context of utilitarianism, refers to people performing actions for social utility. With social utility, he means the well-being of many people.

Utilitarianism is a theory of morality that advocates actions that foster happiness or pleasure and oppose actions that cause unhappiness or harm. When directed toward making social, economic, or political decisions, a utilitarian philosophy would aim for the betterment of society as a whole.

Deontology is an ethical theory that uses rules to distinguish right from wrong. Deontology is often associated with philosophy that ethical actions follow universal moral laws, such as “Don't lie. Don't steal. Don't cheat.”

Virtue ethics is the quest to understand and live a life of moral character. This character-based approach to morality assumes that we acquire virtue through practice.

Ethical decision making is first based on the distinction between statements that are factual and those that are value-based. Suggested information may be divided into two (true and untrue). It is wise to base decisions on truth and morally accepted facts. Ethical decision making is the process of identifying a problem, generating alternatives, and choosing among them maximizing ethical values while also achieving the intended goal. In short, ethical decisions should be fully informed, aware of laws, principles and rules, consciously and reflectively deliberated, intelligently, justifiable, and effectively implemented.

Everything we do, or don't do, is a choice that can affect our lives and the lives of others. Ethical choices are based on principled decisions, not on self-interest or easy-returns. Such principles are based on universal values that have been held across time, culture, politics, religion and ethnicity. Any decision can be evaluated in terms of these universal values or core ethical principles trustworthiness, respect, responsibility, fairness, caring and citizenship before making any decision the responsible person should consider the following points:

- There should be accurate and comprehensive information. As in any other decision-making process, facts and evidence must be organized beforehand.
- Relevant policy and legislation have to be considered
- The advice of others (i.e., professionals in a given area) should be sought before making any final decision.

In making ethical decision, it is necessary to:

- Notice and eliminate unethical options -right vs. wrong. Ethical thinking requires a sensitivity to perceive the ethical implications of decisions.
- Evaluate complex, ambiguous and incomplete facts. It is often difficult to obtain all necessary information.
- Select the best ethical alternative. Resolve any ethical dilemmas-right vs. wrong. Not all ethical responses to a situation are equal.

- Have ethical commitment, ethical consciousness, and ethical competency. Ethical thinking and decision making takes practice

5.3. Rationalization

Studying ethics involves attempting to find valid reasons for the moral arguments that we make. Most people already have general ideas – or what philosophers call intuitions ‘or Presumptions– about what they think is right ‘or wrong ‘. But a philosophical approach to ethics requires people to think critically about the moral ideas that they hold, to support or refute those ideas with convincing arguments, and to be able to articulate and explain the reasons and assumptions on which those arguments are based. The real value of discussing and debating ethical questions is not to win the argument 'or to score points 'against the other person! It is more important to provide carefully considered arguments to support our ideas, and to allow for rational – and deeper – understanding of the reasons underlying our beliefs, ideas and attitudes. Crucially, this requires careful listening to, analysis of and learning from the arguments that others make. Three forms of critical reasoning that individuals can use to justify their arguments are outlined below:

- Reasoning by analogy /comparison/ explains one thing by comparing it to something else that is similar, although different.
- Deductive reasoning applies a principle to a situation. For instance, if every person has human rights, and you are a person, then you have human rights like every person.
- Inductive reasoning involves providing evidence to support a hypothesis.

5.4. Some Steps to Ethical Thinking and Ethical Behaving

Steps in ethical thinking and behaving include the following:

- Clarify/ identify the relevant facts of the case/: Determine precisely what must be decided. What are the alternatives? Eliminate any impractical, illegal or improper alternatives.
- Assess/ identify the relevant ethical principles /- Separate facts from beliefs, desires, theories and opinions. Assess the influence of personal and/or collective world views on

assumptions about 'fact'. Assess the credibility of the sources of information and the motivations of the stakeholders.

- Decide/. Identify other relevant ethical principles and resolve conflicts between them /- Are there some right vs. wrong choices? Classify any ethical dilemmas involving right vs. right choices and evaluate the viable alternatives by prioritizing the ethical values so that you can choose which values to favor.
- Implement/ Decide on ethical principles and standards which are relevant to the case at hand - Develop a plan to implement your decision in a way that maximizes the benefits and minimizes the costs and risks. Involve as many stakeholders as possible during implementation.
- Monitor/ Reconsider any remaining conflicts between the case and ethical principles and standards. /- Monitor the effects of decisions and be prepared to take alternative action based on new information.
- Reflect/ identify whether the decision would withstand public scrutiny, and finally make decisions/ - Review your decision making process. Will I do it differently next time? Were you fully aware of your own values and worldview during the process? What feedback should you seek?

5.5. Important principles for making Ethical decision

- Empathize with another. Put yourself in the other person's shoes and understand how they are feeling. Treat other people the way you would like to be treated.
- Demonstrate selflessness. Do not be selfish and put yourself before others. Do the right thing even when it might not be what you really want to do.
- Be fair. Ethical people are compassionate and caring. They are always honest and fair when dealing with others.
- Respect another's opinions and choices even when you disagree with them.
- Value nature. Do not view it as only a resource for sustaining life, but as a life force in and of itself.
- Act responsibly. Be a trustworthy and responsible person that others can rely on.

There are the 7 steps to ethical decision making. The seven steps are as follows.

- Determining whether there is an ethical dimension to the issue requiring a decision.
- Collecting relevant information for ethical analysis.
- Evaluating information collected on the basis of whether the decision to be made will be in compliance with established regulations and values.
- Considering alternatives that can be made in the process to ensure the decision and the result are ethical.
- A decision should be made and implemented after the considerations.
- The final step is the review of the consequences resulting from the decision.

5.5.1. Factors that affect Professional and Ethical Decision

The code of ethics usually includes the six universal moral values that state you expect employees to be:

- A. Trustworthy: Worthy of confidence specifically: being or deriving from a source worthy of belief or consideration for evidentiary purposes a trustworthy informant.
- B. Respectful: Respect for persons may perhaps be the most fundamental principle in all of ethics. Respect (full) calls on each and every one of us to respect the intrinsic dignity of all other people. If something is intrinsic to us, it is essential to our being and cannot be earned. It is a property of being a person.
- C. Responsible: Responsibility is an ethical concept that refers to the fact that individuals and groups have morally based obligations and duties to others and to larger ethical and moral codes, standards and traditions.
- D. Fair: Fairness is concerned with actions, processes, and consequences, which are morally right, honorable, and equitable. In essence, the virtue of fairness establishes moral standards for decisions that affect others. Fair decisions are made in an appropriate manner based on appropriate criteria.
- E. Caring: The ethics of care is a normative ethical theory that holds that moral action centers on interpersonal relationships and care or benevolence as a virtue.

- F. Good citizens: Celebrating diversity and differences; go to local ethnic festivals and introduce your child to friends who represent a variety of lifestyles, cultures and religions.

5.6. Dress codes and code of conduct

A dress code is a set of rules, often written, with regards to what clothing groups of people must wear. Dress codes are created out of social perceptions. All clothing should be clean, ironed and in good shape. Refrain from wearing clothes that have tears, rips or holes, even if it is the current fashion. All employees should maintain an acceptable level of bodily hygiene to ensure that interactions with other staff and clients remain positive and pleasant. It is a standard set to guide what is appropriate to wear under certain circumstances. Dress codes include social perception, norms, and purposes. The implementation of dress codes creates orderliness and safety. Dress code identifies you that you belong to the group, gives you a sense of belongingness.

A code of conduct is a set of rules outlining the norms, rules, and responsibilities or proper practices of an individual party or an organization. An ethical code of conduct is a guide to principles created to assist practitioners in performing business in the right way. A code of ethics can reinforce the values stated in the mission and vision of the company. A Code of Conduct applies the Code of Ethics to various relevant situations. A rule in the Code of Ethics might state that all employees will obey the law. A Code of Conduct might list several specific laws relevant to different areas of organizational operations, or industry, that employees need to obey.

5.7. Workplace policies and procedures

- Code of conduct.
- Recruitment policy.
- Internet and email policy.
- Mobile phone policy.
- Non-smoking policy.
- Drug and alcohol policy.
- Health and safety policy.
- Anti-discrimination harassment policy

5.8. Industry Compliance and Regulations

Disciplinary actions

Professional practice standards means the set of documents that specify the legal and ethical requirements for professional practice that include the standards of practice and essential competencies for code of ethics, practice guidelines, regulations and bylaws.

There are three important factors that can influence ethical decision making, which are individual, organizational, and opportunity factors. All three of these factors can weigh heavily on a person during the decision-making process, especially in the workplace. No matter how a person comes to make a decision, there are usually three factors that influence a person's ethical decision-making process.

The first factor in the ethical decision-making process is called the individual. Individual factors can affect a person greatly when making ethical or unethical decisions. Every person will have a slightly different belief when it comes to individual factors. It is always best to talk with an employee or consult the code of ethics before making individual decisions.

The next factor is the organizational factor. The organizational factor can be defined as a set of values or norms that is shared by members or employees of an organization. Organizational factors can affect decision making if all of the employees or associates have negative views and make unethical decisions. Organizational factors can be affected by the people closest to the decision maker. If the organization and all of the employees believe in a code of ethics, then most likely the decision will be ethical.

The last factor is the opportunity factor. This can be defined as a situation that encourages or discourages a person. Opportunity factors are based on whether there are opportunities to make ethical or unethical decisions. If the company promotes to do whatever it takes.

Unit Summary

Ethical decision-making is the process of evaluating and choosing among alternatives in a manner consistent with ethical principles. It requires critical thinking, empathy, and consideration of various perspectives to arrive at a well-reasoned and morally justifiable decision. The seven principles of ethical decision-making are integrity, respect, responsibility, fairness, compassion, courage, and wisdom.

Ethical decision-making is necessary to deal with conflicting duties, loyalties, or interests, and requires the ability to distinguish right from wrong and the commitment to do what is right. Workers are expected to make ethical and sound decisions in the workplace, based on established rules, regulations, procedures, and practices.

Ethical decision-making is often guided by ethical frameworks or theories, such as utilitarianism, deontology, and virtue ethics. Utility refers to general well-being or happiness, while deontology uses rules to distinguish right from wrong. Virtue ethics focuses on understanding and living a life of moral character.

Unit Review Questions

Direction 1- Write Short Answer for the Following Question

1. List down the universal moral values that are expected from an employee.
2. What is rationalization?
3. Identify some steps to ethical thinking and ethical behavior.
4. Explain the important principles for making ethical decisions.
5. Identify the factors that impact professional and ethical decision-making.
6. Identify workplace policies and procedures.
7. List out industry compliance and regulations.

UNIT 6

Promoting Ethical Standards, Practice, and Employability Skills

6.1. Introduction

In this module, we are discussing the development of good interpersonal and employability skills that are tantamount to success in your life and career. In today's always-connected world, everyone has immediate access to technical knowledge. Thus, people skills are even more important now. Among the most useful skills are:

- Influence: Wielding effective persuasion tactics.
- Communication: sending clear messages.
- Leadership means inspiring and guiding groups and people.
- Change catalyst: initiating or managing change.
- Conflict management: understanding, negotiating, and resolving disagreements.
- Building bonds: nurturing instrumental relationships.
- Collaboration and cooperation: working with others toward shared goals.
- Team capabilities: Creating group synergy in pursuing collective goals

Employability is “a set of achievements, skills, understandings, and personal attributes that make graduates more likely to gain employment and be successful in their chosen occupations, which benefits themselves, the workforce, the community, and the economy.” Employability, therefore, is not just about getting a job; it is about a broader set of skills and attributes that will enable a graduate to be successful throughout their working life.

Learning outcomes: After training this unit, you will be able to:-

- Identify ethical standards, practices, and employability skills.
- Explain the ways to improve an organization's ethical climate.
- Explain the importance of employability skills.

Key Terms and Concepts

- Employee, Employability, Ethical Decision, Ethical Thinking, Ethical Behaving, Initiative, Digital Skills, Negotiation, Leadership, Empowerment, Communication, Bad Patterns,

6.2. Promoting Ethical Standards and Practice

In order to promote Ethical standards and practice with clients, colleagues and others the following behaviors should be considered:

a) Be a Role Model and Be Visible

Employees look at top managers to understand what behavior is acceptable. Senior management sets the tone for ethics in the workplace.

b) Communicate Ethical Expectations

An organizational code of ethics can reduce ethical ambiguities. The code of ethics should state the organization's primary values and the ethical rules that employees are expected to follow. Managers should remember that a code of ethics is worthless if leaders fail to model ethical behaviors.

c) Offer Ethics Training

Managers should set up seminars, workshops and similar programs to promote ethics in the workplace. Training sessions reinforce the organization's standards of conduct, to clarify what practices are and are not permissible, and to address possible ethical dilemmas.

d) Visibly Reward Ethical Acts and Punish Unethical Ones

Performance appraisals of managers should include evaluations of how actions measure up against the organization's code of ethics. Appraisals need to include how managers achieve these goals, as well as the goals themselves.

e) Provide Protective Mechanisms

The organization provides formal mechanisms that allow employees to discuss ethical dilemmas and report unethical behavior without fear of reprimand that could include developing roles for ethical counselors, ombudsman or ethical officers.

f) Enhancing the Workplace Model Behavior

A better workplace requires leaders to model behavior in every aspect of their role. In your career, the ability to demonstrate leadership with a sound ethical basis is essential to your success for any role.

6.2.1. Employability skills

The employability as relating to “portable competencies and qualifications that enhance an individual’s capacity to make use of the education and training opportunities available in order to secure and retain decent work, to progress within the enterprise and between jobs, and to cope with changing technology and labor market conditions. Individuals are most employable when they have broad-based education and training, basic and portable high-level skills, including teamwork, problem solving, information and communications technology and communication and language skills... This combination of skills enables them to adapt to changes in the world of work.

Employability skills are defined as skills required not only to gain employment, but also to progress within an enterprise so as to achieve one’s potential and contribute successfully to enterprise strategic directions.

6.2.2. Core Employability Skills

Core skills refer to skills, knowledge and competencies that enhance workers ability to secure and retain a job, progress at work and cope with changes. The core skills are:

1. Basic/foundation skills
2. Vocational/technical skills

3. Professional/personal skills
4. Core work skills

Important core skills necessary for employability skills include the following though it is not an exhaustive list:

- Communication skills that contribute to productive and harmonious relations between employees and customers;
- Teamwork skills that contribute to productive working relationships and outcomes;
- Problem-solving skills that contribute to productive outcomes;
- Initiative and enterprise skills that contribute to innovative outcomes;
- Organizational skills :planning and organizing skills that contribute to long-term and short-term strategic planning;
- Self-management skills that contribute to employee satisfaction and growth;
- Learning skills that contribute to ongoing improvement and expansion in employee and company operations and outcomes; and technology skills that contribute to effective execution of tasks.
- Numeracy skills : literacy in basic arithmetic and statistical tools
- Negotiation skill: ability to engage with others
- Valuing diversity and difference
- Digital skills (technology).

6.2.3. Importance of employability skills

Why is Employability Skills Important?

These set of "job-readiness" skills are, in essence, behaviors that are necessary for every job and are essential attitudes that allow you to grow in your career and also efficiently let you:

- Connect with co-workers
- Solve problems
- Be a part of and understand your role within the team
- Make responsible choices for your job and your career
- Be independent and take charge of your career

6.2.4. Elements of Employability Skills

Employability skills are transferable skills that are useful in nearly every job. They involve the development of an expertise, knowledge base or mindset that makes you more attractive to employers. Employability skills are also often referred to as employment skills, soft skills, work-readiness skills or foundational skills. They often improve your performance, minimize errors and promote collaboration with your coworkers, enabling you to perform your role more effectively.

Some employable qualities come naturally, while others can be acquired through education, work or daily practice. You may already have some of the key employment skills, but you can work to improve those skills and develop new ones. Here are 10 common employability skills that employers look for:

1. Communication

Communication is one of the most important employability skills because it is an essential part of almost any job. The communication process involves five elements: the sender, receiver, message, medium and feedback.

2. Teamwork

Good teamwork skills refer to the ability to work harmoniously with your colleagues to achieve a shared goal. Teamwork skills such as collaboration can increase your hiring chances because you may be able to help a company reach its goals more effectively.

3. Reliability

Reliability makes you more employable because it promotes trust between you and your employer. You are a reliable employee if you can consistently complete your tasks on time, deliver quality work and make minimal mistakes. You must also be able to respond to inquiries and emails promptly and only make promises you can keep.

4. Problem-solving

Problem-solving involves identifying key issues and their implications, having a clear understanding of problems and determining the most effective solutions. For more complex problems, you need to know how to divide them into smaller parts that are easier to understand and more manageable.

5. Organization and planning

Being able to organize and plan effectively is important because it helps you and your employer save time, effort and money by improving workflow. It ensures that assignments and projects are completed on time and prevents confusion and errors that can be costly to the company.

6. Initiative

Taking initiative means recognizing a problem and solving it, preparing for a potential crisis by taking preemptive action, taking advantage of opportunities and having a positive attitude.

7. Self-management

Self-management refers to the ability to perform job duties satisfactorily with little or no supervision. For higher-level employees, it also means delegating tasks to ensure you complete them on time. Additionally, self-managed employees can motivate themselves to deliver solid work performance consistently.

8. Leadership

Employers look for good leaders because they can benefit organizations in many ways. As a leader, you play an important role in ensuring that your team shares the same vision as the company and works in unison with other teams and departments to achieve a common goal.

9. Learning

Having strong learning skills means understanding new concepts and methods quickly, taking on new tasks, adapting to change and having the tendency to improve your knowledge and skills continually.

10. Technology

Companies search for candidates with technical skills to help them use the latest technology and stay ahead of their competitors. Depending on your job, the technology skills you need may vary greatly, from word processing and sending email to video editing and using programming languages.

6.3. Ways to Improve an Organization's Ethical Climate

a) Empowerment of Employees

One way to improve the ethical climate of your organization is to give employees more power over their work. If employees have an ethical code and more control over their work outcomes, they are likely to justify your trust in them to make the right decision. Empowerment also can produce better results from employees, such as more creative solutions to business problems, when implemented effectively. Give employees a reason to act more ethically and to be more innovative – such as to keep their autonomy over their work – and you will also increase your company's efficiency.

b) Communication Policies and Procedures

Improve your communication policies and procedures so that information will flow more smoothly and frequently between employees and managers. For example, assign a manager, HR specialist or ad-hoc team to create a series of training scenarios that teach employees the right thing to do in common situations faced in your company.

c) Discipline and Consequences

You can't expect all employees to comply with the ethics policy without some kind of discipline. Enforce your consequences for violations of the ethics policy evenly, regardless of the offender's high or low status. If you don't have the time to do this, delegate the job to a manager or HR director. Be sure to maintain careful documentation of employee violations in case you get sued over disciplining or terminating an employee for an ethical violation.

d) Changing Bad Patterns/Adverse Behavior/

As you look at employee behaviors and enforce your ethics policy, you might see patterns of unethical behavior. This might indicate that some rules and procedures need to be changed in the organization, even if it affects ways of doing things that save time or money or increase profits.

From a business standpoint, the short-term losses should be outweighed by the resulting improvement in the long-term reputation of your business.

Unit Summary

Ethics is: the critical examination and evaluation of what is good, evil, right and wrong in human conduct. A specific set of principles, values and guidelines for a particular group or organization. From the very beginning, these ethics are instilled in an individual, with a large part having been played by their parents, friends, and family. Common examples may include honesty, openness, commitment, unbiased behavior, and sense of responsibility. What a person develops regarding fairness or learns during childhood remains with him all through his life and is reflected by his actions and words. No matter if he is talking to a friend or his relatives or an elderly, his ethics would be clear from what he says and how he says it. A person's personal ethics are revealed in a professional situation through his behavior. Features of Professional Ethics:

- Refrain from causing harm to others
- Be benevolent or good to others
- Be a well-wisher to others

- Respect individual freedom
- Ensure justice to all, without discrimination
- Respect law and code of conduct
- Follow noble practices like honesty, integrity, truthfulness etc.
- Free and fair to all
- Practice non-violence
- Help the people in need
- Help the poor.

Professional ethics are those values and principles that are introduced to an individual in a professional organization. Each employee is meant to strictly follow these principles. They do not have a choice. Also, this approach is imperative in professional settings as it brings a sense of discipline in people as well as helps maintain decorum in offices. Some examples may include confidentiality, fairness, transparency and proficiency. These ethics make employees responsible. Features of professional ethics:

- | | |
|---|--|
| <ul style="list-style-type: none">• Openness• Transparency• privacy• Impartial | <ul style="list-style-type: none">• Practical and unbiased• Loyal• Co-operative• Objective oriented |
|---|--|

Professionalism is the conduct, behavior and attitude of someone in a work or business environment. A person doesn't have to work in a specific profession to demonstrate the important qualities and characteristics of a professional.

In the Ethiopian context, the public working areas are expected to develop their own professional codes that address the particular nature of the profession. The values are stated as principles of ethical service delivery and are 12 in number. These are:

- | | |
|--------------------|------------------------------------|
| 1. Integrity | 5. Honesty |
| 2. Loyalty | 6. Accountability |
| 3. Transparency | 7. Serving the public interest |
| 4. Confidentiality | 8. Exercising legitimate authority |

9. Impartiality
10. Respecting the law
11. Responsiveness
12. Leadership

- In the always connected world of today, developing people skills and employability skills is essential to succeeding in both life and work.
- Nowadays, Technology is changed dramatically and creates dynamic and complex environment, as a result, computation became tough and hard both individual and company level, therefore, you young peoples who seeks job or maintain your work contract you must scale up your employability skill continually.

Unit Review Questions

Direction 1- Write Short Answer for the Following Question

1. What are ways to improve an organization's ethical climate?
2. Why are employability skills important?
3. List down a minimum of seven personal attributes that job seekers should develop in different ways.

Project Assessment

Answer the following summative assessment questions according to the instructions given below.

Activity 1. Codes of ethics and conduct

Study the following statements and indicate whether they apply to a code of ethics or a code of conduct or both.

- A. Becomes a legal agreement once signed
- B. Legitimizes ethics.
- C. Issued by a board of directors
- D. Allows members to make independent judgments about the most appropriate behavior
- E. Encourages specific behavior
- F. A mandatory set of standards that is strictly enforced
- G. Only provides guidelines for behavior during difficult situations.
- H. Ensures standard behavior throughout the organization.

Activity 2. Ethics theoretical assessment

Study the following statements and state whether they are true or false. If you answer false give a reason for your answer.

1. Being ethical is not exactly the same as following the law.
2. A code of conduct is a document that simply guides ethical decision making in all the members of an organization
3. If an employee signs a code of conduct, they enter into a legal agreement with the employer.

Activity 3. Ethics theoretical assessment

Write Short Answer for the Following Question

1. Briefly discuss two reasons ethics are important in the workplace.
2. List five aspects that are related to "personal accountability" when it comes to workplace ethics and behavior.
3. Codes of ethics and conduct are documents that encourage specific behavior in an organization. Discuss the differences between a code of conduct and a code of ethics.
5. All organizations, no matter their size, can benefit from a code of ethics or a code of conduct.
 - A. List two benefits of having a code of ethics in place.
 - B. List two benefits of having a code of conduct in place.
6. Briefly discuss three features that all professions have in common.
7. Why do you think professions and professionals are so important in society?
8. List five qualities that successful professionals have in common
9. List four of the common features of professionalism.
10. List four reasons professionalism is important in the workplace.

Activity 4 Case Study

You are in charge of a tiny, privately held company that employs close to 40 people. Important clients are going to drop by your workplace in a little while. However, your wife calls to tell you that your autistic son is participating in a dance performance at his school, which is a joyous occasion for the family. It is exceedingly tough to reschedule the arranged meeting. Furthermore, if the meeting is a success, your business will get a lot of new business. For you, the commercial agreement is not as significant as your son's performance. You request that your personal assistant inform the clients that the meeting can be rescheduled for tomorrow due to your illness.

- Identify the different personal and professional issues present in the case. Also, examine whether what was done was right or not. If not, then what could have been the best way to resolve the situation?

Activity 5. Features of professionalism in the workplace

Hidden in the block below are seven features of professionalism. After reviewing that part, see if you can find and shade them.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | O | M | M | U | N | I | C | A | T | I | O | N | U | Z | X | I | D | L | M |
| O | K | A | Z | P | O | F | F | D | R | O | N | Y | C | W | G | U | K | R | X |
| Q | R | Z | Y | Q | G | Z | V | N | I | E | L | S | X | N | U | H | W | P | O |
| C | B | X | L | P | V | H | O | N | E | S | T | Y | U | G | G | H | O | B | R |
| F | U | E | C | Z | S | N | T | M | K | N | O | W | L | E | D | G | E | A | P |
| C | C | I | N | T | E | G | R | I | T | Y | P | S | W | P | Y | N | E | W | W |
| Q | T | U | W | I | Z | H | W | O | X | O | L | X | U | Q | N | R | H | F | Y |
| C | F | D | P | X | S | X | C | O | U | R | T | E | S | Y | Y | E | G | R | J |
| F | Z | N | R | E | S | P | E | C | T | Q | W | D | C | L | A | H | L | C | P |
| Q | A | C | L | U | A | C | C | O | U | N | T | A | B | I | L | I | T | Y | S |

References

*Anderson, Benedict, (1991). *Imagined Communities: Reflections on the Origins and Spread of Nationalism*, rev. ed., London: Verso.*

*Appiah, Kwame Anthony, (2005), *the Ethics of Identity*, Princeton: Princeton University Press.*

*ECBP (2006), *National Technical and Vocational Education and Training (TVET) Strategy (draft for discussion)**

*Freeman, N. K. (1999). *Morals and character: The foundations of ethics and Professionalism*. *The Educational Forum*, 63(1)*

*Freeman, N. K. (2000). *Professional ethics: A cornerstone of teachers' pre service Curriculum. Action in Teacher Education*.*

*Lawrence Kohlberg and Richard H. Hersh, (2004) "Moral Development: A Review of The Theory," in *Business Ethics: Problems, Principles, Practical Applications*, 2nd edn. Ed. Keith W. Krasemann (Acton, MA: Copley, 2004), 308–18.*

Module IV

Web design Using HTML

Contents

| No | Contents | page |
|----------|--|------|
| | Module Description | 313 |
| 1 | UNIT 1: MARKUP LANGUAGE | 315 |
| 1.1 | Introduction to markup language | 315 |
| 1.2 | Types of markup languages | 319 |
| 1.3 | Introduction to HTML | 322 |
| | Unit Summary | 327 |
| | Unit Review Questions | 327 |
| 2 | UNIT 2: HTML DEVELOPMENT ENVIRONMENT AND HTML STRUCTURE | 330 |
| 2.1 | Introduction to integrated development environment (IDE) | 330 |
| 2.2 | Basic HTML document structure | 340 |
| | Unit Summary | 348 |
| | Unit Review Questions | 349 |
| 3 | UNIT 3: CREATE WEB CONTENT USING HTML | 355 |
| 3.1 | HTML Elements | 355 |
| 3.2 | HTML Formatting tags | 368 |
| 3.3 | HTML Links | 381 |
| 3.4 | HTML Lists | 386 |
| 3.5 | HTML table | 393 |
| 3.6 | HTML frame | 403 |
| 3.7 | HTML Form | 413 |
| 3.8 | Multimedia elements | 434 |
| | Unit Summary | 440 |
| | Unit Review Questions | 440 |
| | References | 448 |

Module Description

This module aims to support you in acquiring the essential knowledge and skills required to become a good front-end web developer. As you progress through each unit, you will develop new skills and enhance your existing ones. Each unit comprises self-check, review questions and practical activities to evaluate your comprehension before proceeding to the next level. This learning module is designed to help you to achieve the required competency in “Web design and development Level I”. It provides an introduction to fundamental concepts, setting the development environment and applies the necessary tools and resources for development syntax, focusing on the utilization of HTML elements to create content and the structure of web pages. Additionally, it explores the incorporation of multimedia content into web pages, including various methods for including images and embedding video, audio, and external web pages.

Before starting this module, you should have at least basic familiarity with using computers and using the web passively (i.e., just looking at it, consuming the content). You don't need any previous knowledge to start this Unit. we recommended that you should have a basic work environment set up as detailed in Installing basic software, can run web browsers, search google, social media network through internet, and understand how to create and manage folders and files as operate personal computer and a willingness to learn.

This module will also assist you to attain the following learning outcomes:

- Identify markup languages
- Setup the HTML development environment
- Create web pages and insert contents using HTML

Module Instruction:

Learning Instructions: How to use this Module

For effectively use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what are being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get

the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.

4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

UNIT 1

MARKUP LANGUAGE

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 1.1. Introduction to markup language
- 1.2. Types of markup languages
- 1.3. Introduction to HTML

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- Identify markup languages

Key Terms: *HTML, Markup languages, Semantic and Presentation*

Unit Overview

This unit focuses on fundamental concepts in web development that allow for the structuring and presentation of content on the internet. One of the most widely used markup languages is HTML (Hypertext Markup Language), which is responsible for defining the structure and layout of web pages. HTML uses tags to markup elements such as headings, paragraphs, images, and links, enabling browsers to interpret and display the content correctly. Other types of markup languages include XML (eXtensible Markup Language) and XHTML (eXtensible Hypertext Markup Language), each with its own specific purposes.

1.1. Introduction to markup language

Markup languages can be defined as a set of instructions or codes used to annotate and format text in a document. These languages provide a structured way to add meaning and structure to content, allowing for easier interpretation and presentation on various platforms and devices. The term "markup languages" refers to a collection of codes and instructions that are utilized to annotate and format text within a document. These languages enable the addition of structure and meaning to content, facilitating its interpretation and presentation

across different platforms and devices.

A markup language is a set of rules that defines how the layout and presentation of text and images should appear in a digital document. It allows structuring documents, adding formatting, and specifying how different elements should be displayed (or “rendered”) on webpages. This structuring helps search engines like Google understand the information on websites better. If search engines know more about what a page is about, they are more likely to show it to people who are looking for its content. Which, in result, can bring more people to websites with the right markup.



A markup language is not a programming language. It is static, not dynamic. It does not use logic. It simply indicates the style, structure, and formatting of a

1.1.1. History

Markup languages have existed since 1967. The first widely used markup language was SGML, a variant on GML that some systems still use today. Developers at IBM used SGML to create large, complex documents that could be easily read and distributed.

In 1993, the most popular markup language (HTML) appeared. Developers and designers quickly adopted it throughout the World Wide Web. XML is a more recent markup language but is still very popular.

Developed in 1996, XML is simple to read and easy to understand. Frequently, developers use XML to transport (export and import) data between different systems.

1.2.2. Features of Markup Language

When we think, “what is a markup language,” we consider some common features. If you’re wondering whether a language is a markup language, compare the language to these hallmarks:

- A markup language uses tags to annotate texts. Tags are special words or characters that indicate when formatting should be applied.
- Tags come in pairs, with an opening tag and a closing tag. For example, in HTML,

the bold tag is written as `` and ``.

- The text that goes between the opening and closing tags will be affected by the formatting. In the case of the bold tag, the document would show the text in bold font.
- Markup languages are easy to read. For the most part, anyone should be able to read and understand a markup language, even if they can't write it.
- Markup languages also have a syntax, or set of rules, that define how to use the tags. This syntax allows for consistent and structured documents.

1.1.3. Merits and Demerits of Markup languages

I. Merits

Markup languages are easy to learn and use. Even someone with no programming experience can create a basic web page using HTML.

Markup languages also aren't as strict as programming languages; if something goes wrong in HTML, you won't receive a compiler error. For that reason, experts often advise programming novices to start by learning a markup language.

II. Demerits

Markup languages are, by nature, limiting. They are not as versatile as programming languages and cannot be used to create dynamic or interactive applications.

Most serious web development requires programming languages such as PHP, Javascript, or Python.

1.1.4. Reasons to Learn a Markup Language

Millions of people worldwide use markup languages for a wide variety of purposes. Learning a markup language, such as HTML (Hypertext Markup Language) or XML (Extensible Markup Language), can be beneficial for various reasons:

- **Web Development:** Markup languages are fundamental to web development. HTML, in particular, is the backbone of web pages, defining the structure and content of a webpage. By learning HTML, you gain the ability to create and modify web pages, which is a valuable skill in today's digital age.
- **Content Creation:** Markup languages are used not only for web development but also for creating structured documents in various domains such as documentation,

publishing, and data interchange. XML, for instance, is widely used for storing and exchanging structured data in fields like finance, healthcare, and e-commerce.

- Understanding Web Technologies: Learning a markup language provides a solid foundation for understanding other web technologies like CSS (Cascading Style Sheets) and JavaScript. HTML, CSS, and JavaScript form the core technologies of the web, and having a grasp of HTML is essential for effectively working with these other technologies.
- Career Opportunities: Proficiency in markup languages can open up numerous career opportunities in web development, content creation, technical writing, digital marketing, and more. Many companies seek candidates with HTML and XML skills, making it a valuable addition to your resume.
- Customization and Personalization: Knowing how to work with markup languages allows you to customize and personalize your online presence. Whether you want to create your own website, blog, or online portfolio, HTML gives you the power to design and structure your content according to your preferences.
- Enhanced Communication: Markup languages are a universal way of structuring and sharing information. Learning HTML or XML enables you to effectively communicate and collaborate with other developers, designers, and content creators, whether you're discussing web projects or exchanging data in a standardized format.
- Adaptability and Future-Proofing: Markup languages have been around for decades and continue to evolve with the changing landscape of technology. By learning a markup language, you equip yourself with a timeless skill that can adapt to new developments and trends in the digital world.



Self-check 1-1:

Dear Students! Now it is your turn to answer the following self-check questions

- 1.How do markup languages differ from programming languages?
 - 2.What are the Importances of Markup language?
 - 3.Describe the drawbacks of Markup language?
-

1.2. Types of markup languages

Markup languages can be classified into various categories based on their purpose and functionality. These categories include:

- Presentation Markup Languages: These markup languages are primarily used to define the visual presentation of a document or webpage. Examples of presentation markup languages include HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets).
- Data Markup Languages: Data markup languages are designed to structure and organize data within a document or database. XML (eXtensible Markup Language) is a widely used data markup language that allows for the representation and exchange of structured data.
- Mathematical Markup Languages: These markup languages are specifically designed for representing mathematical equations and formulas. MathML (Mathematical Markup Language) is an example of a markup language used for mathematical notation.
- Document Markup Languages: Document markup languages are used to structure and annotate textual content within a document. Examples of document markup languages include SGML (Standard Generalized Markup Language) and LaTeX.
- Domain-Specific Markup Languages: These markup languages are tailored to specific industries or domains. For example, MusicXML is a markup language used for representing sheet music, while SVG (Scalable Vector Graphics) is a markup language for describing two-dimensional graphics.
- eXtensible Business Reporting Language (XBRL): XBRL is a specialized markup language used for financial reporting and analysis. It standardizes the representation of financial data, making it easier to compare and analyze information across different organizations and industries.

Dear Students! Are you ready to take a look at the most popular markup languages? This isn't an exhaustive list. In fact, there are countless markup languages. Because markup languages are so easy to use, many companies create and use their own. But there are a handful of extremely popular markup languages.

Table 1-1: A Comparison of HTML, XML, SGML, and TeX

| Language | Syntax | Typical Usage | Versions | Year |
|---|-------------|---|---------------------|-------------------------------------|
| HTML (Hypertext Markup Language) | Tag-based | Creating structured documents for web pages, including text, images, and links. | HTML4, HTML5 | 1997 (HTML4), 2014 (HTML5) |
| XML (Extensible Markup Language) | Tag-based | Representing and structuring data in a generic, platform- independent manner. | XML 1.0, XML 1.1 | 1998 (XML 1.0) |
| SGML (Standard Generalized Markup Language) | Tag-based | Defining markup languages and document types, serves as a foundation for HTML and XML. | SGML 1.0 | 1986 (SGML 1.0) |
| TeX (Typesetting System) | Macro-based | High-quality typesetting of technical and scientific documents. | TeX, LaTeX | 1978 (TeX), |

Typically, in web development, there are two common categories of markup that are utilized based on the structure of web content and the visual appearance or layout of its content.

Dear Students! *In order to provide you with a more comprehensive understanding of markup languages based on web contents, let us delve deeper into two distinct categories of markup languages: semantic and presentational markup.*

content of a webpage. It is used to structure the content of a web page in a meaningful way, allowing search engines and assistive technologies to understand the information. By using semantic markup, developers can improve the accessibility, Search Engine Optimization (SEO), and overall user experience of a website. It focuses on the meaning and purpose of the elements used, such as headings, paragraphs, lists, and links.

1.2.1.2. Presentational Markup

Presentational markup is used to define the visual appearance and layout of the content. It is primarily concerned with how the web page should look to the users. It involves the use of HTML tags and CSS styles to define the layout, colors, font size, positioning and other websites.

Both types of markup play important roles in creating well-structured or well-designed and accessible, visually appealing web pages.



Self-check 1-2:

Dear Students! *Now it is your turn to answer the following self-check questions. Have you answered the self-check questions? If yes, let us move on to the discussions.*

1. What is the purpose of semantic markup in web development?
2. How does semantic markup benefit web developers?

1.3. Introduction to HTML

HTML stands for **HyperText Markup Language**. It is used to design web pages using a markup language and creates a complete website structure of web pages. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML is the basic language for web content creation. It is used to create a structure or blueprint of the web documents. It uses various tags to define the different elements on a page, such as headings, paragraphs, and links.

1.3.1. HTML Versions

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by **Tim Berners-Lee** in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

Dear Students! *What is the latest version of HTML? Can you try to state the version of HTML based their function designed ? Let us explore together some of most common HTML 4 versions and We will apply most*

In this specific instance, we will thoroughly analyze the differences between different versions of HTML.

- **HTML 1.0 (1993):-** The first version of HTML, HTML 1.0, was released in 1993. It laid down the initial foundations for building web pages. However, it was relatively simple and lacked many features compared to later versions.

- **HTML 2.0 (1995):-**HTML 2.0 was officially published in 1995. It expanded upon the basic structure of HTML 1.0 and introduced new elements and features. This version marked significant progress in web development¹³.
- **HTML 3.2 (1997):-**HTML 3.2 brought more enhancements, including support for tables, forms, and image maps. It was widely used during the late 1990s and early 2000s¹.
- **HTML 4.01 (1999):-**HTML 4.01 was a major version released in late 1999. It introduced features like frames, style sheets, and improved form controls. This version played a crucial role in shaping the web landscapes ¹³.
- **HTML5 (2008):-**HTML5, the most recent and robust version, emerged in 2008. It revolutionized web development by introducing new semantic elements, multimedia support (such as <video> and <audio>), and the powerful <canvas> element for graphics and animations. Additionally, HTML5 works seamlessly with JavaScript and CSS, making it the go-to choice for modern web applications¹.

Remember, each version of HTML builds upon its predecessors, adapting to the evolving needs of web developers and users. HTML 4 and HTML 5 are two versions of the HyperText Markup Language (HTML), which is the standard language used to create and design web pages and web applications. An HTML 4 document starts with a DOCTYPE declaration and contains a HEAD and a BODY. Its elements have three parts: a start tag, content, and end tag. HTML5 is the fifth version of HTML.

Some key differences between HTML and HTML5 include support for multimedia elements (such as video and audio), improved semantics, and better support for mobile devices. It allows developers to build animations and simple to complex applications that run on the browser. HTML5 can help developers build web apps, which are developed for Entertainment, Online streaming, Video players, Audio players, etc. Almost all web browsers, including Google Chrome, Safari, Mozilla Firefox Microsoft Edge, and mobile browsers, such as Android and iOS support most of the HTML5 features.

1.3.2. Characteristics of HTML

The characteristics of HTML are as follows:

- HTML is not a programming language. It's a markup language.
- HTML is a widely used language on the web.
- HTML is used to create web pages and web applications. We can create a static website by HTML only.
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- All the HTML elements/tags are incase-insensitive.
- HTML supports a wide range of media types, including text, images, audio, and video, which makes web pages more engaging and interactive
- HTML is a flexible language that can be used with other technologies, such as CSS and JavaScript, to add additional features and functionality to a web page
- Since HTML is compatible with all browsers, web pages created in HTML are displayed across a variety of platforms and devices.

1.3.3. Features of HTML

- Easy to Learn and Use: HTML has a simple and straightforward syntax, making it relatively easy for beginners to learn. Its tag-based structure is intuitive, and there are plenty of resources available online for learning HTML.
- Platform-Independent: HTML is platform-independent, meaning it can be viewed on any device or operating system that has a web browser. Whether you're using a computer, tablet, or smartphone, HTML content will render consistently across different platforms.
- Support for Multimedia: HTML allows the embedding of multimedia content such as images, videos, and audio files directly into web pages using tags like ``, `<video>`, `<audio>`, etc. This enables the creation of visually engaging and interactive web experiences.

- Hypertext Capabilities: HTML is designed to support hypertext, which allows for the creation of hyperlinks within text. Hyperlinks enable users to navigate between different web pages or sections within the same page, enhancing the interconnectedness of web content.
- Markup Language: HTML stands for Hypertext Markup Language. As a markup language, HTML uses tags to define the structure and content of web pages. Tags are enclosed in angle brackets (< >) and are used to specify elements such as headings, paragraphs, lists, tables, forms, etc.

These features collectively contribute to HTML's popularity and widespread use as the foundation of the World Wide Web. They make it an accessible, versatile, and powerful tool for creating and presenting content on the internet.

1.3.4. The Importance of Learning HTML

- Simple Markup Language: HTML is indeed a simple markup language. Its syntax consists of tags that are easy to understand and use. Tags are enclosed in angle brackets (< >) and provide a straightforward way to structure content on a web page.
- Easy Implementation: HTML's simplicity makes it easy to implement. With just a basic understanding of its syntax, users can start creating web pages. Additionally, there are numerous online resources, tutorials, and documentation available to help beginners learn HTML quickly.
- Website Creation: HTML is the backbone of web development. It serves as the foundation for creating websites by defining the structure and content of web pages. While HTML alone can create static web pages, it is often combined with other technologies such as CSS for styling and JavaScript for interactivity to create dynamic and engaging websites.
- Fundamentals of Web Programming: Learning HTML provides a solid foundation for understanding web programming concepts. It introduces beginners to essential concepts like markup languages, document structure, elements, attributes, and the basics of client-side web development. Mastering HTML lays the groundwork for delving into more advanced web technologies and programming languages.

- Professional Career Boost: Proficiency in HTML can significantly boost one's professional career, especially in fields related to web development, design, and digital marketing. Many job roles in these industries require knowledge of HTML as a fundamental skill. Having HTML skills on your resume demonstrates your ability to create and maintain web content, which is highly valued in today's digital economy.

Overall, HTML's simplicity, versatility, and importance in web development make it a valuable skill to learn for anyone interested in pursuing a career in the digital realm.

1.3.5. Merits and Demerits of HTML

HTML offers several advantages, such as its simplicity and ease of use. It is a straightforward language that allows users to create and structure web pages without requiring extensive coding knowledge. Additionally, HTML is supported by all major web browsers, ensuring compatibility across different platforms and devices. HTML is used to build websites. It is supported by all browsers. It can be integrated with other languages like CSS, JavaScript, etc.

On the other hand, HTML also has its disadvantages. One of the main drawbacks is its limited functionality when it comes to creating dynamic and interactive web pages. HTML alone cannot handle complex tasks such as user input validation or database interactions. To overcome this limitation, developers often need to combine HTML with other programming languages like JavaScript or CSS. HTML can only create static web pages. For dynamic web pages, other languages have to be used. A large amount of code has to be written to create a simple web page. The security feature is not good.

Dear Students! Now it is your turn to answer the following self-check questions.

Self-check 1-3:



Dear Students! Read the following self-check questions and engage in a discussion about each.

- Which person is given credit for developing HTML?
- What are the distinguishing features of HTML?
- What are the notable traits of HTML?
- What are the major limitations of HTML?

- Distinguish the difference between HTML4 and HTML5?
-

Unit Summary

Markup languages are instructions used to annotate and format text in a document, providing structure and meaning to content. They are not programming languages and are static, indicating the style, structure, and formatting of a document. Different types of markup languages include presentation, data, mathematical, document, domain-specific, and XBRL. HTML is a widely used markup language for web development, with HTML5 being the latest version. It is easy to learn, widely used, and supports various media types. Learning HTML is beneficial for creating websites and web applications, and it can be combined with CSS and JavaScript for additional functionality.

Unit Review Questions

Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What is the primary purpose of a markup language?
 - A. To define the style, structure, and formatting of a document
 - B. To animate objects on the screen
 - C. To manage databases
 - D. To compile and execute code
 - E. To perform complex calculations
 2. Which of the following is NOT true about markup languages?
 - A. Markup languages are static
 - B. They are considered programming languages
 - C. The first widely used markup language was SGML
 - D. They use tags to annotate text
 - E. Markup languages are easy to read
 3. What was the first widely used markup language?
 - A. XML
-

- B. SGML
 - C. LaTeX
 - D. HTML
 - E. Markdown
4. What is an advantage of markup languages?
- A. They require compilation
 - B. They are easy to use and learn
 - C. They are difficult to learn
 - D. They can create dynamic applications
 - E. They are primarily used for animation
5. For what purposes are markup languages NOT typically used?
- A. Formatting documents
 - B. Academic research
 - C. Storing data
 - D. Creating websites
 - E. Creating interactive applications
6. Which of these is a type of markup language?
- A. C++
 - B. Swift
 - C. Python
 - D. XBRL
 - E. Java
7. What distinguishes semantic markup from presentational markup?
- A. Semantic markup focuses on meaning and structure
 - B. Presentational markup focuses on meaning and structure
 - C. There is no difference between semantic and presentational markup
 - D. Semantic markup focuses on visual appearance
 - E. Presentational markup can't be used in HTML
8. What is a key characteristic of HTML5 compared to previous versions?
- A. It is less widely used
 - B. It introduced improvements and new features
-

- C. It is not compatible with browsers
 - D. It does not support multimedia
 - E. It has fewer features
9. Why is HTML considered easy to learn?
- A. It is platform-dependent
 - B. It is similar to C++
 - C. It has a simple syntax that is easy to read
 - D. It has complex syntax
 - E. It requires understanding of machine language
10. Which of the following technologies is commonly used with HTML to boost web page functionality?
- A. Perl
 - B. SQL
 - C. Bash
 - D. CSS
 - E. C

UNIT 2

HTML DEVELOPMENT ENVIRONMENT AND HTML STRUCTURE

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

1. *Introduction to integrated development environment/ IDE/*
2. *Basic HTML document structure*

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- Setup the HTML development environment
- Create HTML pages

Key Terms: *IDE, Text-Editor, Notepad++, VS code, SEO, Metadata, and Tool*

Unit Overview

This unit focuses on an integrated development environment (IDE), which is a software application that offers a wide range of tools and functionalities for developers to write, test, and debug their code. It provides a convenient and effective approach to software development by merging different programming tools into a unified interface. IDEs are extensively utilized for programming languages such as HTML, a markup language employed for crafting web pages. By utilizing an IDE, developers can effortlessly write HTML code, preview the resulting output, and efficiently handle their projects, thereby enhancing the efficiency and productivity of the development process.

2.1. Introduction to integrated development environment (IDE)

In the world of web development, setting up a development environment is an important first step for any project. An integrated development environment (IDE) is a software platform that offers a wide range of tools and functionalities to support the development of software applications. In the case of web development, the development environment provides the

necessary tools and resources that allow developers to create HTML code. An HTML integrated development environment (IDE) is a software tool that provides a comprehensive platform for developing HTML code.

Definitely! In web development, an integrated development environment (IDE) provides a convenient and efficient workspace for creating HTML code and building web applications.



Code Editing: IDEs offer robust code editors with features like syntax highlighting, autocompletion, and error checking. Developers can write and edit HTML, CSS, and JavaScript code seamlessly.

HTML editor is a software used for writing code in HTML, which is used for structuring and creating websites. Even though codes can be written from scratch using a normal text editor, HTML editors provide a great deal of ease to the developers by ensuring hassle-free coding. Setting up a development environment for HTML is important for several reasons.

- Firstly, it provides a platform for developers to create and test their HTML code. Without a development environment, developers would have to make changes to their code and upload it to a live website every time they want to test their changes. This process is time-consuming, and it can be risky if the changes break the website.
- Secondly, a development environment allows developers to work on their code in a controlled environment. This means that they can experiment with new techniques and features without the risk of breaking a live website. They can also work on their code without worrying about disrupting other team members who are working on the same project.
- Development environment makes the development process more efficient and productive. Without a proper environment, developers would have to manually test and debug code in a live production environment, which is both time-consuming and risky. A development environment allows developers to test their code locally before deploying it to a live site, minimizing errors and bugs.

2.1.1. Tools Required in web design

The necessary tools for establishing an HTML development environment. Setting up an HTML development environment is crucial for efficient web development. Here are the essential tools you'll need: Code editors and Browsers.

I. Code Editor tools

First, you'll need a good code editor is the foundation of your development environment. It provides syntax highlighting, auto-completion, and other features to enhance your coding experience. it is a program that allows you to write and edit code. These tools enhance productivity, reduce manual tasks, and provide a comfortable environment for web developers. Each has its own features and benefits, so it's important to choose one that suits your needs and preferences.

Some common popular IDEs for HTML web development include notepad++, Visual Studio Code (VS Code), Adobe Dreamweaver CC, Atom, WebStorm, and Sublime Text.

There are broadly two types of HTML Editors:

1. Textual HTML Editor

These are text-based editors where the developers can write their codes and compile them. The code appears in the same manner we write it, thus it requires basic knowledge of HTML. Some of these editors also provide features of making a project, managing all the files related to the web, etc. Examples of HTML Text editors include-Notepad++, VSCode,Sublime Text.

2. WYSIWYG HTML Editor

'What you see is what you get' is its full form. WYSIWYG are editors that provide the preview of the output of the source code i.e. as it would appear on a browser. There is a drag and drop feature available in most of them that eases the handling. It does not require any hardcore knowledge of HTML, thus enabling non-technical to easily develop websites. Examples include-Adobe Dreamweaver, Amaya, BlueGriffon, etc.

3. Browser Tools

Next, you'll need a web browser. A web browser is the program that you use to view web pages. It's also essential for testing HTML code. Popular browsers include Google Chrome, Mozilla Firefox, and Apple Safari. Like text editors, each browser has its own unique

features and quirks, so it's a good idea to test your code in multiple browsers.



Remember, the right tools can significantly boost your productivity and creativity. Choose the ones that align with your workflow and preferences!

While HTML can be written in basic text editors like Notepad, there are several other commonly used code editors that offer more features and functionality. A comparison table of these editors is provided below:

Table 2-1: Comparison development Environment

| Editor | Key Features | Perks to Developers |
|--------|---|--|
| Atom | <ul style="list-style-type: none"> It has 81 built-in packages and 8,700 additional packages that can be installed. Since it's an open-source text-editor, its code is available on GitHub and can be used by the developers to extend and edit its source code. Atom supports multiple sections within a single interface thus helping the developers to write code side by side. It also provides flexible auto-completion and syntax correction along with cross-platform editing. | <ul style="list-style-type: none"> The teletype feature of Atom lets developers interact with their teammates and share their codes in real time. The feature of cross-platform editing enables developers to work across different Operating Systems. Developers can install new packages and build their own. Smart auto-completion facilitates faster and smoother coding. <p>Multiple panes help to edit and compare code across different files</p> |

| | | |
|-----------------|---|---|
| | | |
| Notepad ++ | <ul style="list-style-type: none"> It is a lightweight software i.e. smaller program size. It is also available as an application on mobiles and its source code is available on GitHub. Since it is an open-source code editor, it is extendable and the users can add functionalities as per their needs or add plugins from different communities. It is customizable thus allowing the users to personalize the features as per their preferences. | <ul style="list-style-type: none"> It is a super lightweight editor with color-coding. Developers can customize its interface according to their preference. Due to its extensibility, developers can add plug-ins and create their own. |
| Sublime Text | <ul style="list-style-type: none"> It has the provision of simultaneous editing i.e. it is possible to apply changes to several domains at a particular time. It provides many actions such as- command palette, file switching,goto symbols,multi-edit,alignment,bracket-highlighter,etc. Sublime falls under the category of freemium software i.e. we need to pay additional money to enjoy | <ul style="list-style-type: none"> The ‘Goto Anything’ feature enables developers to open files with just a few keystrokes and immediately go to words,lines or symbols. The feature of multiple selections enables developers to make many changes at the same time such as manipulate many lines at once. Split Editing permits editing of files side-by-side which escalates the rate of development. Easy Customization in Sublime using simple JSON files. |

| | | |
|----------------------|--|--|
| | <p>all its features.</p> <ul style="list-style-type: none"> Some of the packages of sublime are open-source and are available on git. | |
| Adobe Dreamweaver CC | <ul style="list-style-type: none"> It has proven to be an important tool in the domain of web-designing using the aesthetic cloud libraries of Adobe promoting colors, graphics, layers, etc. It can be used as a text as well as WYSIWYG (what you see is what you get) editor i.e. code can be edited with or without a visual guide. This gives the developers additional features of previewing the end results of the code. It is useful for full-stack development. | <ul style="list-style-type: none"> Adobe Dreamweaver provides a dynamic display and builds web-pages that can fit on any screen. The feature of preview enables the developers to know how exactly their web page would look, before publishing it. The workspace is customizable, enabling the developers to see only the tools they need in order to code. Multi-monitor support feature for Windows helps to view web pages on multiple monitors. |
| Visual Studio Code | <ul style="list-style-type: none"> It has the features of built-in Git commands, debugging of code right from the editor, live preview of the web from server, smart code-completion using IntelliSense and many | <ul style="list-style-type: none"> VS Code has an attractive User-Experience and a customizable workspace. IntelliSense provides smart completions based on variables, functions and imported libraries. With VS Code, it is possible to |

| | | |
|--|--|---|
| | <p>more.</p> <ul style="list-style-type: none"> • It's a free and open-source software that is one of the most popular developers' tool. • The most prominent feature of Visual Studio Code is its interactive debugger which enables execution of debugging commands on the console. • IntelliSense of VS Code has proven to be extremely beneficial for code understanding, navigation and refactoring. | <p>debug the code right from the editor, without any hassle.</p> <ul style="list-style-type: none"> • Git commits can be made right from the editor. • Extensions are available to add new languages, debuggers or any other additional services. They run as separate processes, thus not at the cost of speed, of the editor. |
|--|--|---|

Advantages of using HTML Editors

- They are of great benefit since they allow the users to easily check their syntax, insert commonly used HTML tags and structures and also provide auto-completion.
- The code generated through an HTML editor can be translated to other languages such as XML, JavaScript, etc. For example-NVU editor provides this translation functionality.
- Website development can be very exhausting and cumbersome. With the help of online HTML editors, it is possible to create websites with ease and at a faster rate.
- HTML editors provide full control to the developer, hence helping him to delve deeper into the source code and find the hidden intricacies.
- HTML editors provide an amicable and aesthetic designing experience.

2.1.2. Installing required tools and extensions

I. Software Requirements for Web Development

Most operating systems come with a basic text editor. However, if you want additional features like syntax coloring, auto-completion, and code search, consider using third-party

editors.

Some popular text editors include:

- *Windows*: Notepad, Notepad++, Visual Studio Code, WebStorm, Brackets, ShiftEdit, Sublime Text.
- *Mac OS*: TextEdit, TextWrangler, Visual Studio Code, Brackets, ShiftEdit, Sublime Text.
- *Linux*: Vi (for all UNIX systems), GEdit (Gnome), Kate (KDE), LeafPad (Xfce), Emacs, Vim, Visual Studio Code, Brackets, ShiftEdit, Sublime Text.
- *ChromeOS*: ShiftEdit.

II. Hardware Requirements for Web Development:

HTML5-capable browsers vary in their hardware requirements. However, several lightweight devices can run HTML5 browsers effectively:

It lightweight and should easily run on today's hardware of Personal Computer. We recommend:

CPU >1.6 GHz or faster processor and >1 GB of RAM

- Post-PC Devices: Devices like iPods, iPads, iPhones, various Android devices, and Blackberries can run lightweight, full-featured HTML5 browsers based on projects like WebKit. These devices have low hardware specifications but can handle HTML5 content, including video playback and JavaScript transformations.

III. Extensions for web development:

Additionally, as a web developer, having the right browser extensions can significantly enhance your productivity and streamline your workflow. Here are some must-have Chrome extensions for both developers and designers that can enhance your web development experience:-

- *Web Developer*: A suite of tools within a single Chrome extension. It includes various useful tools for both general developers and web developers
- *Wappalyzer*: Although not a Chrome extension, it's worth mentioning. Wappalyzer helps you identify the technologies used on any website, including programming languages, frameworks, and content management systems . It provides insights into

the stack used, including programming languages, frameworks, and content management systems.

- *HTML Validator:* Check your HTML code for errors and ensure proper syntax. This extension helps you maintain clean and valid HTML markup
- *Web Developer Checklist:* Get a checklist of best practices for web development. It covers various aspects like performance optimization, accessibility, and security .



Operation Sheet : 2-1

Title: Installing Notepad++ Text Editor

Objective: This operation sheet provides step-by-step instructions on how to Install Notepad++ with straightforward process

Instructions: Follow the below steps to install Notepad++ on Windows.

Step 1: Download. Visit <https://notepad-plus-plus.org/downloads/> to download Notepad++ latest version.

Step 2: Install. Click the installer to begin installing Notepad++.

Step 3: Open. Click the icon of Notepad++ to open it.



Note:-Remember that you can download most of these programs and extensions for free, so explore them and find what works best for



Operation Sheet : 2-2

Title: Installing Visual Studio Code

Objective: This operation sheet provides step-by-step instructions on how to Installing Visual Studio Code with straightforward process

Instructions: Follow the below steps to Installing Visual Studio Code on Windows.

Step 1: Step 1: Visit the official website of the Visual Studio Code using any web

browser like Google Chrome, Microsoft Edge, etc.

Step 2: Press the “Download for Windows” button on the website to start the download of the Visual Studio Code Application.

Step 3: When the download finishes, then the Visual Studio Code icon appears in the downloads folder.

Step 4: Click on the installer icon to start the installation process of the Visual Studio Code.

Step 5: After the Installer opens, it will ask you for accepting the terms and conditions of the Visual Studio Code. Click on **I accept the agreement** and then click the **Next** button.

Step 6: Choose the location data for running the Visual Studio Code. It will then ask you for browsing the location. Then click on **Next** button.

Step 7: Then it will ask for beginning the installing setup. Click on the **Install** button.

Step 8: After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.

Step 9: After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the “**Launch Visual Studio Code**” checkbox and then click **Next**.

Step 10: After the previous step, the **Visual Studio Code** window opens successfully.

Dear Students! Have you answered the self-check questions? If yes, let us move on to the discussion of the HTML and its text editor.

Self-check 2-1:



1. When should you use an HTML Editor?
2. What is the Difference Between Text HTML Editors and WYSIWYG HTML editors?
3. What are the common features of an HTML editor?

2.2. Basic HTML document structure

HTML document structure

An HTML document contains the text of the page itself and HTML tags, which define the structure and appearance of the document. It also contains hyperlinks to other pages or to include multimedia elements such as audio, video, animations etc. Entire HTML document is bound within a pair of <html> and </html> tags.



HTML tags are generally have an opening and closing tag surrounding the text they affect. It is written between < > brackets and HTML elements consist of several parts, including the opening and closing tags, the content, and the attributes.

HTML depicts each element as shown below Table 2-2:

Example 1:Basic HTML structure

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Table 2-2: Description of The basic HTML structure

| No | HTML element | Description |
|----|-----------------|--|
| 1 | <!DOCTYPE html> | This declaration specifies the version of HTML that the document is written in. It is usually the first line of an HTML file |

| No | HTML element | Description |
|----|------------------|--|
| 2 | <html> element: | This is the root element of the document and encloses all other elements |
| 3 | <head> element | This element contains metadata about the document, such as the title, author, and character set. It does not contain any visible content on the page |
| 4 | <Meta> element: | This element is HTML elements that provide information about a webpage to search engines and website visitors. It is used to provide metadata about a webpage, such as its title, description, and keywords, which can improve its visibility in search engine results |
| 5 | <title> element: | This element is nested within the head element and specifies the title of the web page, which appears in the browser's title bar |
| 6 | <body> element | This element contains all visible content on the page, such as text, images, links, and other HTML elements. |

HTML Hierarchy

HTML elements are hierarchical, which means that they can be nested inside each other to create a tree-like structure of the content on the web page. This hierarchical structure is called the DOM (Document Object Model), and it is used by the web browser to render the web page.

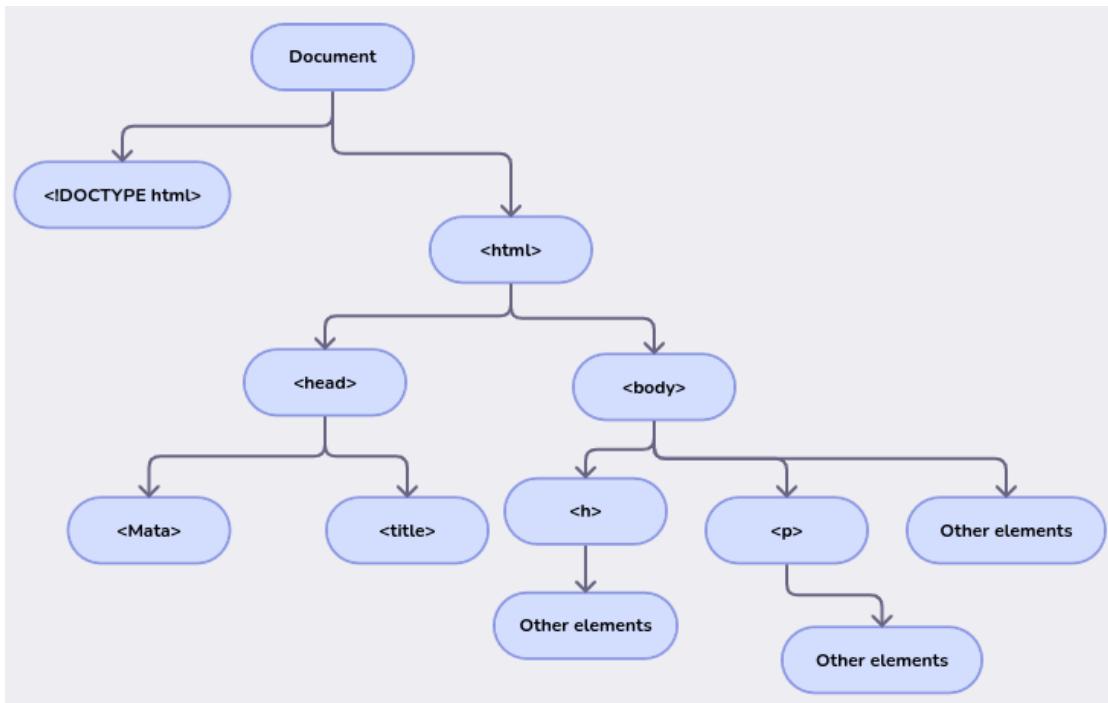


Figure 2.1: Document Object Model

Every web document has two sections. Heading Section and Body Section. The heading section is used to show the title of a webpage in the title bar or tab heading in the browser. The head section should begin with `<head>` tag and end with `</head>` tag. The tag `<title>` is used to specify the title of the webpage.

The body section is used to display the main content on the browser window. The body section should be defined within `<body>` and `</body>` tags. Whatever the text you specify between these tags will display on the browser window.

Example:- Comprehensive of HTML documents

| |
|--|
| Code Description: <i>HTML code to create the first file</i> |
|--|

```

<html>
<head>
    <title> My First Web Page </title>
</head>
<body>
    This is my First Web Page
</body>
  
```

```
</html>
```

Output Description: Basic HTML output will describe each elements

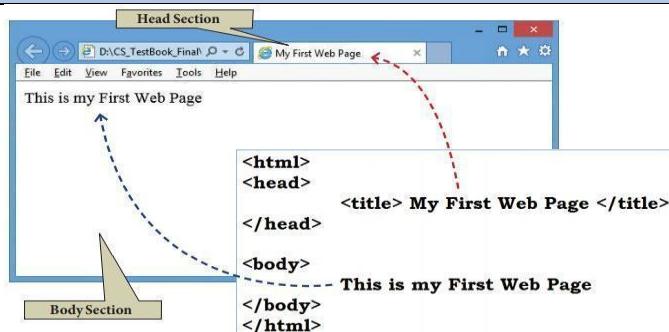


Figure 10.1 – Internet Explorer with my first web page

2.2.1. Create a HTML file

An HTML file is simply a text file saved with an .html or .htm extension. A file that contains Hyper Text Markup Language is called an HTML file. This file is stored in a standard text format with the .html extension and contains tags that define a web page layout and content like text, tables, images, videos, and hyperlinks.

.html is the extension of an HTML file. Like any other file that is on your device, HTML files also have an extension (we'll know more about it later in the article), and that is **.html** at the end of the HTML file name, for example: **index.html**

In the late 90s, when HTML was introduced, **.htm** was used as the extension of an HTML file. However, nowadays, .html is preferred and used commonly. We will discuss this in detail later in this article.

Dear learners! Well, let's get started with creating your first HTML page. Let come that named myfirstpage.html. Are you familiar to create page or not? How to create , save and open your first HTML document?

2.2.2. Saving and opening html documents

In this lesson you will learn how easy it is to create an HTML document or a web page. To begin coding HTML you need only two things: a simple-text editor(NOTEPAD) and a web

browser. Let's walk through the following steps. At the end of this practical lesson, you will have made an HTML file that displays a "Hello world" message in your web browser.



Operation Sheet : 2-3

Title: Save and open html documents

Objective: This operation sheet provides step-by-step instructions on how to save and open properly.

Instructions: Demonstrate how to save and open using HTML document.

Task 1: Save HTML file

The following are the steps to create and save an HTML file with the help of the Notepad text editor.

Step 1: Open Notepad

- Windows7 : Start → All Programs → Accessories → Notepad or
- Windows10 : On the start menu, search for Notepad in Windows, Click on the Notepad editor to open.

Step 2. Type The below HTML Code

```
<!DOCTYPE html>
<html>
  <head>
    <title> My First Web Page </title>
  </head>
  <body>
    This is my First Web Page
  </body>
</html>
```

Step 3. Save the HTML page

Now save the file on your desktop as "myfirstpage.html".

- you can click on File menu and then Save or directly press Ctrl + S.

- Click File → Save (or) Press Ctrl + S
- Save as dialog box appears as shown in the Figure 10.3
- In “File Name” text box, type a file name with .htm or .html extension.
- Select “All Files” from “Save as type” list box. Write the name of the file followed by the .html extension and save the file encoding as UTF-8
- Click “Save” button.

Task 2: View your HTML file /Webpage in a Browser

The following are the steps to open HTML on browser

1. Open a Browser (Internet Explorer / Mozilla Firefox or any)
2. Click File → Open File (or) Press Ctrl + O
3. From the “Open” dialog box, browse the folder in which the HTML document is saved. Choose the File name and click “Open” button.

Task3: View Source file

Source file is an HTML document, what you actually type in text editor (Notepad or getit).

You can view your original source file in the browser.

The following steps are to be followed to view a source file.

1. Right click on the browser
2. Select View Page Source (Firefox and Chrome) / View Source (Internet Explorer) or Press Ctrl + U (all browser)
3. Source file will be displayed.
4. In Internet Explorer, View → Source is also used to open-source file.

Task 4: Edit the source HTML file

Source file can be edit only through the text editor. So, to edit the source file, open the source file with a text editor. When you edit a source file, no need to close browser in which displaying the HTML document.

The following steps are to be followed to open a source file.

1. Go to the folder in which your source files are located.
2. Right click on the source file that you want to edit
3. From the pop-up menu, select Open With → Notepad
4. Source file will be opened in text editor

5. Make the changes and save the file using File → Save or Ctrl + S.

Task 5: Reload / Refresh to see the effect that changed HTML code

The following are the steps to refresh HTML on browser

1. After modify and save the source file, minimize your source file.
2. Go to the browser.
3. Click Refresh (Internet Explorer) / Reload Current Page (Firefox) / Reload this page (Chrome) icons on the address bar.
4. Press Ctrl + R or F5 will be used to refresh / reload the modifications.



If not, please try to write some code on the above HTML file (myfirstpage.html) and see the effect on browser window by make refreshing. It is important that the extension .html is specified — some text editors, such as Notepad, will automatically save it as .txt otherwise.

2.2.3. Metadata and its importance in SEO

Dear learners! Let us commence by presenting significant question. What is metadata and how can it be used in HTML for SEO purposes? The far ahead elucidation offers a comprehensive insight into metadata and its application in HTML for the purpose of SEO

What is HTML metadata?

Metadata is data (information) about data. <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings. Metadata will not be displayed on the page, but is machine parsable.

Metadata is data that provides information about other data. In the context of an HTML page, metadata is information about the content of the page that is not visible to the user, but that can be used by web browsers, search engines, and other systems to understand and interpret the content of the page.

Metadata in an HTML page is typically contained in the head section of the page and is made up of tags and attributes that provide information about the page. Some examples of metadata that can be included in an HTML page are:

- The page's title: This appears in the title bar of the browser and is used to identify the page and to generate search results.
- Keywords: These are words or phrases that describe the content of the page and can be used by search engines to index the page.
- Description: This is a short summary of the page's content that can be used by search engines to provide context for the page in search results.
- Author: This is the name of the person or organization that created the page.
- Copyright information: This is used to protect the intellectual property rights of the page's creator.

Metadata is used in an HTML page to provide additional context and information about the page's content. This can be helpful for search engines, which use metadata to understand the content of a page and to rank it in search results, and for users, who can use metadata to get a better sense of what a page is about before deciding whether or not to visit it. Metadata can also be used by web browsers and other systems to understand and interpret the content of a page and to display it correctly.

What are HTML tags in SEO?

HTML tags are short pieces of code on the web page that provide technical information about its content.

The purpose of these tags is to tell search engines how to “interact” with the page, how it should be displayed in Google Search as well as to visually adjust the page’s content in web browsers (for better user experience).

Why are HTML tags important in SEO?

HTML and meta tags (+ their attributes) indicate to Google what the page is about, how the

web crawler should interact with its content, and whether or not (or how) it should be displayed in Google Search.

Whenever a crawler like Googlebot visits a page, it scans its content along with the provided tags to get a better understanding of the page and decides:

- Whether or not (or how) the page should be crawled and indexed.
- How relevant (and optimized) the page is for certain search queries.
- How it should be displayed in Google SERPs



Self-check 2-2:

1. What file extension do HTML files usually have?
 2. What are some elements usually present in the basic structure of an HTML document?
 3. What is the purpose of metadata in an HTML document?
 4. What type of information do HTML tags provide about web page content?
 5. What is the purpose of HTML metadata?
 6. What are the different types of HTML metadata?
 7. How can HTML metadata be used to improve SEO?
 8. What are some common mistakes that people make when writing HTML metadata?
 9. How can I add HTML metadata to my website?
-

Unit Summary

An integrated development environment (IDE) is a crucial tool for web development, offering a comprehensive platform for creating HTML code. It includes text editors like Notepad, Visual Studio Code (optional), and web browsers. Using an IDE reduces the risk of breaking a live website and makes the development process more efficient and productive. To install VS Code, follow the requirements of a 1.6 GHz or faster processor, 1 GB of RAM, and Windows 10 and 11 (64-bit). An HTML document consists of a declaration, root element, metadata, and element specifying the version of HTML, enclosing all other elements, and providing information about the webpage to search engines and visitors. To create an HTML document, use a simple-text editor (NOTEPAD) and a web browser. Save

the HTML file as an.html or.htm extension and open it in a web browser. Metadata is essential for SEO purposes, providing information about the content of an HTML page, such as title, keywords, description, author, and viewport settings, typically found in the head section.

Unit Review Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. Which of the following is an important first step in web development?
 - A. Publishing the website
 - B. Setting up a development environment
 - C. Creating the website's content
 - D. Deploying the code to a production server
 - E. Designing the website's logo
2. What does an Integrated Development Environment (IDE) typically offer?
 - A. Robust code editors with syntax highlighting, autocompletion, and error checking
 - B. A platform for video editing
 - C. A text-editor with limited functionality
 - D. An application for making spreadsheets
 - E. A tool for browsing the Internet
3. An HTML editor differs from a normal text editor because it:
 - A. Does not allow code to be written from scratch
 - B. Offers features like hassle-free coding
 - C. Is required to test HTML code
 - D. Can only edit HTML code
 - E. Is a mandatory tool for web development
4. Which of the following is NOT a reason for setting up a development environment for HTML?
 - A. To provide a platform for creating and testing HTML code
 - B. To allow developers to work in a controlled environment
 - C. To make changes to live websites directly and frequently
 - D. To experiment without the risk of breaking a live website

- E. To minimize errors and bugs by testing code locally before deployment
5. A good code editor is essential because it:
- Reduces manual tasks and enhances developer productivity
 - Allows users to browse the internet more safely
 - Makes it easier to design website graphics
 - Offers advanced video editing features
 - Provides entertainment with built-in games
6. Which of the following code editors is known for having a WYSIWYG feature?
- Adobe Dreamweaver
 - VS Code
 - Notepad++
 - Atom
 - Sublime Text
7. Which statement about WYSIWYG editors is incorrect?
- They are text-based editors with no preview feature
 - They do not require extensive knowledge of HTML for basic usage
 - "What you see is what you get" is the full form of WYSIWYG
 - They enable non-technical users to develop websites more easily
 - They allow users to drag and drop elements into a web page
8. What is the primary function of a web browser in web development?
- To provide a platform for video conferencing
 - To write and run server-side code
 - To play online games
 - To view web pages and test HTML code
 - To create graphics and logos for websites
9. What does the `<!DOCTYPE html>` declaration signify in an HTML document?
- It specifies the encoding of the document's characters.
 - It links a CSS stylesheet to the HTML document.
 - It is the opening tag of the HTML document.
 - It specifies the version of HTML that the document is written in.
 - It is a comment for developers to read.
-

10. Which HTML element serves as the root of the document?
 - A. <meta>
 - B. <head>
 - C. <html>
 - D. <header>
 - E. <body>
11. Where in an HTML document should metadata typically be included?
 - A. Inside the <title> element
 - B. Inside the <html> element
 - C. In the <footer> section
 - D. At the end of the <body> section
 - E. Inside the <head> element
12. What is the main purpose of the `<title>` element in an HTML document?
 - A. To specify the author of the content
 - B. To create a visible heading
 - C. To specify the title of the web page, which appears in the browser's title bar
 - D. To display body content prominently
 - E. To style the webpage
13. What is the correct sequence of elements for the basic structure of an HTML document?
 - A. <head>, <meta>, <title>, <html>, <body>
 - B. <body>, <head>, <title>, <meta>, <html>
 - C. <html>, <head>, <title>, <meta>, <body>
 - D. <title>, <meta>, <head>, <body>, <html>
 - E. <meta>, <title>, <head>, <html>, <body>

Answer key for Self-check questions

Self-check 2-1: Answer Key

1. You should use an HTML editor in the following situations:
 - Learning HTML: Helps beginners with syntax highlighting and error checking.
 - Creating and Updating Web Pages: Facilitates efficient coding and maintenance.

- Ensuring Code Quality: Provides validation and linting for standards compliance.
- Improving Productivity: Features like autocomplete and snippets speed up coding.

2. Difference Between Text HTML Editors and WYSIWYG HTML Editors

A. Text HTML Editors

Advantages:

- Greater control over the code.
- Ideal for learning and understanding HTML.
- Typically supports multiple languages beyond HTML.

Disadvantages:

- Steeper learning curve for beginners.
- Requires knowledge of HTML syntax.

B. WYSIWYG HTML Editors

Advantages:

- Easier for beginners and non-coders.
- Faster to create basic web pages.
- Provides a visual representation of the final product.

Disadvantages:

- Less control over the generated code.
- May produce bloated or less optimized code.
- Can obscure the understanding of HTML.

3. Common Features of an HTML Editor: Regardless of whether it's a text HTML editor or a WYSIWYG HTML editor, common features include:

- Syntax Highlighting: Differentiates various elements of the code with colors, making it easier to read and debug.
- Autocomplete: Suggests and completes code as you type, increasing efficiency.
- Code Folding: Allows you to collapse and expand sections of code, making navigation easier.
- Error Checking and Validation: Identifies syntax errors and validates code against web standards.

- Search and Replace: Enables quick search and replacement of text or code across the document or entire project.
- Multiple Language Support: Supports not just HTML, but also CSS, JavaScript, and other web technologies.

Self-check 2-2: Answer Key

1. HTML files usually have the file extension ".html" or ".htm".
2. Some elements usually present in the basic structure of an HTML document include the doctype declaration, HTML opening and closing tags, head and body sections, and title tags.
3. The purpose of metadata in an HTML document is to provide additional information about the document, such as the document's title, author, description, and keywords. Metadata helps search engines and other applications understand and categorize the content of the document.
4. HTML tags provide information about the structure and formatting of web page content. They define elements such as headings, paragraphs, links, images, lists, tables, and more.
5. The purpose of HTML metadata is to provide information about a web page that is not displayed directly to the user. This information is used by search engines, social media platforms, and other applications to understand and categorize the content of the page.
6. The different types of HTML metadata include the title tag, which specifies the title of the web page, meta tags, which provide additional information such as description and keywords, and link tags, which define relationships between the current document and external resources.
7. HTML metadata can be used to improve SEO by including relevant keywords in the meta tags, writing compelling and accurate descriptions, and ensuring that the title tag accurately reflects the content of the page. This helps search engines understand the content and relevance of the web page, potentially improving its visibility in search results.

8. Some common mistakes when writing HTML metadata include using duplicate or irrelevant keywords, writing vague or misleading descriptions, and neglecting to update the metadata when the content of the page changes. It's important to ensure that the metadata accurately reflects the content of the web page.
9. To add HTML metadata to your website, you can include relevant tags within the head section of your HTML document.

UNIT 3

CREATE WEB CONTENT USING HTML

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 3.1 HTML Elements
- 3.2 HTML Text formatting elements
- 3.3 HTML links
- 3.4 HTML lists
- 3.5 HTML table
- 3.6 HTML frame
- 3.7 HTML form
- 3.8 HTML multimedia

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- Create web pages
- Insert contents using HTML

Key Terms: Tags, Element, attribute, paragraph, heading, anchor, link, hyperlink, list, table, form and multimedia

Unit Overview

The emphasis of this unit was on HTML elements, which include forms, frames, lists, tables, links, text formatting elements, and multimedia elements. Links enable users to move between sites, and text formatting components style and format content. Lists classify and present data; tables show tabular information; frames divide pages into sections; and forms gather user input via text fields, checkboxes, radio buttons, and submit buttons. The user experience is improved with multimedia features that incorporate photos, movies, and audio files. In general, HTML elements are essential for building dynamic, aesthetically pleasing webpages.

3.1 HTML Elements

HTML uses predefined tags and elements which tell the browser how to properly display the content. HTML Tags and Elements are sometimes perceived as the same. However, they are not. There is a subtle difference between HTML elements and tags that many people aren't aware of.

Dear learners! Before delving into the HTML element, it is essential to familiarize ourselves with the tag and attribute, as this will provide us with a thorough comprehension. What are tags and attributes of HTML element? The key elements of an HTML element are its tag, content, and attribute, which work together to define its

An HTML element is defined by a start tag, some content, and an end tag. The Element can be articulated in the following form or syntax.

Syntax: <element> content </element>

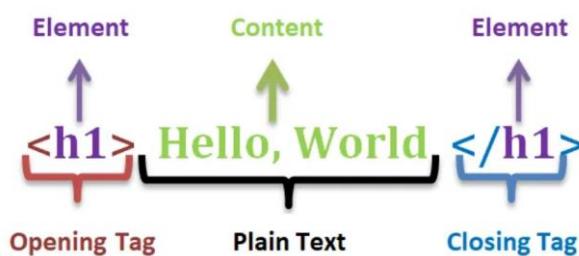


Figure 3.1: HTML element

First, we acquire knowledge about the tag and attribute adjacent to the next HTML element in order to gain a comprehensive understanding.

3.1.1 HTML Tags

HTML Tags are special words wrapped in angle brackets (<>) which are used to define different types of content on a webpage. Every tag in HTML performs different tasks. If you have used an open tag <tag>, then you must use a close tag </tag> .

There are two types of most common tags:

- Paired Tags: These tags come in pairs. That is they have both opening(< >) and closing(</ >) tags

Example 1:

```
<p> content </p>  
<h2> content </h2>
```

- Empty/ unpaired / Self-Closing:These tags do not require to be closed

Example 2:

```
<br/>  
<hr/>  
<img/>
```

Dear learners! Let's Sort the HTML5 general tags into different categories for more detail to understand and utilize easily.what are the different b/n Paired, unpaired and Utility-Based Tags?

For a more detailed analysis, it is recommended to sort HTML tags into distinct categories based on Utility, function and purpose. Now, tags can be categorized into three distinct types: Paired tags, unpaired tags, and Utility-Based tags.

Utility-Based Tags

The HTML tags can be widely differentiated on the basis of their utility, that is, on the basis of the purpose they serve. We can divide them basically into three categories as discussed below:

1. Formatting Tags

The HTML tags that help us in the formatting of the texts like the size of the text, font styles, making a text bold, etc. This is done using tags like ``, ``, `<u>`, etc. Tables, divisions, and span tags are also those tags that help format a web page or document and set the layout of the page.

2. Structure Tags

The HTML tags that help in structuring the HTML document are called Structure Tags. Description, head, html, title, body, etc., form the group of the page structure tags. The structure tags only assist in creating or forming the basic html page from the root; that is, they do not affect or has any hand in the formatting of texts.

3. Control Tags

Another category of tags that can be created is ‘Control Tags’. The Script tags, radio buttons or checkboxes, the Form tags, etc., forms the control tags. These are the tags that are used in managing content or managing scripts or libraries that are external. All the form tags, drop-down lists, input text boxes, etc., are used in interacting with the visitor or the user.

The above distinction of the HTML tags is based on the type of tags and their utility. The HTML tags can also be simply divided based on basic categories like :

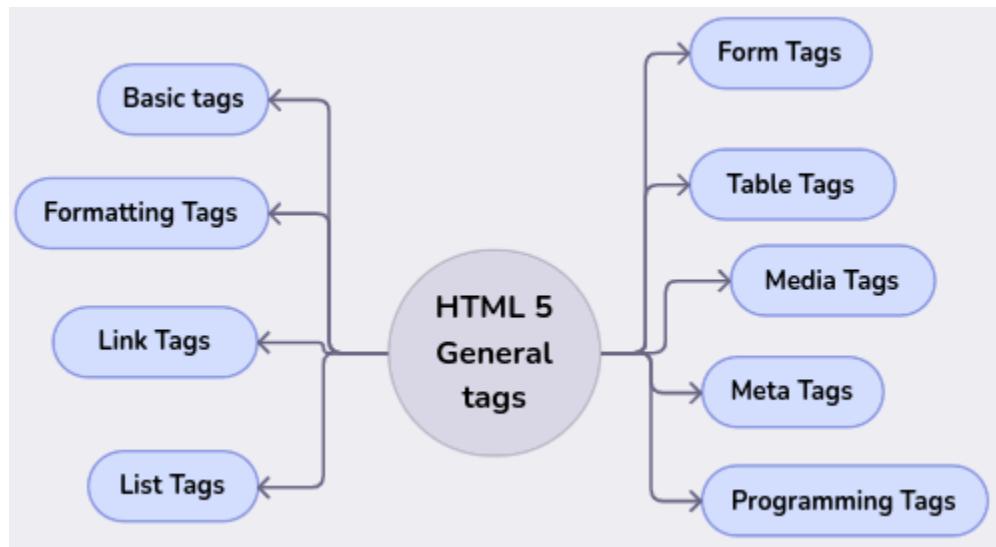


Figure 3.2: Categories of basic HTML tags categories

Table 3-2: HTML tags and Groups

| HTML Groups | Tags |
|---------------------------|--|
| HTML Meta Tags | DOCTYPE, title, link, meta and style |
| Basic Root Tags | html, head, and body |
| HTML Text formatting Tags | <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, |

| HTML Groups | Tags |
|---------------------------------|---|
| | <ins>, , <dfn>, <kbd>, <pre>, <samp>, <var> and |
| HTML Link Tags | <a> and <base> |
| HTML Image and Object Tags | , <area>, <map>, <param> and <object> |
| HTML List Tags | , , , <dl>, <dt> and <dd> |
| HTML Table Tags | table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption |
| HTML Form Tags | form, input, textarea, select, option, optgroup, button, label, fieldset and legend |
| HTML media Tags | Audio, Video, SVG (Scalable Vector Graphics), and Canvas Tags |
| HTML script or programming tags | Canvas, Script and noscript |

Characteristics of HTML Elements:-

- The HTML element is everything from the start tag to the end tag:
- HTML Elements are components that are used in HTML Page.
- HTML Elements contain a starting tag, content, and an ending tag. If there is no content in an HTML Element, it is called an Empty HTML Element.
- HTML Elements can be nested. There can be an HTML element with another HTML Element as its content.

3.1.2 Types of HTML element

There are two types of elements. Inline and block level elements.

1. Inline elements

As the name is describing, it does not start on a new line and takes the space as small as possible. It means it takes the least space.

2. Block elements

Block elements are those elements that occupy their own space and always start on a new line. They make big structures.



All elements don't require the end tag or closing tag to be present. These are referred as empty elements, self-

HTML Comments are used to insert comments in the HTML code. It is a good practice of coding so that the coder and the reader can get help to understand the code. It is a simple piece of code that is wiped off (ignored) by web browsers i.e., not displayed by the browser.

Types of Comment

There are three types of comments in HTML which are:

- Single-line Comment
- Multi-line Comment

The single-line comment is given inside the (<!-- comment -->) tag.

Syntax:

```
<!-- elements or comments -->
```

Example 3: : Single-line Comment

```
<!-- This is a comment -->  
<p>This is a paragraph.</p>  
<!-- add more content here later! -->
```

Multi-line Comment

To create a multi-line comment , you can enclose your comments between <!-- and -->

Syntax:

```
<!--  
    HTML elements  
    comments  
-->
```

Example 4: Multi-line Comment

```
<!--
```

```
<p>This part of my website isn't working right now...I'll fix it later.</p>
<p>This is a paragraph.</p>
-->
```

3.1.3 HTML attributes

In HTML, attributes are used to modify the behavior or appearance of an element or to provide additional information about an HTML element that cannot be conveyed through the element's content alone. Attributes are specified within the opening tag of an element and take the form of name/value pairs, separated by an equals sign.

In HTML, global attributes are attributes that can be used with all HTML elements. These attributes provide common functionality and can be applied universally. They can be used on all elements, though they may have no effect on some elements. Global attributes specified on both standard and non-standard elements. Let's explore some common of these global attributes:

- **class:** Refers to one or more class names for an element, allowing CSS and JavaScript to select and access specific elements via class selectors or DOM methods like `document.getElementsByClassName`
- **data-*:** Used to store custom data private to the page or application.
- **dir:** Specifies the text direction for the content within an element.
- **hidden:** Marks an element as not yet relevant or no longer relevant.
- **id:** Assigns a unique identifier to an element.
- **lang:** Specifies the language of the element's content.
- **spellcheck:** Controls whether the element's spelling and grammar should be checked.
- **style:** Allows inline CSS styling for an element.
- **tabindex:** Defines the tabbing order of an element.
- **title:** Provides extra information about an element.
- **translate:** Determines whether the content of an element should be translated or not.

Let's take a look at the characteristics of HTML attributes:

- Each element or tag can have attributes, which defines the behaviour of that element.
- Attributes should always be applied with start tag.
- The Attribute should always be applied with its name and value pair.

- The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
- You can add multiple attributes in one HTML element, but need to give space between two attributes.
- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"

Syntax:

```
<Tag_name attribute_name="value"> contents </ Tag_name>
```

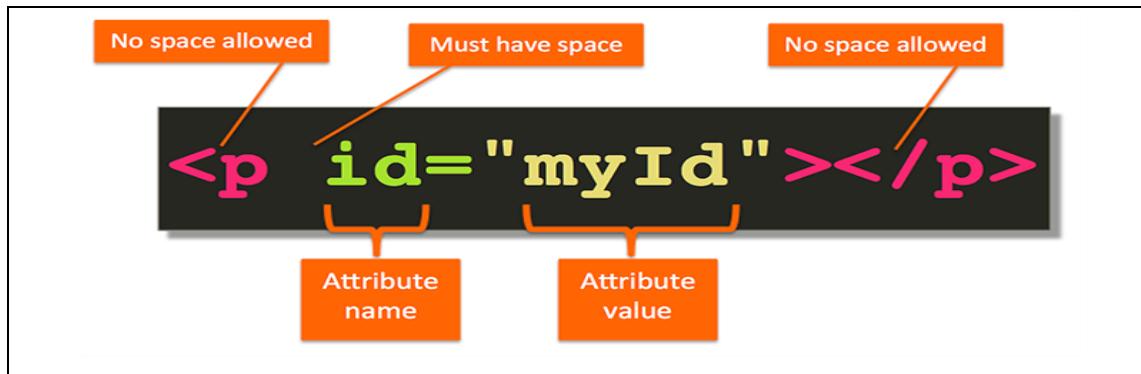


Figure 3.3: Attribute of HTML element

3.1.4 Attributes for HTML document structure Tags

Let's explore the attributes related to HTML document structure tags. These attributes play a crucial role in shaping the structure and behavior of your web page. Here are some fundamental HTML structural tags for documents, including HTML, Head, and Body tags, along with their attributes

I. Attributes of `<html>` tag

The `<html>` tag has two common attributes, such as `dir` and `lang`. The `dir` **attribute** in HTML plays a crucial role in specifying the text direction of an element's content.

Dear learners! How do HTML tag attributes and values work together to define the behavior and appearance of elements on a web page? Here you will see the details of the *HTML* tag attribute and its value.

For Example:Attributes of HTML tag

```
<html dir="rtl" lang="fr">
```

Every attributes has name and value. We define like name="value".dir is name, rtl is value

Table 3-3: Attributes of html tag

| Name of Attribute | Values | Description |
|-------------------|---|---|
| dir | " ltr" (align left-to-right) | dir="ltr": This is the default value. It indicates left-to-right text direction, suitable for languages written from left to right (like English) . |
| | " rtl" (align right-to-left) | dir="rtl": This value signifies right-to-left text direction, commonly used for languages written from right to left (such as Arabic) |
| | " Atuo" | dir="auto": When set to "auto," the browser dynamically determines the text direction based on the content. This option is recommended when the text direction is unknown |
| lang | " en" for English, "es" for Spanish, "fr" for French, and so on | The lang attribute indicates the language of the element's content. The value of the lang attribute is an ISO language code. Typically, this is a two-letter code (e.g., "en" for English), but it can also be an extended code (e.g., "en-gb" for British English) |

II. Attributes of <body> tag

The <body> tag defines the document's body. The contents of an HTML page reside within the <body> tag. <body> tag contains several attributes. There are many attributes in the <body> tag that are deprecated from HTML5 are listed below:

Table 3-4: Attributes of body tag

| Attribute name | Value | Description |
|----------------|-------|---|
| background | image | It contains the URL of the background image. It is used to set the background image. |
| bgcolor | color | It is used to specify the background color of an image. All modern browsers support the following 140 color names |
| alink | color | It is used to specify the color of the active link. |
| link | color | It is used to specify the color of visited links. |
| text | color | It specifies the color of the text in a document. |
| vlink | color | It specifies the color of visited links. |

The <body> tag defines the document's body. The contents of an HTML page reside within the <body> tag. <body> tag contains several attributes.

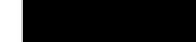
A. Background Colour: bgcolor = color

By default all the browsers display the text on white background. However, the background color of the browser can be changed by using bgcolor tag.

Attribute Values of bgcolor: HTML colors are specified with predefined color names, hex and RGB (*Red, Green, Blue*) values.

- Color name: It sets the text color by using the color name. It specify the name of the color for the text in the Document
For example “red”.
- hex_number: It sets the text color by using the color hex code. It specify the hex code of the color of the Text in the Document
For example “#0000ff”.
- rgb_number: It sets the text color by using the rgb code. It specify the rgb value of the Text in the Document
For example: “RGB(0, 153, 0)” .

Table 3-5: Some Basic colors

| No | Color | HTML / CSS Name | Hex Code #RRGGBB | Decimal Code (R,G,B) |
|----|---|-----------------|------------------|----------------------|
| 1 |  | Black | #000000 | (0,0,0) |
| 2 |  | White | #FFFFFF | (255,255,255) |
| 3 |  | Red | #FF0000 | (255,0,0) |
| 4 |  | Lime | #00FF00 | (0,255,0) |
| 5 |  | Blue | #0000FF | (0,0,255) |
| 6 |  | Yellow | #FFFF00 | (255,255,0) |
| 7 |  | Cyan / Aqua | #00FFFF | (0,255,255) |

Example 5: To change body background color

```
<body bgcolor = yellow> </body>
<body bgcolor = #FFFF00> </body>
<body bgcolor ="RGB(100, 153, 0)"> </body>
```

B. Body text Color: text = color

The default text color of the body section is “black”, it is often called as automatic color.

Example 6: To change body text color

```
<body text="green"> </body>
```

C. Background image: background=image

An image or picture can be applied as background to a webpage. When you insert an image as background, the text will be displayed on top of the image. Background images can be a texture or bitmap or even a photo. When you insert a small image, the browser takes the image and repeats it across and down to fill the browser window. Inserting animated images (GIF images) creates more interest.

Example 7: To Insert images as background

```
<body background = “flower01.gif”></body>
<body background = “E:\Images\ flower01.gif”>
```



Remember that, if the image file and HTML source are located in different locations i.e. in different folder or drive, file path should be clearly specified along with image file name. In the above case, image file and HTML source file both are located within the same folder. So, path name is not mentioned.

D. Setting Margins: margin = value

The margin refers the blank area from left or top edge of the browser window. Generally there is no default margin setting in any browser. If you want to leave some space as margin to left or top; leftmargin or topmargin attributes will be used respectively.

The tag to specify the left and top margin: <body leftmargin = value topmargin = value>

Example 8: To Insert margin in body

```
<body leftmargin = 50 topmargin = 50> </body>
```

Dear Students! Remember that, when you use more than one attribute within an HTML element (Tag) space is used as separator.

Self-check 3-1:

Instruction: Create a web page (HTML document) to the following specification.

- Title: My First Web Page
- Text to be display: Welcome to Computer Applications
- Background color: Lime
- Body text color: Blue
- Margin: from left and top 72 inch

HTML writing best practices

Dear learners! In suggestion, following these best practices for writing HTML can help ensure your code is organized, maintainable, and accessible. By using proper indentation and formatting, semantic HTML, proper naming conventions, external stylesheets, comments, validation, and staying updated with the latest standards,

When it comes to writing HTML, there are several best practices that can help ensure your code is clean, organized, and easy to maintain.

Most Common HTML best practices:

- Use proper indentation and formatting. This means using tabs or spaces to indent nested elements, as well as using line breaks to separate different sections of code. This makes it easier to read and understand the structure of your code.
- Use semantic HTML. It's also important to use proper naming conventions for your classes and IDs. This means using descriptive, meaningful names that accurately describe the purpose of the element, rather than generic names like 'container' or 'box'. This not only makes your code easier to understand, but it also helps with maintenance and debugging.
- Use Proper Document Structure With Doctype
- Use a Meaningful Title and Descriptive Meta Tags
- Use Heading Elements Wisely
- Close the Tags
- Keep your HTML and CSS separate. This means avoiding inline styles and instead using external stylesheets. This helps with organization and makes it easier to make changes to the design of your website without having to sift through HTML code.
- Use proper commenting in your HTML code. Comments are lines of code that are not displayed on the webpage, but are used to explain the purpose or function of certain elements. This is especially helpful when working on a project with a team or when revisiting your code after a period of time.

- Lastly, it's important to stay updated with the latest HTML standards and technologies. HTML is constantly evolving, and staying up to date with the latest best practices and techniques can help improve the overall quality and performance of your website.

3.2 HTML Formatting tags

HTML Formatting basically refers to the enhancement of text in order to increase the visual appeal. HTML provides a range of formatting tags that can be used to make the text attractive to the users. There are many options available that can be used for formatting, just like any other text editor.

3.2.1 Paragraph format

The HTML `<p>` element defines a paragraph. Alternatively, it is used to create paragraphs. A paragraph always starts on a new line.

Syntax:

```
<p> Content </p>
```

Example 9: To add paragraph in HTML

Code Description: *HTML code to create paragraphs*

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph </title>
</head>
<body>
<p>Paragraph 1. </p>
<p>Paragraph 2. </p>
</body>
</html>
```

Output Description: *How the HTML code above looks in a browser*

Paragraph 1.

Paragraph 2.

As we can see, a paragraph starts with the `<p>` and ends with the `</p>` tag.



By default, browsers automatically add whitespace (margin) above and below paragraphs. It is possible to change whitespace and other design aspects using *CSS*.

Break and Line tags

The HTML `
` or `
` tag element creates a line break, giving you a new line without starting a new paragraph. The `<hr/>` or `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

Example 10: To Add Line Breaks and horizontal rule in Paragraphs

Code Description: *HTML code to create paragraphs with breaks and horizontal*

```
<!DOCTYPE html>
<html>
<p>This is <br> a paragraph <br> with line breaks.</p>
<p>This is some text.</p>
<hr/>
<p>This is some other text.</p>
<hr>
</html>
```

Output Description: *How the HTML code above looks in a browser*

This is
a paragraph
with line breaks.

This is some text.

This is some other text.

Example 11: To add Extra Space Inside Paragraphs**Code Description:** *HTML code to create paragraphs with breaks and horizontal*

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph </title>
</head>
<body>
<p> Extra space &nbsp;&nbsp; inside a paragraph </p>
</body>
</html>
```

Output Description: *How the HTML code above looks in a browser*

Extra space inside a paragraph

3.2.2 Heading format Tags

HTML also has six levels of headings, which use the elements `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. It is displayed on the browser in the bold format and size of the text depends on the number of headings. `h1` is the largest heading tag and `h6` is the smallest one. So `h1` is used for most important heading and `h6` is used for least important.

**TIPS**

Browsers automatically add some white space (a margin) before and after a heading. Use HTML headings for headings only. Don't use headings to make text BIG or bold.

Example 12: To add all heading format

Code Description: *HTML code to define headings*

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph </title>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```

Output Description: *How the HTML code above looks in a browser*

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Table 3-6: Attributes for paragraph, heading and horizontal tags

| Attribute name | Value | Description |
|-----------------------|---|--|
| align | left, center and right. | It indicates the alignment of paragraph on the page left, center and right. Horizontally aligned |
| valign | top, middle, bottom | Vertically aligns tags within an HTML element. |
| id | User Defined | The id attribute of an HTML tag can be used to uniquely identify any element within an HTML page. Names an element for use with Cascading Style Sheets. |
| class | User Defined | The class attribute is used to associate an element with a style sheet, and specifies the class of element. Classifies an element for use with Cascading Style Sheets. |
| Style | color: blue; text-align: center; background-color: DodgerBlue; font-size:40px; | The style attribute in HTML allows you to define inline styles sheet for an element. When you use the style attribute, you can specify various visual properties directly within the HTML tag. These properties include things like color, font size, alignment, and more. |
| title | User Defined | The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element: |



The upcoming module will teach you. Keep in mind that using inline styles can make your HTML code less maintainable, so it's often better to separate your styles into a separate CSS file or a `<style>` block in the `<head>` section of your document. While semicolons are used to separate style attribute values, spaces can be used to separate multiple attributes in HTML tag.

Example 13: To apply the Attribute of Paragraph, horizontal and heading tags

Code Description: Comprehensive HTML code that added Attribute of Paragraph, horizontal and heading

```
<p align = "center">This is center aligned</p>
<h2 align=right> Welcome to Computer Application </h2>
<p style="color:green;font-size:40px; text-align: center "> This is a paragraph.</p>
<h1 style="color: blue">This is a header</h1>
<p title="I'm a tooltip">This is a paragraph.</p>
<p style = "font-family:arial; color:#FF0000;">Some text...</p>
<h3 title = "Hello HTML!">Titled Heading Tag Example</h3>
<hr style="border-color:orange">
<hr size="12" width=30% noshade, color="red">
```

Output Description: How the HTML code above looks in a browser

This is center aligned

Welcome to Computer Application

This is a paragraph.

This is a header

This is a paragraph.

Some text...

Titled Heading Tag Example

3.2.3 Text format tags

In Html, there are many tags that are useful to highlight text with special meaning. These tags are known as formatting tags. There are a total of 11 text formatting tags in Html. Formatting tags are designed to highlight special types of text.

The following specific types of text can be displayed using these formatting elements:

- Defines bold text.
- Defines important text.
- <i>Defines a part of the text in an alternate voice or mood.
- <u>..... Specifies underlined text.
- Defines emphasized text.
- <mark>.....Defines marked/highlighted text.
- <small>Defines smaller text.
- Defines deleted text.
- <ins>Defines inserted text.
- <sub>Defines subscripted text.
- <sup>Defines superscripted text.

Example 14: Illustration of some text Formatting

Code Description: *HTML code that show the effect of text format*

```
<!DOCTYPE html>
<html>
<b>This text is bold</b><br>
<strong>This text is important!</strong><br>
<i>This text is italic</i><br>
<em>This text is emphasized</em><br>
<mark>milk</mark><br>
<del>blue</del><br>
<del>blue</del> <ins>red</ins><br>
Water is H<sub>2</sub>O <br>
y=x<sup>3</sup>- 9
</html>
```

3.2.4 Font Formatting tag

HTML tag is used to define the font style for the text contained within it. It defines the font size, color, and face or the text in an HTML document.

Table 3-7: Attributes of font tags

| Attribute | Value | Description |
|-------------|---------------------------------------|---|
| name | | |
| Size | Number | The size attribute is used to set size of the text. |
| Color | Color name, hexadecimal and RGB | The color attribute is used to set the color to the text. |
| Face | Font names | The face is an attribute to set different font style. |

Dear learners: Do not use HTML tag as it is deprecated in HTML5, so you can use CSS properties to change the font size, face, color, etc.

Example 15: Illustration of some text Formatting

| |
|--|
| Code Description: HTML code that show the effect of font |
| <!DOCTYPE html> <html> Font.... Welcome to HTML Content.... </html> |
| Output Description: How the HTML code above looks in a browser |

Font....

Welcome to HTML
Content....



TIPS

In Font tag Multiple attribute, you can assign more than one

HTML Nested Element

HTML elements can be *nested*, meaning that one element can be placed inside another element. Nesting allows you to apply multiple HTML tags to a single piece of content. The element can be nested by other elements. The inner element is called child or nested element and the outer element is called parent or nesting element.

Example 18: Illustration of Nesting HTML tags

Code Description: HTML code that show who to apply nest tags

```
<!DOCTYPE html>
<html>
  <body>
    <h1><u>My First Heading</u></h1>
    <p><b><u><font size="12">Nesting element...
      </font></u></b></P>
    </body>
  </html>
```

Output Description: How the HTML code above looks in a browser

My First Heading

Nesting element...

Text may be both bold and italicized. This is done by using both the **** and *<I>* tags. When doing so, it is important to remember not to overlap HTML tags. In other words:

<I> Hello!!**</I>** is correct, but

<I> Hello!! **></I>** is wrong.

Note that it is recommended to always close nested tags in the reverse order that they were opened. This means that the last tag that was opened should be closed first, followed by the second-to-last tag, and so on, until the first tag that was opened is closed. This is important because it helps maintain the structure and hierarchy of the HTML elements, ensuring that they are properly nested within each other. By following this practice, you can avoid any potential errors or rendering issues that may occur if the tags are not closed in the correct order.



Self-check 3-2:

1. Write code that can provide the output shown below?

Output

Welcome to My Website

This is a paragraph of text.

Another paragraph can start here.

1st and 2nd year students

2. What output is produced by the HTML document code below?

Code Description: *HTML code that show the effect of font*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Formatting Example</title>
</head>
<body>
    <h1>Main Heading</h1>
    <p>This is a basic paragraph of text.
    <h2>Subheading</h2>
    <p>You can also use <i>italic text</i> with the <code>&lt;i&gt;</code> tag and
    <em>emphasized text</em> with the <code>&lt;em&gt;</code> tag.</p>

    <h3>Monospaced Text</h3>
    <p>Inline <code>monospaced text</code> is useful for code snippets.</p>
    <pre>This is a preformatted text block.</pre>
    <p>You can <b><i>combine</i></b> <u>multiple</u>
    <strong><em>formatting</em></strong> <s>options</s> in one paragraph.</p>
</body>
</html>
```

3.2.5 Image tag

Images are an essential element to make an HTML presentation in a more attractive manner. Moreover images are used to depict many complex concepts in a simple way. To make more attractive and communicative web pages, images should be added in the appropriate places. Images displayed on the web page should be converted to universally supported format. Most of the browsers support GIF(Graphical Interchange Format), JPEG (Joint Photographic

Experts Group) and PNG (Portable Network Graphics) images formats. HTML-5 introduces SVG images. One format of image can be converted to another format by using Image editing applications such as Photoshop, Picasa, GIMP etc.

Inserting Images with HTML document

The `` tag along with the attribute `src` (Source) is used to add images in HTML document.

Syntax:

```
<img src = image_name_with_extension> OR  
<img src = URL>
```

Table 3-8: Attributes of image tags

| Attribute name | Value | Description |
|-----------------------|--|--|
| src | Image file | Src attribute is the main attribute used to specify the file name of the image to be inserted. |
| alt | Text | The alt attribute is used to describe the image, so that some text is conveyed even when the image cannot be displayed. |
| Width and Height | Percentage number | Width and Height attributes are used to set the width and height of an image. The values of these attributes should be either pixels or percentage of its actual size. |
| Vspace and Hspace | Integer number | Vspace (Vertical Space) and Hspace (Horizontal Space) attributes are used to set Vertical and Horizontal space between the images. |
| Align | Bottom , Middle , Top , Left and Right | The align attribute used to aligns the image with respect to the base line of the text. Using these values with align attribute, displayed the image on the bottom ,middle ,top , bottom ,middle ,left and right side of the text. |

Example 19: Illustration of how to insert image**Code Description:** *HTML code that show how to insert image tags*

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Inserting Image</h2>

</body>
</html>
```

Output Description: *How the HTML code above looks in a browser*

HTML Inserting Image



3.2.6 Marquee tag

The **<marquee>** tag in HTML is used to create scrolling text or images on a webpage. It can scroll horizontally from left to right or right to left, and vertically from top to bottom or bottom to top. It includes attributes like **direction** to specify whether the content moves left, right, up, or down.

Syntax:

```
<marquee> contents </marquee>
```

Table 3-9: Attributes of image tags

| Attributes | Values | Description |
|------------|------------------------|--|
| bgcolor | Color Name | Define the background color of the marquee. |
| direction | Top, Down, Left, Right | Define the direction of scrolling the content |
| loop | Number | Specifies how many times content moves. The default value is infinite. |
| height | px or % | Define the height of marquee |
| width | px or % | Define the width of marquee |
| hspace | px | Specify horizontal space around marquee |
| vspace | px | Specify vertical space around marquee |

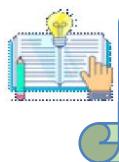
Example 20: Illustration of how to apply the attributes of marquee

```
<marquee behavior="scroll" direction="left" scrollamount="1">Slow scroll speed </marquee>
<marquee behavior="scroll" direction="left" scrollamount="10"> Medium scroll speed
</marquee>
<marquee behavior="scroll" direction="up">Your upward scrolling text goes here
</marquee>
</marquee>
```

3.3 HTML Links

Link is an important feature of HTML to connect web resources. Link in HTML is used to create hyperlinks to web content. Web content may be an HTML document or an external webpage or any multimedia content such as an image, video, audio, animation etc., or even a

part of the current document. When you move the mouse over a link, the mouse arrow will turn into a little hand. Hyperlinks, also known as anchor tags, allow users to navigate from one web page to another. They are found in nearly all web pages.



A picture or any other HTML element can be used as a link in addition to text!

Anchor tags

The **<a> tag (anchor tag)** in HTML is used to create a hyperlink on the webpage. The **<a>** HTML element (or anchor element), with its href (Hypertext Reference) attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address.

HTML offers several types of links, each serving a specific purpose. Let's see some of them in action in the following sections.

- Anatomy /Text links or clickable text
- Image links or clickable image
- email links
- mailto links

Table 3-10: Attributes of the anchor tag

| Attribute name | Values | Description |
|----------------|-------------------|---|
| hreflang | | Specifies the language of the linked resource. |
| download | | Directs the browser to download the linked resource rather than opening it. |
| target | _blank _parent | Specifies the context in which the linked resource will open. |

| | | |
|-------|-----------------------------|--|
| | _self _top frame name | |
| title | text | Defines the title of a link, which appears to the user as a tooltip. |
| href | url or File | Specifies the linked document, resource, or location. |

Text links

Text links are the most common type of links. They are created by wrapping text with an anchor () element. When users click on the linked text, they are directed to the URL specified in the link's href attribute:

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

Text links are versatile and can be used for various purposes, such as linking to other web pages, external websites, or even specific sections within a page using anchor tags.

Image links

You can turn images into clickable links by wrapping them in an anchor element. This is useful for creating an image-based navigation or linking to larger versions of images:

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

Image links are visually engaging and are often used for elements like logos, banners, or thumbnail images that, when clicked, lead users to a related web page or resource.

Email links

To create links that open an email client with a pre-filled recipient address, use the mailto scheme:

```
<a href="mailto:contact@example.com">Send an Email</a>
```

Email links are convenient for enabling users to initiate email communication with a simple click. They are commonly used for contact information on websites.

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link. The target attribute specifies where to open the linked document. The target attribute can have one of the following values:

Table 3-11: Value of target

| Value | Description |
|-----------|---|
| _self | It's Default. Opens the document in the same window/tab as it was clicked |
| _blank | Opens the document in a new window or tab |
| _parent | Opens the document in the parent frame |
| _top | Opens the document in the full body of the window |
| Framename | Opens the linked document in the named iframe |

Syntax:

```
<a target="_blank / _self /_parent/_top/ framename">
```

For example:

```
<a href="https://www.google.com" target="_blank">click here to open google in a new tab </a>
```

| | |
|---|--------------------------------|
|  | <h3>Operation Sheet : 3-1</h3> |
| <p>Title: Internal / relative and external /absolute Link</p> <p>Objective: To demonstrate an Internal / relative and external /absolute Links among pages:</p> <p>Instructions: Let us create two files “page1.html” & “page2.html” in same folder. We will create a link between these two files using relative URL. We will also test a link using absolute url as well.</p> <p>Task 1. Create a HTML file “page1.html” and write following code:</p> <p>Type this code and Save by named “ page.html” file</p> <pre><!DOCTYPE html> <html> <body> Example of Relative Link to page 2 </pre> | |

```
<br/>
<a href="https://www.w3schools.com/html/" target="_blank">Example of Absolute Link to
w3schools</a>
</body>
</html>
```

Task 2. Create a HTML file “page2.html” and write following code:

Type this code and Save by named “ page.html” file

```
<!DOCTYPE html>
<html>
<body>
<p>Make Anatomy / text link</p>
<a href="page1.html">Example of Relative Link to page1</a>
<br/>
<hr/>
<a href="http://google.com" target="_blank" >Example of Absolute Link to google</a>
<br/>
<p>click on the below Image</p>
<a href="https://www.youtube.com/" target="_blank">
</a>
<br/>
<hr/>
</body>
</html>
```



Self-check 3-3:

How would you make a relative link to a page located in a subfolder?

1. How can you use an image as a hyperlink in HTML?
 2. How do you create a mailto link in HTML?
 3. How would you link to a downloadable file, such as a PDF, in HTML?
 4. How can you create a link that scrolls to the top of the current page?
-

3.4. HTML Lists

A list is a record of short pieces of related information or used to display the data or any information on web pages in the ordered or unordered form. For instance, to purchase the items, we need to prepare a list that can either be ordered or unordered list which helps us to organize the data & make it easy to find the item. Please refer to the HTML `` type Attribute article for the various types of attributes that can be used with the ordered & unordered list.

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (`ol`)
2. Unordered List or Bulleted List (`ul`)
3. Description List or Definition List (`dl`)

3.4.1 HTML Ordered List or Numbered List

An ordered list is a list in which the items are numbered and the order matters. HTML Ordered List or Numbered List displays elements in numbered format. The HTML `ol` tag is used for ordered list. We can use ordered list to represent items either in numerical order format or alphabetical order format, or any format where an order is emphasized. There can be different types of numbered list:

1. Numeric Number (1, 2, 3)
2. Capital Roman Number (I II III)
3. Small Romal Number (i ii iii)
4. Capital Alphabet (A B C)
5. Small Alphabet (a b c)

In the ordered HTML lists, all the list items are marked with numbers by default. It is known as a numbered list. The ordered list starts with `` tag and the list items start with `` tag.

Basic Syntax of the `` tag

The `` tag defines ordered lists in HTML. And the list items are defined by the `` tag.

The `` tag is not an empty element, so it has a closing tag in ``

Syntax:

```
<ol>
```

```
<li>...</li>
<li>...</li>
<li>...</li>
</ol>
```

Example 19: Illustration of how to add numbered list

Code Description: *HTML code that show who to insert image tags*

```
<!DOCTYPE html>
<html>
<body>
<OL>
<h3>types of Fruit</h3>
<li>Apple</li>
<li>Banana</li>
<li>Mango</li>
<li>others</li>
</OL>
</body>
</html>
```

Output Description: *How the HTML code above looks in a browser*

Types of Fruit

1. Apple
2. Banana
3. Mango
4. others

Attributes of the Tag

There are two attributes can be used to customize ordered list, they are Type, Start and Reversed.

Table 3-12: Attributes of the order list

Let's see the example of an HTML ordered list that displays a numbered list. Here we are not defining type="1" because it is the default type.

Example 20: Demonstrate Provide an example of the type attribute with four different values such as roman number, and letter

| Code Description: <i>HTML code that shows how to apply the four different values of tag.</i> | | |
|--|--|--|
| <i>image tags</i> | | |
| <ol type="I"> HTML CSS JS | <ol type="i"> HTML CSS JS | <ol type="A"> HTML CSS JS |

| Attribute name | Values | Description |
|----------------|--|---|
| Type | Arabic numerals, upper or lower case roman numbers and upper or lower case letters.Example: Type=1 Type=I or i Type= A or a | The type attribute is used to specify which type of numbering you want to use for the list. It is changing numbering style. |
| Start | Arabic numerals Example: Start=3 | The start attribute is used with ol tag to specify from where to start the list items. It is changing numbering order. |
| Reversed | Arabic numerals, upper or lower case roman numbers and upper or lower case letters.Example: | When you use the reversed attribute on an ordered list, the list items are rendered in reverse order. That is, from the highest number to the lowest. <small>B88</small> |

| Output will be: upper roman list | Output will be: Lower case roman | Output will be: Upper case letters |
|---|---|---|
| I. HTML II. CSS III. JS | i. HTML ii. CSS iii. JS | A. HTML B. CSS C. JS |

The start attribute can be brought in to specify which number to start the list from. So, it accepts an integer as a value. The default is 1. If you specify a number like 22, the next list item takes the next number 23, on and on...

Example 21: Demonstrate type and start attribute

| Code Description: HTML code that show who to use type and start form any number |
|--|
| <Ol type="1" start="6"> HTML CSS JS |
| Output Description: How the HTML code above looks in a browser |
| 6. HTML 7. CSS 8. JS |

The start attribute is used with the tag to specify from where to start the list items.

<ol type="1" start="5"> : It will show numeric values starting with "5".

<ol type="A" start="5"> : It will show capital alphabets starting with "E".

<ol type="a" start="5"> : It will show lower case alphabets starting with "e".

<ol type="I" start="5"> : It will show Roman upper case value starting with "V".

3.4.2 HTML Unordered List | HTML Bulleted List

HTML Unordered List or Bulleted List displays elements in bulleted format .We can use unordered list where we do not need to display items in any particular order. The HTML ul tag is used for the unordered list. Unordered lists are often referred as bulleted lists. Instead of numbers, each element in the list has prefixed with a special bullet symbol. Unordered list is surrounded within tags. As discussed above, each list element is defined by tag. Like ordered list, type attribute is used to customize bullet style for the list of elements. There can be 4 types of bulleted list:

- Disc : A solid circle
- Square : A solid square
- Circle : An unfilled circle

To represent different ordered lists, there are 4 types of attributes in tag. By default, a solid circle is used as bullets.

Table 3-13: Attributes of the Unorder list

| Type | Description |
|---------------|---|
| Type "disc" | This is the default style. In this style, the list items are marked with bullets. |
| Type "circle" | In this style, the list items are marked with circles. |
| Type "square" | In this style, the list items are marked with squares. |
| Type "none" | In this style, the list items are not marked . |

Example 22: An HTML code illustrating the four different attributes of the element.

Code Description: *HTML code that show who to use all possible types of unordered list*

```
<UL>
<li>HTML</li> <li>Java</li> <li>JavaScript</li> <li>SQL</li>
</UL>
```

```
<UL type= "square" >  
  <li>HTML</li> <li>Java</li> <li>JavaScript</li> <li>SQL</li>  
</UL>  
<UL type="circle "'>  
  <li>HTML</li> <li>Java</li> <li>JavaScript</li> <li>SQL</li>  
</UL>
```

Output Description: How the HTML code above looks in a browser

- HTML
 - Java
 - JavaScript
 - SQL
-
- HTML
 - Java
 - JavaScript
 - SQL
-
- HTML
 - Java
 - JavaScript
 - SQL

3.4.3 Definition lists

Definition list is different from other two types of list. No bullet or number is provided for the list items. In this list type, the list element has two parts.

1. A definition term
2. The definition description

HTML Description Lists Tags

- **<dl> tag:** This tag defines the description list.
- **<dt> tag:** This tag defines the data terms inside the list.
- **<dd> tag:** This tag defines the description of data.

Example 22: An HTML code illustrating the Definition list element.

Code Description: *HTML code that show how to use all possible types of unordered list*

```
<DL>
<DT> HTML: </DT>
    <DD> Hyper Text Markup Language </DD>
<DT> Webpage:
    <DD> A web page is a document that is suitable for the World Wide Web and web browsers.
    </DD>
</DL>
```

Output Description: *How the HTML code above looks in a browser*

HTML:
Hyper Text Markup Language
Webpage:
A web page is a document that is suitable for the World Wide Web and web browsers.

3.4.4 HTML Nested List

A list block can be defined inside another list is called nested list.

An example of a nested list in HTML

Code Description: *HTML code that show how to use all possible types of unordered list*

```
<OL>
<LI> Regional state of Ethiopia
    <UL type="square">
        <LI> Afar Region</li>
        <LI> Amhara Region</li>
        <LI> Oromia</li>
        <LI> Others</li>
    </UL>
```

```
<LI> District of Addis Ababa  
<OL type="I">  
    <LI> Addis Ketema</li>  
    <LI> Akaky Kaliti</li>  
    <LI> Arada</li>  
    <LI> Bole</li>  
    <LI> Others</li>  
</OL>  
</OL>
```

Output Description: How the HTML code above looks in a browser**1. Regional state of Ethiopia**

- Afar Region
- Amhara Region
- Oromia
- Others

2. District of Addis Ababa

- I. Addis Ketema
- II. Akaky Kaliti
- III. Arada
- IV. Bole
- V. Others

3.5 HTML table

HTML Table is an arrangement of data in *rows* and *columns* in tabular format. Tables are useful for various tasks such as presenting text information and numerical data. A table is a useful tool for quickly and easily finding connections between different types of data. Tables are also used to create databases.

HTML Table tags

The HTML tables are created using the `<table>` tag. In Each table, table row is defined by `<tr>` tag, table header is defined by `<th>`, and table data is defined by `<td>` tags.

Table 3-14: Common HTML Table Tags

| HTML Tags | Descriptions |
|------------|---|
| <table> | Defines the structure for organizing data in rows and columns within a web page. |
| <tr> | Represents a row within an HTML table, containing individual cells. |
| <th> | Shows a table header cell that typically holds titles or headings. |
| <td> | Represents a standard data cell, holding content or data. |
| <caption> | Provides a title or description for the entire table. |
| <thead> | Defines the header section of a table, often containing column labels. |
| <tbody> | Represents the main content area of a table, separating it from the header or footer. |
| <tfoot> | Specifies the footer section of a table, typically holding summaries or totals. |
| <col> | Defines attributes for table columns that can be applied to multiple columns at once. |
| <colgroup> | Groups together a set of columns in a table to which you can apply formatting or properties collectively. |

Example 23: An HTML code illustrating the Table tags element.

Code Description: *HTML code that show how to create table without border*

```
<table>
  <caption> A summary of the UK's most famous punk bands </caption>
  <thead>
    <tr>
      <th scope="col">Band</th>
      <th scope="col">Year formed</th>
      <th scope="col">No. of Albums</th>
      <th scope="col">Most famous song</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Buzzcocks</th>
      <td>1976</td>
      <td>9</td>
      <td>Ever fallen in love (with someone you shouldn't've)</td>
    </tr>
    <tr>
      <th scope="row">The Clash</th>
      <td>1976</td>
      <td>6</td>
      <td>London Calling</td>
    </tr>
    <!-- several other great bands -->
    <tr>
      <th scope="row">The Stranglers</th>
      <td>1974</td>
```

```

<td>17</td>
<td>No More Heroes</td>
</tr>
</tbody>
<tfoot>
<tr>
<th scope="row" colspan="2">Total albums</th>
<td colspan="2">77</td>
</tr>
</tfoot>
</table>

```

Output Description: How the HTML code above looks in a browser

| A summary of the UK's most famous punk bands | | | |
|--|--------------------|----------------------|---|
| Band | Year formed | No. of Albums | Most famous song |
| Buzzcocks | 1976 | 9 | Ever fallen in love (with someone you shouldn't've) |
| The Clash | 1976 | 6 | London Calling |
| The Stranglers | 1974 | 17 | No More Heroes |
| Total albums | | 77 | |

Attributes of Table Tag

The following table shows the attributes that are specific to the <table> tag.

| Attribute | Value | Description |
|-----------|-------------------------|--|
| align | left center right | This attribute provides content alignment inside an element. It specifies the alignment of the table with respect to the document. |
| bgcolor | color | This attribute specifies the background color for the table. It sets the background color of the table. |

| Attribute | Value | Description |
|-------------|-----------------------------|--|
| border | 1 0 in Numbers | This attribute specifies the border for the table cells. It Specifies whether the table cells should have borders or not. |
| cellpadding | Length in Number/percentage | This attribute displays the padding between table cells and table content .It Specifies the space between the edge of a cell and its content |
| cellspacing | Length in Number/percentage | This attribute indicates the space between table cells.It Specifies the amount of space between individual cells. |
| height | Length in Number/percentage | This attribute specifies the height of the table |
| width | Length in Number/percentage | This attribute tells the width of the table .It Specifies the width of the entire table. |
| Style | CSS Value | A attribute on a <table> tag assigns a unique style to the table. Its value is CSS that defines the appearance of the table. |



TIPS

Do not use tables for creating web page layouts. Table layouts are slower at rendering, and very difficult to maintain. It should be used only to display tabular data.



Operation Sheet: 3-6

Title: Table attribute

Objective: This operation sheet provides step-by-step instructions on how to use These attributes help control the appearance, layout, and behavior of the table and its contents.

Instructions: type each HTML code and see the effects of them in the page.

Task1: Specifies the width of the border around the table cells

```
<table border="1"></table>
```

Task2: Specifies the space between the cell content and its borders html

```
<table cellpadding="10"></table>
```

Task3: Specifies the space between the table cells.

```
<table cellspacing="10"></table>
```

Task4: Specifies the width of the table.

```
<table width="100%"></table>
```

Task5: Specifies the height of the table.

```
<table height="200"></table>
```

Task6: Specifies the alignment of the table on the page. Values can be left, right, or center.

```
<table align="center"></table>
```

Task7: Specifies the background color of the table.

```
<table bgcolor="lightblue"></table>
```

Example 24: An HTML code to demonstrate table Border ,bgcolor and Align Attribute

Code Description: *HTML code that show who to use all possible types of unordered list*

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Tag Usage</title>
</head>
<body>
<table align="center" border="1" bgcolor="yellow">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
```

```
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

Output Description: How the HTML code above looks in a browser

| Header 1 | Header 2 |
|-----------------|-----------------|
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

Attributes for Table Rows and Cells

HTML also provides developers with the option to merge and split table cells. Merging cells allows us to combine the content of two or more cells into a single cell, while splitting cells allows us to divide a single cell into multiple cells. Here's an example of how to use HTML elements to split and merge table cells:

- Colspan is an attribute which assigns multiple columns to a cell of a table. The number of columns depends on the value entered by you in colspan="" attribute. It Allows a single table cell to span the width of more than one cell or column.
- Rowspan in table works similar to the colspan for columns, but here, we assign multiple rows to a cell using an attribute rowspan"". It Allows a single table cell to span the height of more than one cell or row.

Dear Students! You can merge two or more table rows into a single row by using the rowspan attribute, and you can merge table columns into a single column by using the colspan attribute.

Example 25: Merge column and row cells

Code Description: HTML code that shows how to make merged cells based on column

```
<!DOCTYPE html>
<table border="1">
  <caption> New Employees Records </caption>
  <tr>
    <th>Name</th>
    <th colspan="2">Jobs</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Software Engineer</td>
    <td>Data Analyst</td>
    <td rowspan="2">5 Years</td>
  </tr>
  <tr>
    <td>Abebe</td>
    <td colspan="2">Senior Web developer</td>
  </tr>
  <tr>
    <td>Hirut</td>
    <td>Junior Tech Writer</td>
    <td>Blogger</td>
    <td>6 Months</td>
  </tr>
</table>
```

| New Employees Records | | | |
|-----------------------|----------------------|--------------------|----------|
| Name | Jobs | Working Experience | |
| John | Software Engineer | Data Analyst | 5 Years |
| Abebe | Senior Web developer | | |
| Hirut | Junior Tech Writer | Blogger | 6 Months |

Example 26:Demonstration of Height and Width of the Table

| Code Description: HTML code that demonstrates how to make a table with dimensions | |
|---|--|
| <!DOCTYPE html> | |

```

<html>
<head>
<title></title>
</head>
<body>
<table border = "1" width = "500" height = "250" bgcolor = "lightblue">
<tr>
<th>Student Name</th>
<th>Grade</th>
</tr>
<tr>
<td>Abebe</td>
<td>10</td>
</tr>
<tr>
<td>Selam</td>
<td>11</td>

```

```

</tr>
<tr>
<td>Hirut</td>
<td>11</td>
</tr>
</table>
</body>
</html>

```

Output Description: How the HTML code above looks in a browser

New Employees Records

| Name | Jobs | | Working Experience |
|-------|----------------------|--------------|--------------------|
| John | Software Engineer | Data Analyst | 5 Years |
| Abebe | Senior Web developer | | |
| Hirut | Junior Tech Writer | Blogger | 6 Months |



Self-check 3-4:

1. Write code that can provide the output table shown below?

| Item | Quantity | Price | Amount |
|--------------|----------|-------|-----------|
| Computer Box | 2,000 | 1,000 | 2,000,000 |
| Paper | 100 | 800 | 80,000 |
| USB Flash | 600 | 1500 | 900,000 |
| Sub total | | | 2,980,000 |
| Tax | 15% | | 447,000 |
| Total | | | 3,427,000 |

3.6 HTML frame

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized.

3.6.1. <frameset> Element

The <frameset> element replaces the <body> element in a document, and it defines how to divide the window into frames. Each <frameset> can contain multiple <frame> elements or other <frameset> elements to create nested frames.

Using the <frameset> tag with percentages and the asterisk (*) allows you to create flexible layouts where certain sections of the page take up fixed or proportional amounts of space, and the remaining space is allocated to other sections.

The <frameset> element was used in HTML 4 to define a web page that should use frames, where each frame could display a separate HTML document. However, the use of frames and the <frameset> element is deprecated in HTML5, meaning it is no longer recommended for use, and support for it may be removed from browsers. Instead, CSS and modern layout techniques such as Flexbox and Grid Layout are recommended for creating complex page layouts.



HTML frames are obsolete and should not be used in modern web development.

Undoubtedly, there are several drawbacks associated with using frames, which is why they are generally not recommended for modern web development. Here is a detailed explanation of these drawbacks:

Drawbacks of Using HTML Frames

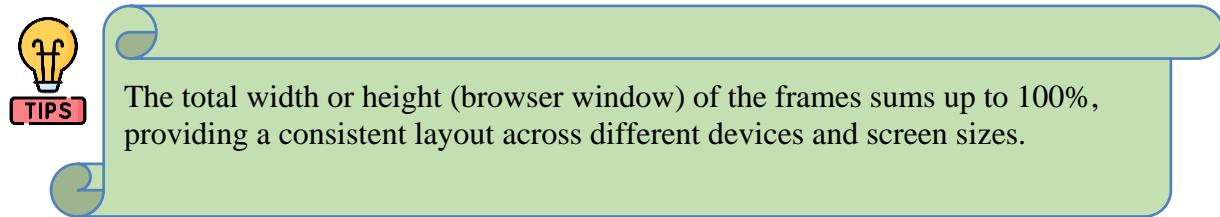
1. Incompatibility with Smaller Devices:
 - *Problem:* Many smaller devices, such as smartphones and tablets, struggle to display frames properly because their screens are not large enough to be divided into multiple sections.
 - *Impact:* Users on these devices may experience poor usability and an impaired browsing experience, leading to frustration and potentially causing them to leave the site.
2. Inconsistent Display Across Different Computers:
 - *Problem:* Web pages using frames can be displayed differently on different computers due to varying screen resolutions and browser configurations.
 - *Impact:* This inconsistency can make it difficult to ensure a uniform user experience across all devices and screen sizes. Elements may not align correctly, and content can become difficult to read or navigate.
3. Browser Back Button Issues:
 - *Problem:* The browser's back button may not work as expected when using frames. Typically, the back button reloads the entire frameset instead of just the content of a single frame.
 - *Impact:* This can confuse users and make navigation more cumbersome, as they might not be able to easily return to previously viewed content within a frame.
4. Lack of Support in Some Browsers:
 - *Problem:* Although most modern browsers support frames, there are still a few browsers that do not support frame technology. Additionally, as HTML5 does not include frameset elements, future browser versions may further limit support.
 - *Impact:* Users of these browsers will be unable to view the content as intended, which can lead to a loss of accessibility and a diminished user experience.

5. Search Engine Optimization (SEO) Challenges:

- *Problem:* Search engines may have difficulty indexing framed content correctly. Since each frame is a separate HTML document, search engines might not be able to associate the content within frames with the main site effectively.
- *Impact:* This can negatively affect the site's search engine rankings and reduce its visibility in search results.

6. Accessibility Issues:

- *Problem:* Frames can create barriers for users with disabilities. Screen readers and other assistive technologies may struggle to navigate framed content effectively.
- *Impact:* This can lead to compliance issues with accessibility standards (such as WCAG) and make the website less usable for people with disabilities.



3.6.2. Attributes of <frameset>

1. **rows:** Defines the number and size of the rows in the frameset. Values can be specified in pixels, as a percentage of the total window size, or using asterisks (*) for proportional resizing.
2. **cols:** Defines the number and size of the columns in the frameset. Similar to rows, values can be in pixels, percentages, or asterisks.

3.6.3. Attributes of <frame>

1. **src:** Specifies the URL of the document to be displayed in the frame.
2. **name:** Assigns a name to the frame. This can be used as the target for links and form submissions.
3. **frameborder:** Specifies whether the frame should have a border (1 for yes, 0 for no).

4. **marginwidth:** Specifies the width of the margin, in pixels, between the frame's content and its left and right edges.
5. **marginheight:** Specifies the height of the margin, in pixels, between the frame's content and its top and bottom edges.
6. **noresize:** Prevents the user from resizing the frame.
7. **scrolling:** Specifies whether or not to display scrollbars. Values can be **yes**, **no**, or **auto**.
8. **Target:** Target attribute in HTML is used to specify where to open the linked document.
 - **Target Attribute Values**
 - **_self:** Opens the linked document in the same frame or window as the link (default behavior).
 - **_blank:** Opens the linked document in a new window or tab.
 - **_parent:** Opens the linked document in the parent frame.
 - **_top:** Opens the linked document in the full body of the window (i.e., breaks out of any framesets).
 - **framename:** Opens the linked document in the specified frame.

Basic HTML Frame Structure

Frames are created using the `<frameset>` and `<frame>` elements. It defines how to divide the window into frames.

Example: Basic Frameset:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<frameset>
  <frame src="">
  <frame src="">
</frameset>
</html>
```

3.6.4. Create row and column Frames

Example: Columns Frameset

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<frameset cols="50%, *">
    <frame src="">
    <frame src="">
</frameset>
</html>
```



The browser window is divided into two columns (cols="50%, *"). The first column (left) will always be 50% of the width of the browser window and the second column (right) will take up the remaining width, regardless of the window size. Since

Example:

Rows Frameset

Frameset:

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
```

```
</head>
<frameset rows="50%,80%,20%" >
    <frame src="">
    <frame src="">
    <frame src="">
</frameset>
</html>
```

Nested Framesets

To create a nested frameset, you can place one `<frameset>` inside another `<frameset>`. This allows you to create more complex layouts with both rows and columns. Here is an example demonstrating how to nest framesets:

Example: Rows and Columns Frameset

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Row and Column Frames Example</title>
</head>
<frameset rows="30%,70%">
    <frame src="" name="">
    <frameset cols="50%,50%">
        <frame src="" name="">
        <frame src="" name="">
    </frameset>
</frameset>
</html>
```



Operation Sheet : 3-2

Title: Create comprehensive Frameset with Target Attributes

Objective: This operation sheet provides step-by-step instructions on how to demonstrate the target attribute with frames

Tools Needed

- A text editor (like VSCode, Sublime Text, or Notepad++)

- A web browser (like Chrome, Firefox, or Safari)

Instructions: To demonstrate the target attribute with frames, consider the following example:create simple HTML files to the linked pages like target.html,menu.html, content.html,page1.html,page2.html and page3.html

<!-- Step 1: Create content.html file ? -->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Content</title>
</head>
<body>
    <h1>Welcome to the content frame!</h1>
    <p>Select a page from the menu to display here.</p>
</body>
</html>
```

<!-- Step 2: Create the page1.html? -->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Page 1</title>
</head>
<body>
    <h1>This is Page 1</h1>
</body>
</html>
```

<!-- Step 3: Create the page2.html -->

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <title>Page 2</title>
</head>
<body>
  <h1>This is Page 2</h1>
</body>
</html>

<!-- Step 4: Create the page3.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Page 3</title>
</head>
<body>
  <h1>This is Page 3</h1>
</body>
</html>

<!-- Step 5: Create the menu.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Menu</title>
</head>
<body>
  <h2>Menu</h2>
  <ul>
    <li><a href="page1.html" target="content">Page 1</a></li>
    <li><a href="page2.html" target="content">Page 2</a></li>
    <li><a href="page3.html" target="content">Page 3</a></li>
    <li><a href="https://www.example.com" target="_blank">External Link</a></li>
```

```
</ul>
</body>
</html>
<!-- Step 6: Create the target.html-->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Frameset Target Example</title>
</head>
<frameset cols="25%,75%">
    <frame src="menu.html" name="menu">
    <frame src="content.html" name="content">
</frameset>
</html>
<!-- Step 7: open the target.html-->
```

No Frames Fallback

Since frames are not supported by all browsers and can present accessibility issues, it's important to provide alternative content using the `<noframes>` tag.

Example of No Frames Fallback:

```
<!DOCTYPE html>
<html>
    <head>
        <title>No Frames Example</title>
    </head>
    <frameset cols="50%,50%">
        <frame src="frame1.html">
        <frame src="frame2.html">
    <noframes>
        <body>
            <p>Your browser does not support frames. Here is the content:</p>
            <p><a href="frame1.html">Frame 1</a></p>
            <p><a href="frame2.html">Frame 2</a></p>
```

```
</body>  
</noframes>  
</frameset>  
</html>
```

Modern Alternatives

While traditional frames are obsolete, iframes are still used to embed content from another source within a web page.

HTML Iframe is used to display a nested webpage (a webpage within a webpage). The HTML `<iframe>` tag defines an inline frame, hence it is also called as an Inline frame. In other ways, an iframe or inline frame is used to display external objects including other web pages within a web page.

An HTML iframe embeds another document within the current HTML document in the rectangular region.

The webpage content and iframe contents can interact with each other using JavaScript.

The basic syntax for adding an iframe to a web page can be given with:

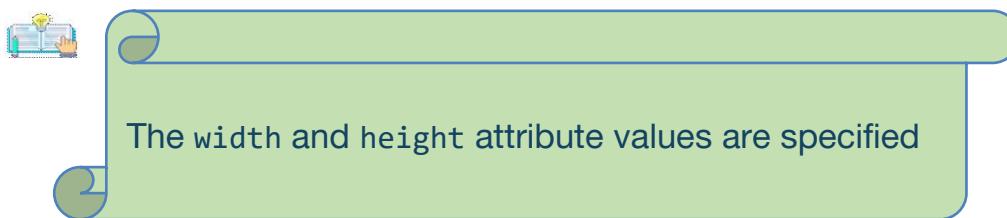
```
<iframe src="URL"></iframe>
```

The URL specified in the src attribute points to the location of an external object or a web page.

Setting Width and Height of an iFrame

The height and width attributes are used to specify the height and width of the iframe.

```
<iframe src="url or.html" width="400" height="200"></iframe>
```



Example of an `<iframe>`:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Iframe Example</title>
```

```
</head>
<body>
    <h1>Embedding an Iframe</h1>
    <iframe src="https://www.example.com" width="600" height="400" title="Example Iframe"></iframe>
</body>
</html>
```

3.7 HTML Form

The HTML forms are one of the essential components of the web that enables the user to submit the information accordingly on the website. Forms allow you to collect data or pass information from users. An HTML form is used to collect and store user input. The user input is most often sent to a server for processing. When we move into JavaScript, you'll be able to collect inputs or data from a user and act upon it.

In simple words, we can refer to the HTML forms as an easy way of collecting user information and then sending the collected information to the server. The HTML forms consist of a set of form elements that allow the user to enter the data, make selections, and then submit it to the server for processing. There are different types of form elements called input controls such as the text box, checkbox, radio buttons, drop-down, submit button, etc.

3.7.1. Importance of HTML Forms

The importance of HTML forms is explained below:

- They are used to gather information from the user like his contact details, survey responses, feedback, user registrations, login, signup, search box, order and checkout forms, and more.
- It is also used for validation which makes sure the data submitted is correct and accurate.
- They also enhance the user experience as when the user submits his review of the website the developer can change the website according to the reviews.
- They also provide security as when we use these forms in login or signup forms then only the valid candidate can access our website.
- We can also have some additional features in the form like file upload, etc, which will make the user upload the details.

3.7.2. HTML Form Tag

The <form> tag tells the browser where the form starts and ends. You can add all kinds of HTML tags between the <form> and </form> tags.

Syntax:

```
<form>
    //Form elements or input controls ...
</form>
```

3.7.2.1. Attributes of the FORM tag

The HTML <form> tag has specific attributes that define the behavior of the form itself. The Form tag supports several attributes that control the behavior and appearance of the form and how its data is submitted, such as action, method, enctype, target, name, autocomplete, and more.

Example 1:

```
<form action="index.php" method="POST/GET" name="Form1">
    //... form elements go here ...
</form>
```

Dear Students! Now, let's explore each attribute form tag in detail. It is crucial to bear in mind that, at the present moment, all of these attributes are not compulsory and can be omitted when using them in the Form tag.

- action attribute:
 - The action attribute specifies the URL to which the form data should be submitted upon submission.
 - This URL can be relative or absolute. If it's relative, it's interpreted relative to the current page's URL.
 - If the action attribute is omitted, the form data is submitted to the current page URL (also known as the "self" URL).
 - Example: <form action="/submit_form" method="post">

- method attribute:
 - The method attribute specifies the HTTP method used to submit the form data.
 - The two primary methods are GET and POST:
 - GET: Appends form data to the URL as query parameters. Suitable for retrieving data from the server.
 - POST: Sends form data in the request body. Suitable for submitting sensitive or large amounts of data.
 - If the method attribute is omitted, the default method is GET.
 - Example: <form action="/submit_form" method="post">
- enctype attribute: Specifies the encoding type used to submit form data to the server. This attribute is especially relevant when the form includes file uploads. Common values include application/x-www-form-urlencoded, multipart/form-data, and text/plain.
 - Example: <form enctype="multipart/form-data">
- target Attribute: Determines where the response from submitting the form will be displayed. The _self value loads the response in the same window or frame, while _blank opens it in a new window or tab. Other values include _parent, _top, and the name of a specific frame.
 - Example:<form target="_blank">
- name Attribute: Provides a name for the form, which can be useful when scripting or styling. It's also used to reference the form in client-side scripting.
 - Example:<form name="myForm">
- autocomplete Attribute: Controls whether the browser should automatically fill in form fields based on the user's browsing history and previously entered data. Values are on (default) and off.
 - Example:<form autocomplete="on">

Table 1-9: A comparison between the GET and POST methods

3.7.3. HTML Forms Structure

The basic structure of an HTML form consists of the following elements:

1. <form> tag: This is used to define the beginning and end of the form. It contains all the form elements that the user will interact with.

2. Form fields: These are the individual elements that make up the form, such as text boxes, dropdown lists, checkboxes, and radio buttons. Form fields are created using the <input> tag, and each field has its own unique attributes.

3. Submit button: This is a button that the user clicks on to submit the form data to the server for processing. It is created using the <input> tag with a type of "submit".



Tip: Nesting <form> elements within another <form> element is not allowed in HTML. This is because it can cause issues with form submission and data processing. However, you can organize your

| Feature | GET | POST |
|-----------------|--|--|
| Data in Request | Appends data to URL as query parameters | Sends data in the request body |
| Visibility | Data is visible in URL | Data is not visible in URL |
| Caching | Results can be cached by browser and intermediaries | Results are typically not cached |
| Security | Less secure as parameters are visible in browser history and server logs | More secure as data is not visible in URL |
| Data Limit | Limited (~2048 characters) | No specific limit |
| Idempotent | Idempotent (repeating request does not cause side effects) | Non-idempotent (may cause side effects on server) |
| Usage | Retrieving data, non-sensitive requests | Submitting sensitive or large data, modifying server state |

Here's an example of a basic HTML form structure:

```
<!-- Omitting both action and method attributes -->
<form>
    <label>Username: <input type="text"></label>
    <label>Password: <input type="password"></label>
    <input type="submit" value="Submit">
</form>
```

3.7.4. HTML `<form>` element

The HTML `<form>` element provides a document section to take input from the user. It provides various interactive controls for submitting information to the web server, such as a text field, a text area, a password field, etc.

1. HTML `<input>` tag:

- Used to create various types of input fields such as text input, password input, checkbox, radio buttons, etc.

2. HTML `<label>` tag:

- Provides a label for an `<input>`, `<select>`, `<textarea>`, or `<button>` element.
- Clicking on the label focuses on the associated form control.

3. HTML `<button>` tag:

- Creates a clickable button.
- Can be used to submit forms, trigger JavaScript functions, or perform other actions.

4. HTML `<select>`, `<option>`, and `<optgroup>` tags:

- `<select>`: Creates a dropdown list.
- `<option>`: Defines an option in a dropdown list.
- `<optgroup>`: Groups related options within a dropdown list.

5. HTML `<textarea>` tag:

- Creates a multiline text input area.
- Allows users to enter multiple lines of text.

6. HTML <fieldset> tag:

- Groups related form elements together.
- Often used with <legend> to provide a title or description for the group.

7. HTML <legend> tag:

- Provides a title or caption for a <fieldset> element.
- Describes the purpose or content of the grouped form elements.

8. HTML <datalist> tag:

- Specifies a list of predefined options for an <input> element.
- Provides suggestions as the user types in an input field.

9. HTML <output> tag:

- Displays the result of a calculation or script.
- Typically used in conjunction with JavaScript to dynamically update content based on user input.

3.7.5. Insert different form types

3.7.5.1.The <input> Tag

The HTML <input> tag is used to create interactive controls in web forms to collect user data. This is the most commonly used tag within HTML forms. It allows you to specify various types of user information fields such as text, checkboxes, radio buttons, file, Submit etc.

Syntax:

```
<form>
<input type = "value" ... />
</from>
```

3.7.5.2. HTML Input attributes

Here are various types of <input> elements along with their attributes and examples:

Table 1-2: The description of input control attributes in form

| Attribute | Values | Description | Example |
|-------------|---|---|--|
| type | text, password, email, number, date, radio, checkbox, file, submit,reset, button etc. | Specifies the type of input. | <pre><input type="text">, <input type="password">, <input type="email"></pre> <ul style="list-style-type: none"> • Values: text, password, email, number, date, radio, checkbox, file, submit, etc. |
| name | Any string. | Specifies the name of the input field. | <pre><input type="text" name="username"></pre> |
| value | Any string or number, depending on the input type. | Specifies the default value of the input field. | <pre><input type="text" value="JohnDoe"></pre> |
| placeholder | Any string | Provides a hint to the user. | <pre><input type="text" placeholder="Enter your username"></pre> |
| required | None (boolean attribute) | Indicates a required field. | <pre><input type="text" required></pre> |
| maxlength | Any positive integer | Specifies the maximum number of characters allowed. | <pre><input type="text" maxlength="15"></pre> |

| Attribute | Values | Description | Example |
|------------|--------------------------|---|--|
| min or max | Any number or date | Defines the minimum and maximum values | <code>input type="number" min="1" max="10"><input type="date" min="2023-01-01" max="2024-01-01"></code> |
| readonly | None (boolean attribute) | Makes the input field read-only. | <code><input type="text" readonly></code> |
| disabled | None (boolean attribute) | Disables the input field. | <code><input type="text" disabled></code> |
| pattern | A regular expression | Specifies a regular expression for validation. | <code><input type="text" pattern="[A-Za-z]{3}"></code> |
| autofocus | None (boolean attribute) | Automatically focuses the input field on page load. | <code><input type="text" autofocus></code> |
| size | Any positive integer | Specifies the visible width in characters. | <code><input type="text" size="20"></code> |
| step | Any positive number | Specifies the legal number intervals. | <code><input type="number" step="1">, <input type="date" step="7"></code> |
| multiple | None (boolean attribute) | Allows multiple values. | <code><input type="file" multiple>, <input type="email" multiple></code> |

| Attribute | Values | Description | Example |
|----------------|-------------------------------------|-------------------------------|--|
| autocapitalize | none, sentences, words, characters. | Controls text capitalization. | <input type="text" autocapitalize="words"> |

In this section, we will delve into the utilization of different HTML form controls that are frequently used in web development. It is important to explore their attributes within the <form> tag. By doing so, we can delve into the functionality and purpose of each control individually. These controls include:

- Text Input Control
- Password Control
- Radio Button Control
- Checkbox Control
- File Select box Control
- Submit Button Control
- Select dropdown and optgroup Control
- Fieldset Control

I. HTML Input Type Text

Text Input: Text fields are used for when you want the user to type short amounts of text into the form. Text fields are one line areas that allow the user to input text. Using a value of text for the type attribute on an input field will display a single-line text box for user input. The default width of a text input field is 20 characters, but the width can be manipulated using CSS.

Here's an example of a single-line text input used to take First name, Last name and Username:

Code Description: HTML code to create a text input in form

```
<!DOCTYPE html>
<form>
  <p> First name: <input type="text" name="firstname"></p>
  <p> Last name: <input type="text" name="lastname"></p>
```

```
<p>
<label for="username">Username:</label>
<input type="text" id="username" name="username" placeholder="Enter your username"
       required maxlength="20">
</p>
</form>
```

Output Description: *The output of the above example will look something like this:*

First name:

Last name:

Username:



Tip: The default value of the type attribute is "text".

TIPS

II. HTML Input Type Password

Password Input: Password fields are similar to text fields. The only difference is that characters in a password field are masked, i.e they are shown as asterisks (*) or dots (•). This is to prevent someone else from reading the password on the screen. This is also a single-line text input control created using an `<input>` element whose type attribute has a value of password.

Here's an example of a single-line password input used to take user password:

Code Description: *HTML code to create a password input in form*

```
<!DOCTYPE html>
<form>
    Password 1: <input type="password" name="User-Password" size="15" maxlength="15">
        <!-- password 1 is an optional field-->
    </br>
```

```
</br>
Password 2: <input type="password" name="User-Password" placeholder="Enter your password"
required>
<!-- password 2 is a required field and default size is equal to 20 --&gt;
&lt;/form&gt;</pre>
```

Output Description: *The output of the above example will look something like this:*

Password 1:

Password 2:

III. HTML Input Type Radio Buttons

Radio Buttons: Radio buttons are used to let the user select exactly one option from a predefined set of options. It is created using an `<input>` element whose `type` attribute has a value of `radio`.

Here's an example of radio buttons that can be used to collect user's gender information:

Code Description: *HTML code to create a Gender radio buttons in form*

```
<!DOCTYPE html>
<form>
  <h3>Gender:</h3>
  <input type="radio" name="gender" id="male">
  <label for="male">Male</label>
  <input type="radio" name="gender" id="female" checked>
  <!--if add checked for select female as default value - ->
  <label for="female">Female</label>
</form>
```

Output Description: *The output of the above example will look something like this:*

Gender:

Male Female

To set a default value for a radio button, you can use the checked attribute in the HTML markup. When the checked attribute is added to a radio button input element, that radio button will be selected by default when the form loads. When the form is displayed in the browser, the "Female" radio button will be selected by default because it has the checked attribute. Users can still change their selection by clicking on one of the other radio buttons.

IV. HTML Input Type Checkboxes

Checkboxes: They are similar to the radio buttons, but they enable the user to make multiple selections. It is created using an `<input>` element whose type attribute has the value of a checkbox. It should be used when you want to allow users to make more than one selection.

Here's an example of checkboxes that can be used to collect information about user's hobbies:

Code Description: *HTML code that created a hobbies checkbox buttons in form*

```
<!DOCTYPE html>
<form>
Hobby:<br>
<input type="checkbox" id="football" name="football" value="football"/>
<label for="football">Football</label> <br>
<input type="checkbox" id="read" name="read" value="Reading Book"/>
<label for="read">Reading Book</label> <br>
<input type="checkbox" id="watch" name="watch" value="Watch TV"/> Watch TV
</form>
```

Output Description: *The output of the above example will look something like this:*

Hobby:

- Football
- Reading Book
- Watch TV



If you want to make a radio button or checkbox selected by default, you can add the attribute checked to the input element, like <input type="checkbox" checked>.

V. File Select box

The file fields allow a user to browse for a local file and send it as an attachment with the form data. Web browsers such as Google Chrome and Firefox render a file select input field with a Browse button that enables the user to navigate the local hard drive and select a file. This is also created using an <input> element, whose type attribute value is set to file.

Here's an example of file fields that can be used to upload file:

Code Description: *HTML code that created a choose file buttons in form*

```
<!DOCTYPE html>
<form>
    <label for="file-select">Upload:</label>
    <input type="file" name="upload" id="file-select">
</form>
</form>
```

Output Description: *The output of the above example will look something like this:*

Upload: No file chosen

VI. Submit and Reset Buttons

A submit button is used to send the form data to a web server. When the submit button is clicked the form data is sent to the file specified in the form's action attribute to process the submitted data. A reset button resets all the form controls to default values. Try out the following example by typing your name in the text field, and click on submit button to see it in action.

Here's an example of file fields that can be used to Submit and Reset Buttons

Code Description: *HTML code that created Submit and Reset Buttons*

```
<!DOCTYPE html>
<form action="" method="post">
<P>
First Name:<input type="text" name="first-name" id="first-name">
</P>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>
```

Output Description: *The output of the above example will look something like this:*

First Name:

Dear teacher! Encourage your students to explore and engage in discussions about the different input types available in HTML form. By doing so, they will not only gain a deeper understanding of them but also obtain practical skills on how to create each input type effectively. Your guidance and support in this endeavor will undoubtedly contribute to their overall growth and development in the field of web design.

Dear Students! Create an HTML form with a button, color, Textarea ,date, time, week, month, datetime-local, email, hidden, image, number, range, search ,tel and url input types or fields.

VII. Select Dropdown

A select box is a dropdown list of options that allows the user to select one or more options from a pull-down list of options. Select box is created using the `<select>` element and `<option>` element. The `<option>` elements within the `<select>` element define each list item.

Here's an example of file fields that can be used to Select dropdown

Code Description: *HTML code that created Select City dropdown field*

```
<!DOCTYPE html>
<form action="" method="post">
<label for="city">City:</label>
<select name="city" id="city">
```

```

<option selected disabled>--Select City--</option>
    <option value="AA">Addis Ababa</option>
    <option value="Adama">Adama</option>
    <option value="BahirDar">Bahir Dar</option>
    <option value="diredawa">Dire Dawa</option>
</select>
</form>

```

Output Description: The output of the above example will look something like this:

City:

Select with Option group

Optgroup element is used to group multiple options in the select dropdown. The name of optgroup is set using the label attribute in optgroup.

Here's an example of file fields that can be used to Select Multiple item

Code Description: HTML code that created select an Option from group

```

<!DOCTYPE html>
<b><i>Please select Multiple subject form Grade 9 or 10 </i></b><br>
<select multiple>
    <option selected disabled>--Select Subject--</option>
    <optgroup label="Grade 9">
        <option>MATHEMATICS</option>
        <option>English</option>
        <option>BIOLOGY</option>
        <option>CHEMISTRY</option>
    </optgroup>
    <optgroup label="Grade 10">
        <option>MATHEMATICS</option>
        <option>English</option>
        <option>BIOLOGY</option>
        <option>CHEMISTRY</option>
    </optgroup>
</select>

```

```
</select>
```

Output Description: This will produce the following result:

Please select Multiple subject form Grade 9 or 10

| |
|------------------|
| -Select Subject- |
| Grade 9 |
| MATHEMATICS |
| English |

VIII. Grouping Form Controls

Fieldset Controls: The `<fieldset>` element in HTML is used to group related elements within a form. Form controls can also be placed inside the fieldset tag. Fieldset tag is used to group form or multiple input controls. Fieldset group form controls in the bordered area. We can also use legend tag inside the fieldset. The `<legend>` element is used to provide a caption for the fieldset.

Here's an example of file fields that can be used to Form bordered area

Code Description: HTML code that created a form bordered area.

```
<!DOCTYPE html>
<fieldset>
  <legend>Fill Form</legend>
  <form>
    <label>Name:<input type="text"></label>
    <input type="submit">
    <input type="reset">
  </form>
</fieldset>
```

Output Description: This will produce the following result:

Fill Form

| | | |
|----------------------------|---------------------------------------|--------------------------------------|
| Name: <input type="text"/> | <input type="button" value="Submit"/> | <input type="button" value="Reset"/> |
|----------------------------|---------------------------------------|--------------------------------------|



Operation Sheet : 2-1

Title: Create student registration form

Objective: This operation sheet provides step-by-step instructions on how to create a detailed HTML form that can use various input types for better user experience.

Tools Needed

- A text editor (like VSCode, Sublime Text, or Notepad++)
- A web browser (like Chrome, Firefox, or Safari)

Instructions: Write the HTML code that produce the below output named by registration form

Registration Form:

Name:

Password:

Gender: Male Female

Birthday:

Education Background:

Masters
 Degree
 Level 4 & 5
 Level 1 & 3

Email:

City:

Address:

<!-- Step 1: Create a basic HTML file ?

Task 1: Open a new file in a notepad or Notepad++

Task 2: Add the basic structure of a HTML file

Task 3: Save it as a Reg_form .html file

-->

<! -- Step 2: Create the form ?

Inside the “body” create the input element “form”.

-->

<!-- Step 3: Add input elements in the form fields -->**<!-- Task 1: Enter fieldset tag -->**

```
<fieldset>
```

```
<legend>Registration Form:</legend>
```

<!-- Task 2: Enter Name input using label tag -->

```
<label for="name">Name:</label>
```

```
<input type="text" id="name" name="name" size="25" placeholder="Enter your  
name"><br><br>
```

<!-- Task 3: Enter Password input using label tag -->

```
<label for="password">Password:</label>
```

```
<input type="password" id="password" name="password" size="22" placeholder="Enter your  
password"><br>
```

<!-- Task 4: Enter Gender radio buttons -->

```
</br>
```

```
<label>Gender:</label>
```

```
<input type="radio" id="male" name="gender" value="male">
```

```
<label for="male">Male</label>
```

```
<input type="radio" id="female" name="gender" value="female" checked>
```

```
<label for="female">Female</label><br>
```

```
</br>
```

<!-- Task 5: Enter Birthday input field -->

```
<label for="birthday">Birthday:</label>
```

```
<input type="date" id="birthday" name="birthday"><br><br>
```

<!-- Task 6:Enter Education checkboxes -->

```
<label>Education Backgound:</label><br>
<input type="checkbox" id="education1" name="education" value="education1">
<label for="education1">Masters</label><br>
<input type="checkbox" id="education2" name="education" value="education2">
<label for="education2">Degree</label><br>
<input type="checkbox" id="education3" name="education" value="education3">
<label for="education3">Level 4 & 5</label><br>
<input type="checkbox" id="education4" name="education" value="education4">
<label for="education3">Level 1 & 3</label><br>
<br>

<!-- Task 7: Enter email field box -->
<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>

<!-- Task 8: Enter City selection dropdown -->
<label for="city">City:</label>
<select id="city" name="city">
    <option value="city1">City1</option>
    <option value="city2">City2</option>
    <option value="city3">City3</option>
</select><br><br>

<!-- Task 9: Enter Address textarea -->
<label for="address">Address:</label><br>
<textarea id="address" name="address" rows="4" cols="50"></textarea>
<br>

<!-- Task 10: Finally create Register and Reset buttons -->
<button type="submit">Register</button>
<button type="reset">Reset</button>
</fieldset>
</form>
</body>
```

Self-check 3-5:

1. Create an HTML form with a number input field for entering a quantity of products to purchase.
2. Produce an HTML form with a time input field for scheduling appointments.
3. Make an HTML form with a week input field for selecting a week for project deadlines.

3.7.6. HTML Form Validation

HTML form validation is used to validate form data before the data is forwarded or saved in any format. Automatic validation can be as easy as adding a required attribute to any given form input, requiring the input to be filled before submission. Unfortunately, with just HTML there is no real way to validate the data itself, just that it exists.

In order to validate the data to ensure that the user input is clean, correct and useful, we need to employ the use of JavaScript. Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- has the user filled in all required fields?
- has the user entered a valid date?
- has the user entered text in a numeric field?



Most often, the purpose of data validation is to ensure correct user input. Validation can be defined by many different methods, and deployed in many different ways in the next javascript Module. Server-side validation is performed by a web server, after input has been sent to the server. Client-side validation is performed by a web browser, before input is sent to a web server.

Types of Validation

- **Built-in Validation Attributes:**

- The **no validate** attribute on the form element disables the browser's default validation UI, allowing custom validation handling.
- **required**: Ensures that the user does not leave the field empty.
For example: <input type="text" name="username" required>
- **pattern**: Validates the input against a regular expression. The pattern attribute on the name input enforces that only letters and spaces are allowed, and the title attribute provides a custom validation message.
For example: <input type="text" name="username" pattern="[A-Za-z\s]+" title="Only letters and spaces are allowed.">
- **type**: Specifies the type of input (e.g., name , age,email, url, number, date, time, week, month, etc.).
For example: <input type="email" name="useremail" required>
- **min and max**: Specify the minimum and maximum values for numeric inputs.
For example: <input type="number" name="age" min="18" max="100">
- **minlength and maxlength**: Specify the minimum and maximum number of characters for text inputs.
For example: <input type="text" name="username" minlength="5" maxlength="15">
- **step**: Specifies the legal number intervals for numeric inputs.
For example: <input type="number" name="steps" step="5">
- **multiple**: Allows multiple values to be entered (e.g., multiple email addresses).
- For example: <input type="email" name="emails" multiple>

- **Custom Validation:**

- **JavaScript**: Use JavaScript to implement custom validation logic when built-in validation attributes are not sufficient.

Example 1:

This example demonstrates how to use HTML5 form validation attributes to enforce input constraints and provide user-friendly validation feedback.

Code Description: *HTML code to create a data validation input in form*

```
<form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required pattern="[A-Za-z\s]+" title="Name should
only contain letters and spaces.">
    <br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
    <br><br>
    <label for="age">Age:</label>
    <input type="number" id="age" name="age" min="1" max="100" required>
    <br><br>
    <button type="submit">Submit</button>
</form>
```

Output Description: *How the HTML code above looks in a browser*

Name:

Email:

Age:

3.8 Multimedia elements

Multimedia and embedding

We've looked at a lot of text so far in this course, but the web would be really boring only using text. Let's start looking at how to make the web come alive with more interesting content! This module explores how to use HTML to include multimedia in your web pages, including the different ways that images can be included, and how to embed video, audio, and even entire webpages.

Prerequisites

Before starting this module, you should have a reasonable understanding of the basics of HTML, as previously covered in Introduction to HTML. If you have not worked through this module (or something similar), work through it first, then come back!

This module demonstrates the utilization of HTML elements for embedding diverse forms of multimedia content. It encompasses images, audio, and video, as well as external content such as YouTube videos and Flash SWF files.

3.8.1 Embedding audio and video using audio and video tags

Images in HTML

There are other types of multimedia to consider, but it is logical to start with the humble `` element used to embed a simple image in a webpage. In this article we'll look at how to use it in more depth, including basics, annotating it with captions using `<figure>`, and how it relates to CSS background images.

Video and audio content

Next, we'll look at how to use the HTML `<video>` and `<audio>` elements to embed video and audio on our pages, including basics, providing access to different file formats to different browsers, adding captions and subtitles, and how to add fallbacks for older browsers.

From `<object>` to `<iframe>` — other embedding technologies

At this point we'd like to take somewhat of a sideways step, looking at a couple of elements that allow you to embed a wide variety of content types into your webpages: the `<iframe>`, `<embed>` and `<object>` elements. `<iframe>`s are for embedding other web pages, and the other two allow you to embed external resources such as PDF files.

Adding vector graphics to the Web

Vector graphics can be very useful in certain situations. Unlike regular formats like PNG/JPG, they don't distort/pixelate when zoomed in — they can remain smooth when scaled. This article introduces you to what vector graphics are and how to include the popular SVG format in web pages.



Note: Different browsers support different audio and video formats. Providing multiple formats ensures compatibility across all browsers.

Multimedia and embedded HTML elements, including their attributes and popular file extensions, will be discussed along with examples in this section.

- **Image ():**

- Attributes: src, alt, width, height
- File Extensions: .jpg, .jpeg, .png, .gif
- Example:

```

```

- **Audio (<audio>):**

- Attributes: src, controls
- File Extensions: .mp3, .ogg, .wav
- Example:

```
<audio src="example.mp3" controls>
```

Your browser does not support the audio element.

```
</audio>
```

- **Video (<video>):**

- Attributes: src, controls, width, height
- File Extensions: .mp4, .webm, .ogg
- Example:

```
<video src="example.mp4" controls width="400" height="300">
```

Your browser does not support the video element.

```
</video>
```

- **Iframe (<iframe>):**

- Attributes: src, width, height, frameborder, allowfullscreen
 - Example:
- ```
<iframe src="https://www.youtube.com/embed/example" width="560" height="315" frameborder="0" allowfullscreen></iframe>
```

- **Object (<object>):**

- Attributes: data, type, width, height
- File Extensions: Varies depending on the content (e.g., .swf for Flash, .pdf for PDF)
- Example (Flash SWF File):

```
<object type="application/x-shockwave-flash" data="example.swf" width="400"
height="300">
 <param name="movie" value="example.swf">
 <param name="quality" value="high">
</object>
```

### **3.8.2 Adding source, controls, and fallback content**

In web development, adding source, controls, and fallback content is crucial for creating robust and user-friendly multimedia experiences, particularly when dealing with HTML5 `<audio>` and `<video>` elements. Here's a detailed guide on how to incorporate these features effectively:

#### **3.8.2.1 Adding Source Elements**

The `<source>` element allows you to specify multiple media files for browsers to choose from. This is useful for providing different file formats to ensure compatibility across all browsers.

##### **Example for <video>:**

```
<video controls>
 <source src="movie.mp4" type="video/mp4">
 <source src="movie.ogg" type="video/ogg">
 Your browser does not support the video tag.
</video>
```

##### **Example for <audio>:**

```
<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
 <source src="audio.ogg" type="audio/ogg">
 Your browser does not support the audio element.
</audio>
```

#### **3.8.2.2. Adding Controls**

---

The controls attribute adds play, pause, and volume controls to the media element. This enhances the user experience by providing built-in functionality without requiring additional JavaScript.

**Example code:** Video with Controls

```
<video controls>
 <source src="movie.mp4" type="video/mp4">
 <source src="movie.ogg" type="video/ogg">
 Your browser does not support the video tag.
</video>
```

**Example code:** Audio with Controls

```
<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
 <source src="audio.ogg" type="audio/ogg">
 Your browser does not support the audio element.
</audio>
```

### 3.8.2.3 Adding Fallback Content

Fallback content is crucial for providing an alternative message or link when the browser does not support the `<audio>` or `<video>` elements. This ensures that all users have access to your content, even if they are using an outdated browser.

**Example code:** Video with Fallback Content

```
<video controls>
 <source src="movie.mp4" type="video/mp4">
 <source src="movie.ogg" type="video/ogg">
 Your browser does not support the video tag.
 Download the video
</video>
```

**Example code:** Audio with Fallback Content

```
<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
 <source src="audio.ogg" type="audio/ogg">
 Your browser does not support the audio element.
 Download the audio
```

```
</audio>
```

Inclusive Example for <video>

```
<video controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
Download the video
</video>
```

Inclusive Example for <audio>

```
<audio controls>
<source src="audio.mp3" type="audio/mpeg">
<source src="audio.ogg" type="audio/ogg">
Your browser does not support the audio element.
Download the audio
</audio>
```



**Tip:** Different browsers support different audio and video formats. Providing multiple formats ensures compatibility across all browsers. Include at least MP4 and WebM for video, and MP3 and Ogg for audio.



### Self-check 3-6:

1. How do you embed a video using the HTML <video> tag?
2. What is the purpose of the controls attribute in the <audio> and <video> tags?
3. Why should you include multiple <source> elements for a single media file?
4. What is fallback content, and why is it important?
5. How do you add subtitles or captions to a video?
6. What is the purpose of the preload attribute in <audio> and <video> tags?

# Unit Summary

## Unit Review Questions

## Multiple Choice Questions

**Instruction:** Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. Which tag is used to create a hyperlink in HTML?  
A. <a>  
B. <link>  
C. <href>  
D. <hyperlink>
  2. Which attribute specifies an alternate text for an image, if the image cannot be displayed?  
A. src  
B. alt  
C. title  
D. href
  3. Which tag is used to define an unordered list?  
A. <ul>  
B. <ol>  
C. <li>  
D. <list>
  4. What does the <thead> tag represent in a table?  
A. The main body of the table  
B. The footer of the table  
C. The head of the table  
D. The title of the table
  5. Which attribute is used to merge two or more table cells horizontally?  
A. colspan  
B. rowspan  
C. merge  
D. span
  6. Which HTML tag is used to create a drop-down list?  
A. <list>  
B. <dropdown>  
C. <select>  
D. <option>
  7. Which tag is used to create a line break in HTML?  
A. <break>  
B. <br>  
C. <lb>  
D. <newline>

8. Which tag is used to create a numbered list?

  - A. <ul>
  - B. <ol>
  - C. <li>
  - D. <dl>

9. What is the purpose of the target attribute in the <a> tag?

  - A. To define the URL of the hyperlink
  - B. To specify the name of the link
  - C. To specify where to open the linked document
  - D. To set the style of the hyperlink

10. Which tag is used to embed an image in an HTML page?

  - A. <picture>
  - B. <img>
  - C. <image>
  - D. <media>

11. Which tag is used to define a table in HTML?

  - A. <tab>
  - B. <table>
  - C. <tr>
  - D. <td>

12. Which attribute is used to specify the width of a table cell?

  - A. width
  - B. cellwidth
  - C. colspan
  - D. size

13. Which tag is used to create a text input field in a form?

  - A. <textarea>
  - B. <input type="text">
  - C. <textfield>
  - D. <input type="textfield">

14. Which attribute is used to specify the size of a text area in HTML?

  - A. rows
  - B. cols
  - C. size
  - D. Both A and B

15. Which tag is used to define a division or a section in an HTML document?

  - A. <div>
  - B. <section>
  - C. <span>
  - D. <part>

## Answer key for Self-check questions

### Self-check 3-1: Answer key

```
<html>
 <head>
 <title> My First Web Page </title>
 </head>
 <body bgcolor = Lime text = blue leftmargin = 72 topmargin = 72>
 Welcome to Computer Applications
 </body>
</html>
```

### Self-check 3-2: Answer key

#### 1. HTML code

**Code Description:** *HTML code that show the effect of font*

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Paragraph Example</title>
 </head>
 <body>
 <h1><u></u>Welcome to My Website</u></h1>
 <p> This is a paragraph of
text.</p>
 <p><mark>Another paragraph can start here.</mark></p>
 1st and 2nd year students
 </body>
</html>
```

2. Output will be :What output is produced by the HTML document code below?

# Main Heading

This is a basic paragraph of text.

## Subheading

You can also use *italic text* with the `<i>` tag and *emphasized text* with the `<em>` tag.

### Monospaced Text

Inline monospaced text is useful for code snippets.

This is a preformatted text block.

You can combine multiple formatting options in one paragraph.

### Self-check 3-3: Answer key

1. To link to a page in a subfolder, you would specify the path relative to the current location. For instance, if the page is `articles/tips.html`, the link would be `<a href="articles/tips.html">Read our Tips</a>`.
2. An image can be used as a hyperlink by wrapping the `<img>` tag with an `<a>` tag, like so: `<a href="products.html"></a>`.
3. A mailto link is created with the `mailto:` scheme in the `href` attribute, like this: `<a href="mailto:info@example.com">Email Us</a>`
4. To link to a downloadable file, you would use the path to the file in the `href` attribute, for example: `<a href="documents/instructions.pdf" download>Download Instructions</a>`.

5. To create a link that scrolls to the top of the current page, you can link to `#` or use a named anchor at the top of the page, like so: `<a href="#">Back to Top</a>` or if you have an anchor named `top`: `<a href="#top">Back to Top</a>`.

### **Self-check 3-4: Answer key**

```
<table border="1">
<caption>Invoice</caption>
<tr> <th>Item</th> <th>Quantity</th> <th>Price</th> <th>Amount</th> </tr>
<tr> <td>Computer Box</td> <td>2,000</td> <td>1,000</td> <td>2,000,000</td> </tr>
<tr> <td>Paper</td> <td>100</td> <td>800</td> <td>80,000</td> </tr>
<tr> <td>USB Flash</td> <td>600</td> <td>1500</td> <td>900,000</td> </tr>
<tr> <th colspan="3">Subtotal</th> <td>2,980,000</td></tr>
<tr> <th colspan="2">Tax</th> <td>15%</td> <td>447,000</td> </tr>
<tr> <th colspan="3">Total</th> <td>3,427,000</td> </tr>
</table>
```

### **Self-check 3-5: Answer key**

1. Number input field for entering a quantity of products to purchase.

**Code:**

```
<form>
<label for="quantity">Quantity:</label>
<input type="number" id="quantity" name="quantity" min="1" required>
<button type="submit">Purchase</button>
</form>
```

2. Time input field for scheduling appointments.

**Code:**

```
<form>
<label for="appointment-time">Choose a time:</label>
<input type="time" id="appointment-time" name="appointment-time" required>
<button type="submit">Schedule</button>
```

```
</form>
```

**Output:**

3. Week input field for selecting a week for project deadlines.

**Code:**

```
<form>
 <label for="deadline-week">Select a week:</label>
 <input type="week" id="deadline-week" name="deadline-week" required>
 <button type="submit">Set Deadline</button>
</form>
```

**Self-check 3-6: Answer key**

1. To embed a video using the HTML `<video>` tag, you include the `<video>` element with appropriate attributes and `<source>` elements for the video files.

**Code:**

```
<video controls>
 <source src="video.mp4" type="video/mp4">
 <source src="video.webm" type="video/webm">
 Your browser does not support the video tag.
</video>
```

2. The `controls` attribute adds built-in controls (play, pause, volume, etc.) to the media player. This allows users to interact with the media without requiring custom JavaScript controls.

**Code:**

```
<video controls>
 <source src="video.mp4" type="video/mp4">
</video>
<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
</audio>
```

3. Including multiple `<source>` elements ensures that the media is compatible with different browsers and devices, as not all browsers support the same media formats. By providing

alternatives like MP4, WebM, and Ogg, you maximize the chances that your media will play for all users.

**Code:**

```
<video controls>
 <source src="video.mp4" type="video/mp4">
 <source src="video.webm" type="video/webm">
 <source src="video.ogv" type="video/ogg">
 Your browser does not support the video tag.
</video>
```

4. Fallback content is a message or link provided inside the `<audio>` or `<video>` tags that is displayed if the browser does not support these elements. It ensures that users still have access to the content even if they can't play the media directly in their browser.

**Code:**

```
<video controls>
 <source src="video.mp4" type="video/mp4">
 Your browser does not support the video tag.
 Download the video
</video>

<audio controls>
 <source src="audio.mp3" type="audio/mpeg">
 Your browser does not support the audio element.
 Download the audio
</audio>
```

5. Answer: You add subtitles or captions to a video using the `<track>` element within the `<video>` tag. The `<track>` element specifies the source of the subtitles and other necessary attributes.

**Code:**

```
<video controls>
 <source src="video.mp4" type="video/mp4">
 <track src="subtitles_en.vtt" kind="subtitles" srclang="en" label="English">
 Your browser does not support the video tag.
</video>
```

6. The `preload` attribute provides a hint to the browser about how much of the media file to download when the page loads. It can help balance performance and bandwidth usage.

- none: Do not preload the media.
- metadata: Preload only the metadata (e.g., length).
- auto: Preload the entire media file.

**Code:**

```
<video controls preload="auto">
 <source src="video.mp4" type="video/mp4">
</video>
<audio controls preload="metadata">
 <source src="audio.mp3" type="audio/mpeg">
</audio>
```

## References

1. <https://www.scaler.com/topics/html/>
2. <https://www.tutorialrepublic.com/html-tutorial/>
3. <https://www.w3schools.com/html/default.asp>
4. <https://www.javatpoint.com/html-tutorial>

# **Module V**

# **CASCADING STYLE SHEETS**

## Contents

| No       | Contents                                           | page |
|----------|----------------------------------------------------|------|
|          | <b>Module Description</b>                          | 449  |
| <b>1</b> | <b>Unit 1: Basic of Cascading Style Sheets</b>     | 450  |
| 1.1      | Overview of CSS                                    | 451  |
| 1.2      | Role of CSS                                        | 455  |
| 1.3      | CSS syntax and structure                           | 456  |
| 1.4      | CSS Comments                                       | 475  |
| 1.5      | Method to insert style sheets in HTML document     | 476  |
| 1.6      | CSS Properties                                     | 486  |
| 1.7      | CSS positioning and layout Properties              | 505  |
|          | Unit Summary                                       | 512  |
|          | Unit Review Questions                              | 513  |
| <b>2</b> | <b>Unit 2: Layout and navigation of a web page</b> | 518  |
| 2.1      | Division and multicolumn layouts                   | 518  |
| 2.2      | Layout using HTML div and table tags               | 522  |
| 2.3      | Navigation Formatting                              | 535  |
| 2.4      | CSS properties for the navigation bar              | 553  |
|          | Unit Summary                                       | 565  |
|          | Unit Review Questions                              | 565  |
|          | References                                         | 571  |

## Module Description

This Module provides Web developers with a standard way to define, apply, and manage sets of style characteristics. It offers these capabilities through a technical model based on a hierarchical scope of effect, the separation of style from content, and a well-defined set of published standards. It focused on how CSS can help the Web developer create maintainable, apply and reusable code.

This module will also assist you to attain the following learning outcomes:

- Apply basic Cascading Style Sheets/ CSS / Properties
- Create layout and navigation of a web page

### Module Instruction:

#### Learning Instructions: How to use this Module

**For effectively use this module you are expected to follow the following module instructions:**

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what is being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

## UNIT 1

### Basic of Cascading Style Sheets

#### Unit Coverage

*This unit is designed to provide you the necessary information and practice regarding the following content coverage:*

1. *Introduction to CSS*
2. *CSS Properties*

#### Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- Apply basic CSS selector and Properties
- Insert Style sheet in HTML

**Key Terms:** *CSS, properties, dv, Selector, Id, Class, Inline, Internal, External, Layout, and Position*

#### Unit Overview

This module emphasizes that CSS, also known as Cascading Style Sheets, is an essential technology to learn after HTML. Unlike HTML, which focuses on defining the structure and meaning of content, CSS is used to style and arrange it. With CSS, you can customize the font, color, size, and spacing of your content, create multi-column layouts, and incorporate animations and decorative elements. As Prerequisites, we recommend that you install basic software, have basic knowledge of working with files, and work through the HTML module first. This is because HTML is far more interesting and much more fun to learn when you apply CSS, and you can't learn CSS without knowing HTML.

## 1.1. Introduction of CSS

CSS (Cascading Style Sheets) is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to HTML documents. It describes how a web page should look. It prescribes colors, fonts, spacing, etc. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

HTML uses tags and CSS uses rulesets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

With CSS, developers can change the color, size, font, layout, and other visual aspects of HTML elements. By defining styles in a separate CSS file or embedding them within HTML documents, designers can apply consistent styles to an entire website or specific sections.

The "cascading" in Cascading Style Sheets refers to how property values are applied in the context of the parent/child hierarchy of the Web document. Child elements either inherit or override property values bound to their parent elements.

A style sheet is the encapsulation of style rules in a centralized location, either in the head section of the HTML document or in a separate linked file. The Web browser reads these styles and applies the specified formatting rules before displaying the content.

CSS or Cascading Style Sheets is a style sheet language used to add styles to the HTML document. It describes how HTML elements should be displayed on the web page. The reason for using CSS is to simplify the process of making web pages presentable. CSS allows web developers to control the visual appearance of web pages.

### 1.1.1. Short history

CSS was created in 1994 by Håkon Wium Lie, a web standards activist. Back then, he worked at CERN with Tim Berners-Lee, the creator of the World Wide Web, so he had good insight into HTML and the first websites. In 1995, he joined W3C (World Wide Web Consortium), the international standards organization for the Web, which was created by Berners-Lee. There, he worked with Bert Bos, a computer scientist, who created the Argo web browser as a test application for CSS. Together, they created the CSS1 (1996), CSS2 (1998) and CSS3 (1999) specifications.

The work of Håkon Wium Lie as a web standards activist has led to a number of significant changes in the World Wide Web. He campaigned for browsers to support downloadable web fonts, for easier video publishing (essentially proposing the <video> HTML tag for websites), and for printing directly from the Web, instead of copying content to a text/image editor first. CSS1 allowed web developers to specify font properties, background color, text alignment, padding and basic positioning of different elements.

CSS2 expanded the above functionalities by adding relative, absolute and fixed positioning, and media types – different styles for screens, printers, etc.

CSS3, with some minor revisions, is the current specification. Unlike the previous versions, CSS3 does not include a single set of features. Instead, it is divided into different modules, each one including various features. CSS3 includes things such as transitions, shadows, embedded fonts, rounded corners, and responsive design. As some modules are changed independently, some people refer to the latest updates of CSS3 and CSS4.

### **1.1.2. Advantages and Disadvantage of CSS**

#### **Advantages of CSS**

CSS has many advantages that enable developers to design a website. Some of these advantages are:

1. Better Website Speed: -For a website to function efficiently, it should have a faster load time. In modern times, people usually wait for just a couple of seconds for a website to load. So, it's important to ensure faster speed. For companies wanting to ensure a faster and smooth website experience, CSS becomes paramount to their success.
2. Easier to Maintain: - CSS is easy to maintain due to less maintenance time. This is because a single line code change affects the entire web page. Also, if improvements are required, then less effort is required to affect changes in the webpage code.
3. Consistent Design: -You would have seen many websites that are elegant and user-friendly. One thing common to all these websites is consistency in design. CSS enables developers to ensure the style elements are applied consistently across several web pages.

4. Time-Saving: -Due to faster speed and easier maintenance, CSS saves a lot of time and effort in the web development process due to faster loading time. Here, lesser time ensures designer efficiency.
5. Better Device Compatibility: -People use different smart devices to view a particular website. It can be a smartphone, PC or laptop. For this purpose, websites are required to be device compatible. CSS ensures the task is done smoothly by providing better compatibility.
6. Positioning of Design Elements: - You can change the position of an HTML tag with the help of CSS. You can place the elements like an image on any part of the webpage as and when required.

### **Disadvantages of CSS**

There are a few downsides while using CSS. One must know these disadvantages so that he or she is aware and takes care of them while designing a website.

1. Confusion due to many CSS levels: - Beginners are more vulnerable to this issue. They might get confused while opting to learn CSS as there are many levels of CSS such as CSS2, CSS3, etc.
2. Cross-Browser Issues:-Different browsers work differently. So, you have to check that changes implemented in the website via CSS codes are reflected properly among all browsers.
3. Security Issues:-Security is important in today's world driven by technology and data. One of the major disadvantages of CSS is that it has limited security.
4. Extra Work for Developers:-Design services are required to consider and test all CSS codes across different browsers for compatibility. Due to developers testing compatibility for different browsers, their workload increases.

Generally, we can say that if you are passionate about web development, try to learn HTML and CSS. For device compatibility, learn the Bootstrap framework as well. Though you might see some disadvantages of CSS, many advantages counter them and ensure that your web development process is smooth and efficient.

### 1.1.3. CSS Text Editors

A CSS text editor is a software that enables a developer to write and edit CSS codes. There are many editors available online. But, to make your job easier, we provide you with some of the popular ones.

*Sublime Text:* Known for its speed and versatility, Sublime Text is highly customizable with a vast array of plugins and packages available. It supports multiple programming languages, including CSS, and offers features like syntax highlighting, auto-completion, and a distraction-free mode.

*Notepad++:* It is a free and open-source text editor for Windows. It provides basic features like syntax highlighting, code folding, and auto-completion. While it may not be as feature-rich as some other editors, it's lightweight and fast, making it a popular choice for many developers.

*Atom:* Developed by GitHub, Atom is a modern and customizable text editor built on web technologies. It offers a rich ecosystem of packages and themes, making it highly extensible. Atom provides features like built-in Git integration, multiple panes, and a user-friendly interface.

*Stylizer:* It is a CSS editor specifically designed for web designers. It offers real-time editing and previewing of CSS styles, allowing you to see changes instantly. Stylizer also includes features like a visual CSS grid editor, color picker, and browser compatibility testing.

*Espresso:* It is a web development tool for macOS that includes a powerful CSS editor among its features. It provides syntax highlighting, code folding, and live previewing of CSS changes. Espresso also offers tools for HTML, JavaScript, and PHP development.

*Brackets:* It is an open-source text editor developed by Adobe. It's designed for web development and includes features like live previewing, inline editing, and preprocessor support. Brackets also offer a large collection of extensions to enhance its functionality.

Visual Studio Code (*VS Code*) is a popular source-code editor developed by Microsoft, and it includes robust features for editing CSS files. While it's not specifically a "CSS editor" in the traditional sense, it provides extensive support for editing CSS and other web-related languages.

**Dear Students!** *It is now your opportunity to provide answers to the following self-check questions.*

**Self-check 1-1:**

- 
1. What are the merits of using CSS in web development?
  2. What are the demerits or challenges associated with CSS?
  3. How has the adoption of CSS contributed to the evolution of web design and development?
- 

## 1.2. Role of CSS

With its primary focus on a website's presentation, CSS (Cascading Style Sheets) is an essential component of web development. CSS plays the following important roles:

*Styling:* The visual presentation of HTML elements is mostly styled with CSS. It gives you complete control over your web pages' visual elements, including fonts, colors, and layout. This covers a variety of settings, such as padding, margins, borders, and backgrounds.

*Responsive Design:* CSS enables developers to create responsive designs that adapt to different screen sizes and devices. Through techniques like media queries and flexible layouts, CSS helps ensure that websites look good and are functional across various devices, such as desktops, laptops, tablets, and smartphones.

*Layout Control:* CSS provides various layout techniques to position elements on a web page. This includes using CSS grid, flexbox, and positioning properties to create complex layouts and achieve specific design goals.

*Accessibility:* CSS plays a role in making websites accessible to users with disabilities. By properly structuring and styling content, developers can enhance readability and usability for all users, including those who rely on screen readers or have other accessibility needs.

*Consistency:* CSS allows developers to create consistent styles across multiple web pages or even an entire website. By defining styles in external CSS files and linking them to HTML

documents, you can maintain consistency in design elements like colors, typography, and spacing throughout your site.

*Ease of Maintenance:* Separating style from content using CSS makes it easier to maintain and update websites. By making changes to CSS files, you can update the visual appearance of your site without modifying the underlying HTML structure, which improves code maintainability and scalability.

*Animation and Effects:* CSS enables the creation of animations, transitions, and other visual effects without relying on JavaScript or other scripting languages. This allows developers to enhance user interactions and create engaging user experiences.

Generally, CSS is essential for creating visually appealing, responsive, accessible, and consistent web experiences. It works hand in hand with HTML and JavaScript to build modern, user-friendly websites.

## 1.3. CSS syntax and structure

### 1.3.1. CSS – Syntax

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. CSS syntax consists of selectors and declarations.

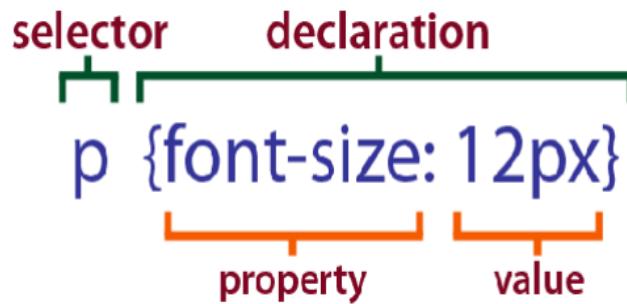
- *Selector* – A selector is an HTML tag at which a style will be applied. This could be any tag like `<h1>` or `<table>` etc.
- *Declarations*- A declaration consists of a property and a value, separated by a colon and terminated by a semicolon. Declaration blocks are surrounded by curly braces.
  - *Property* – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
  - *Value* – Values are assigned to properties. For example, color property can have value either red or `#F1F1F1` etc.

The syntax follows a simple structure:

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

For example:- `p { font-size: 12px }`

Let's provide a clearer explanation of the above CSS syntax, including selectors, properties, and values that define the styling of HTML elements.



*Figure 1.1: Describe of selector and declaration*

The syntax for Multiple CSS declarations:

**Selector {property1: value1; property 2: value2;.. ;}**

For example:-

```
h1 {
 color: blue; font-size: 24px; margin-top: 20px;
}
```

Let's provide a clearer explanation of the above CSS syntax, including selectors, properties, and values that define the styling of HTML elements.

- h1 is a selector (it This selector targets all <h1> elements in the HTML document).
- color: blue; This property sets the color of the text within the <h1> elements to blue.
- font-size: 24px; This property sets the font size of the text within the <h1> elements to 24 pixels.
- margin-top: 20px; This property sets the top margin of the <h1> elements to 20 pixels

### 1.3.2. Selectors

Selectors are used to find / select the HTML elements that are to be styled by CSS. They are a mechanism that selects a unique set of elements and applies CSS designs to them. This can be achieved using generic HTML properties or manually attaching an identifier to the elements

while coding them. For example, let's consider a simple scenario where a web application has 100 elements on the page that display information in textual form.

CSS selectors are the backbone of any stylish web page. They target HTML elements on your pages, allowing you to add styles based on their id, class, type, attribute, and more. This guide will delve into the intricacies of CSS selectors and their pivotal role in enhancing the aesthetics and user experience of your web pages.

## Types of Selector

There are the following different types of selectors in CSS.

1. Element selector
2. Id selector
3. Class selector
4. Universal selector
5. Group selector
6. Attribute selector

Now, let's learn about them in detail.

### 1. Element Selector

Element Selector, allows you to target and style HTML elements based on their tag names. It is one of the simplest and most straightforward selectors. With an element selector, you can apply styles to all occurrences of a particular HTML tag on a web page. The element selector selects HTML elements (p, div, h1, etc) and applies CSS to them.

#### Syntax:

```
element{ property: value; ...}
```

#### Description:

- **Element:** This is the name of the HTML element you want to target. It matches the tag name of the elements you want to style.
- **CSS Properties and Values:** Inside the curly braces, you define the styles you want to apply to the selected HTML element.

Let's see an example,

```
h1 { color: red;}
p { color: orange;}
```

In the above example, the element selector

- h1 selects all h1 elements and styles their color to red
- p selects all p elements and styles their color to orange



The element selector is also referred to as a tag selector because it selects HTML elements based on their tag names.

### Merits and demerits of the element selector

Merits:

- *Global Styling*: Element selectors are useful when you want to apply a style globally to all occurrences of a specific HTML element.
- *Default Styling*: They are commonly used for setting default styles for standard HTML elements like paragraphs, headings, links, etc.

Demerits:

- *Broad Application*: While element selectors are useful, they apply styles to all instances of a specific HTML element on the page, which might not be desirable in some cases.
- *Lack of Specificity*: They lack the specificity of other selectors like class or ID selectors, making it harder to target specific elements in complex layouts.

Element selectors are foundational and provide a simple way to set styles for standard HTML elements. However, in more complex designs, combining them with other types of selectors may be necessary to achieve a more targeted and flexible styling approach.

Here's an example to illustrate how to apply CSS in HTML to change text color to red for all <p> elements

**Code Description:** CSS in HTML code to create an inline named mycss.HTML that changes text color of paragraph

```
<!DOCTYPE html>

<html>

<head>
<style>
p {
 color: red;
}
</style>
</head>

<body>
<p>Every paragraph will be affected.</p>
<p>Me too!</p>
<p>And me!</p>
</body>
</html>
```

**Output Description:** How the CSS code above looks in a browser

Every paragraph will be affected.

Me too!

And me!

## 2. ID Selector

An ID Selector is a type of selector that allows you to target and style a specific HTML element based on its unique identifier. The unique identifier is assigned using the id attribute in HTML. ID selectors are denoted by a hash (#) followed by the ID name.

### Syntax:

```
#idName {
 /* CSS properties and values */
}
```

#### Description:

1. # (Hash): The hash symbol is used to denote that this is an ID selector.
2. idName: This is the unique identifier assigned to an HTML element using the id attribute. It must match exactly the ID name you want to target.
3. CSS Properties and Values: Inside the curly braces, you define the styles you want to apply to the selected HTML element.

## Merits and demerits of the ID selector

### Merits:

- *Unique Identification:* IDs must be unique within an HTML document, ensuring that the styles are applied to a specific and singular element.
- *Specificity:* ID selectors have higher specificity compared to other types of selectors, making them powerful for targeting specific elements.

### Demerits:

- *Reusability:* Unlike class selectors, which can be applied to multiple elements, IDs are intended to be unique. This limits the reusability of styles across different elements.
- *Global Scope:* Styles applied with ID selectors have a global scope, affecting the entire document. This may lead to potential conflicts in larger projects.

The id selector selects the HTML element with a unique identifier (id) and adds CSS to it. Let's see an example,

```
#second-paragraph{
```

```
 color: red;
}
```

In the example above, each element is described in detail.

- (#) hash represents the id selector
- , the name of the id
- The id selector name is “second-paragraph”.
- 

The id selector #second-paragraph selects the second paragraph and styles the text color to red.



The id selector is unique and selects one unique element.

Here's an example to illustrate how to apply CSS to the HTML code to modify the text color of the first, second, and third <p> elements to red, green, and yellow, respectively.

**Code Description:** *CSS in HTML code to create an inline named mycss.HTML that changes the text color of paragraphs by ID selector.*

```
<!DOCTYPE html>

<html>

<head>

<style>

#paragraph1{color: red;}

#paragraph2{color: green;}

#paragraph3{color: yellow;}

</style>
```

```
</head>

<body>

<p id="paragraph1">Every paragraph will be affected.</p>

<p id="paragraph2">Me too!</p>

<p id="paragraph3">And me!</p>

</body>

</html>
```

**Output Description:** How the CSS code above looks in a browser

Every paragraph will be affected.

Me too!

And me!

### 3. Class Selector

In CSS, a Class Selector is a type of selector that allows you to target and style HTML elements with a specific class attribute. Class selectors are prefixed with a dot (.) followed by the class name. They provide a way to apply styles consistently to multiple elements that share the same class, regardless of their tag or position in the HTML document.

#### Syntax:

```
.className {
 /* CSS properties and values */
}
```

Description:

1. **.** (Dot): The dot preceding the class name indicates that this is a class selector.
2. **className:** This is the name of the class you want to target. It should match the class attribute in your HTML.

3. **CSS Properties and Values:** Inside the curly braces, you define the styles you want to apply to elements with the specified class.

Let's see an example,

```
.first-paragraph {
 background-color: orange;
}
```

In the example above, each element is described in detail.

- (.) dot represents the class selector
- The class selector name is “first-paragraph-name”.

The class selector .first-paragraph selects all the paragraphs having the first-paragraph class name and styles background-color to orange.

Here's an example to illustrate how to apply CSS to the HTML code to modify the text color of the first, second, and third <p> elements to red, green, and yellow, respectively.

**Code Description:** *CSS in HTML code to create an inline named mycss.HTML that changes the text color of paragraphs by class selector.*

```
<!DOCTYPE html>

<html>

<head>

<style>

.paragraph1{color: red;}

.paragraph2{color: green;}

.paragraph3{color: yellow;}

</style>

</head>

<body>

<p class="paragraph1">Every paragraph will be affected.</p>

<p class="paragraph2">Me too!</p>
```

```
<p class="paragraph3">And me!</p>
</body>
</html>
```

**Output Description:** How the CSS code above looks in a browser

Every paragraph will be affected.

Me too!

And me!

#### 4. Universal Selector

The universal selector selects every single HTML element on the page. It is written using the asterisk ( \* ) character.

Let's see an example,

```
* {
 color: red;
}
```

In the above example, the universal selector \* selects all the HTML elements and applies the red color.



The universal selector is also referred to as the wildcard selector.

Here's an example to illustrate how use CSS in HTML to change background color for a page:

**Code Description:** CSS in HTML code to create an inline named mycss.HTML that changes the background color of paragraphs.

```
<!DOCTYPE html>

<html>
 <head>
 <style>
 * {
 background-color: yellow;
 }
 </style>
 </head>
 <body>
 <h1>Demo of the * selector</h1>
 <div class="intro">
 <p id="firstname">My name is Donald.</p>
 <p id="hometown">I live in Duckburg.</p>
 </div>
 <p>My best friend is Mickey.</p>
 </body>
</html>
```

**Output Description:** How the CSS code above looks in a browser

## Demo of the \* selector

My name is Donald.

I live in Duckburg.

My best friend is Mickey.

## 5. #group-selector Group Selector

The group selector allows you to select multiple elements and apply the same style to all of them.

Let's see an example,

```
h1, p {
 color: blue;
}
```

Here, the code applies CSS styling to all `<h1>` and `<p>` elements. Notice that we have used `,` to separate the HTML elements.

Here's an example to illustrate how the CSS group selector works:

**Code Description:** *CSS in HTML code to apply group selector*

```
<!DOCTYPE html>

<html>

<head>

<style>

p,h2{color: red;}

</style>

</head>

<body>

<p>Every paragraph will be affected.</p>
```

```
<p>Me too!</p>

<h2>And me!</h2>

</body>

</html>
```

**Output Description:** How the CSS code above looks in a browser

Every paragraph will be affected.

Me too!

And me!

## 6. Attribute Selector

In CSS, an Attribute Selector allows you to target and style HTML elements based on the presence or value of their attributes. Attribute selectors provide a flexible way to select elements based on various conditions related to their attributes. The syntax for attribute selectors includes the element name followed by square brackets containing the attribute and, optionally, a specific value or value condition.

### Syntax:

```
element[attribute="value"] {
 /* CSS properties and values */
}
```

### Description:

1. element: This is the HTML element you want to target.
2. [attribute="value"]: Inside the square brackets, you specify the attribute and, optionally, the attribute value or condition.
3. CSS Properties and Values: Inside the curly braces, you define the styles you want to apply to the selected elements based on the attribute condition.

### Attribute Selector Types:

- Presence Selector ([attribute]): Selects elements that have the specified attribute, regardless of its value.

```
a[target] {
 color: #e74c3c;
}
```

- Exact Value Selector ([attribute="value"]): Selects elements with the exact specified attribute value.

```
input[type="checkbox"] {
 margin-right: 5px;
}
```

- Substring Value Selector ([attribute\*="value"]): Selects elements with an attribute value containing the specified substring.

```
[class*="btn"] {
 background-color: #3498db;
 color: #fff;}
```

The attribute selector selects elements based on specific attribute values. Let's see an example,

```
p[class] {
 background-color: orange;
}
p[class="third"] {
 color: blue;
}
```

In the above example, the attribute selector, p[class] selects all p elements having the class attribute and styles their background color to red. p[class="third"] selects all p elements with the .third class name and styles their color to blue.



This selector only selects an element if a specified given attribute exists.

Here's an example to illustrate how the CSS Attribute selector works:

**Code Description:** CSS embedded in HTML code to apply an attribute selector

```
<!DOCTYPE html>

<html>

<head>

<style>

a[target] {

background-color: yellow;

}

</style>

</head>

<body>

<h2>CSS [attribute] Selector</h2>

<p>The links with a target attribute gets a yellow background:</p>

w3schools.com

disney.com

wikipedia.org

</body>

</html>
```

**Output Description:** How the CSS code above looks in a browser

## CSS [attribute] Selector

The links with a target attribute gets a yellow background:

[w3schools.com](https://www.w3schools.com) [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)

## 7. Descendant Selector:

A Descendant Selector is a type of selector that allows you to target and style an HTML element that is a descendant of another specified element. Descendant selectors are used to apply styles to a particular element only when it is nested within another specific element in the HTML document. The selector consists of the ancestor element followed by a space and the descendant element.

### Syntax:

```
ancestor descendant {
 /* CSS properties and values */
}
```

#### Description:

- Ancestor:** This is the ancestor element, and it represents the parent element in the HTML structure.
- Descendant:** This is the descendant element, representing the child element nested inside the ancestor.
- CSS Properties and Values:** Inside the curly braces, you define the styles you want to apply to the descendant element when it is a descendant of the specified ancestor.

Here's an example to illustrate how utilize the HTML and CSS descendant selector:

| HTML code                                                                                                                                                                           | CSS code:                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <pre>&lt;nav&gt;<br/>    &lt;ul&gt;<br/>        &lt;li&gt;Home&lt;/li&gt;<br/>        &lt;li&gt;About&lt;/li&gt;<br/>        &lt;li&gt;Contact&lt;/li&gt;<br/>    &lt;/ul&gt;</pre> | <pre>nav ul {<br/>    list-style-type: none;<br/>    margin: 0;<br/>    padding: 0;<br/>}</pre> |

|        |  |
|--------|--|
| </nav> |  |
|--------|--|

In the above example The styles specified in the descendant selector will be applied to all `<ul>` elements that are descendants of a `<nav>` element. In this example, it removes the default list styling by setting `list-style-type` to `none` and resets margin and padding.

### **Merits and demerits of the Descendant selector**

Merits:

- Targeted Styling: Descendant selectors provide a way to style specific elements based on their position within the HTML document.
- Structured Styling: They are useful for maintaining a structured and organized approach to styling, especially in navigation menus and nested structures.

Demerits:

- Specificity Concerns: Descendant selectors have a lower specificity compared to other selectors like IDs or classes, which may lead to unintended styling conflicts in more complex layouts.
- Performance Considerations: Overuse of descendant selectors with deeply nested structures may impact performance, especially in larger documents.

Descendant selectors are valuable for styling nested elements in a structured way. When used thoughtfully, they contribute to a modular and maintainable CSS codebase. However, it's essential to balance their use with considerations of specificity and performance in more extensive projects.

## **8. Child Selector**

Child Selector is a type of selector that allows you to target and style an HTML element that is a direct child of another specified element. Child selectors are used to apply styles to an element only when it is an immediate child of a specific parent element. The selector consists of the parent element followed by the greater-than symbol (`>`), and then the child element.

### **Syntax:**

```
parent > child {
 /* CSS properties and values */
}
```

**Description:**

1. **parent:** This is the parent element, and it represents the direct parent in the HTML structure.
2. **> (Greater-than Symbol):** It is the child combinator, specifying that the styles should apply only to direct children.
3. **child:** This is the child element, representing the immediate child nested inside the parent.
4. **CSS Properties and Values:** Inside the curly braces, you define the styles you want to apply to the child element when it is a direct child of the specified parent.

Here's an example to illustrate how utilize CSS child selector in the HTML:

| <b>HTML code</b>                                                                                                                                                                                 | <b>CSS code:</b>                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <pre>&lt;article&gt;   &lt;p&gt;This paragraph has italicized text.&lt;/p&gt; &lt;div&gt;   &lt;p&gt;This paragraph does not have italicized text.&lt;/p&gt; &lt;/div&gt; &lt;/article&gt;</pre> | <pre>article &gt; p {   font-style: italic; }</pre> |

In the above example, the styles specified in the child selector will be applied to all `<p>` elements that are direct children of an `<article>` element. In this example, it makes the text inside these paragraphs italic.

**Merits and demerits of the Child selector**

**Merits:**

- **Precise Targeting:** Child selectors allow you to target elements with a specific parent, providing precise control over styling.

- Avoiding Indirect Descendants: By specifying the direct child relationship, you can avoid styling elements that are nested deeper in the hierarchy.

Demerits:

- Compatibility: Child selectors are supported in modern browsers, but older browsers may not fully support them.
- Specificity: While more specific than descendant selectors, child selectors have lower specificity compared to IDs or classes.

Child selectors are particularly useful when you want to apply styles exclusively to immediate children of a specific parent. They contribute to maintaining a clean and modular CSS structure, especially in scenarios where you need to style specific elements in a parent-child relationship without affecting other nested elements.

## 9. Precedence of CSS selectors

The browser will also decide the priority based on CSS selectors (eg: #id, .class, and Tag name). The below image shows the priority among the selectors:

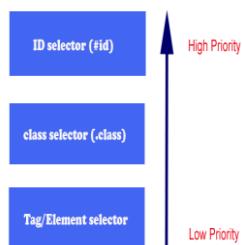


Figure 1.2: Precedence of CSS selectors

**Dear Students!** It is now your opportunity to provide answers to the following self-check questions.



### Self-check 1-2:

1. Explain the importance of proper indentation in CSS code.

- 
2. Explain the purpose of the semicolon (;) in CSS rules. Answer: The semicolon separates individual CSS declarations within a rule. Forgetting to include semicolons or using them incorrectly can result in CSS syntax errors.
  3. What is a common error when writing CSS rules within a media query?
  4. What is the purpose of the colon (:) in CSS syntax, and where should it be used?
  5. What is the correct way to specify multiple CSS selectors for a rule? sans-serif; }.
  6. Explain a frequent error associated with CSS property values.
  7. Describe a common mistake when using units in CSS values.
  8. What is the correct way to specify multiple CSS selectors for a rule?
- 

## 1.4. CSS Comments

CSS comments are used to add explanatory notes or disable certain styles without affecting the functionality of the stylesheet. Comments are not rendered in the browser and are meant for developers' reference only. There are two ways to write CSS comments:

- **Single-line Comments:** Single-line comments start with /\* and end with \*/.  
For example:  
/\* This is a single-line comment \*/
- **Multi-line Comments:** Multi-line comments begin with /\* and end with \*/, allowing for comments spanning multiple lines.

For example:

/\*

This is a multi-line comment.

It can span across multiple lines.

\*/

Comments are useful for documenting code, explaining style choices, or temporarily disabling styles during development and debugging.

To sum up, CSS is an essential language for web development that gives designers the ability to produce aesthetically pleasing and coherent websites. Developers are able to properly style

HTML components by knowing the syntax, inclusion techniques, selectors, and comments in CSS.

## 1.5. Method to insert style sheets in HTML document

There are three most common ways of inserting/ adding a style sheet:-

1. External CSS
2. Internal CSS
3. Inline CSS

**Dear Students!** Let's see through each of these methods in more details and discuss their ideal use case

### 1.5.1. Inline Style

Inline CSS involves applying CSS directly within HTML elements using the style attribute. This method is suitable for applying unique styles to individual elements.

#### Syntax:

```
<element style="CSS property: value">
```

Here's an example to illustrate how insert Inline CSS

#### Code Description: HTML embedded with CSS code to use Inline CSS

```
<!DOCTYPE html>

<html>

<head>

 <title>Inline CSS Example</title>

</head>

<body>
```

```
<h1 style="color: red; font-size: 24px;">Heading with Inline CSS </h1>

<p style="color: blue; font-size: 16px;">Paragraph with Inline CSS</p>

<div style="background-color: yellow; padding: 10px;">
 <p>Div with Inline CSS</p>
</div>

</body>

</html>
```

**Output Description:** How the HTML code above looks in a browser

## Heading with Inline CSS

Paragraph with Inline CSS

Div with Inline CSS

In the above example, CSS styles such as color, font-size, and background-color are directly applied to HTML elements using the *style* attribute.

### 1.5.2. Internal Style sheets

Internal CSS involves placing CSS code within the `<style>` element in the `<head>` section of the HTML document. This method is useful for applying styles to specific pages.

Here's an example to illustrate how insert Internal CSS

**Code Description:** HTML embedded with CSS code to use Internal CSS

```
<!DOCTYPE html>

<html>
 <head>
 <title>Internal CSS Example</title>
 <style>
 h1 {
 color: red;
 font-size: 24px;
 }
 p {
 color: blue;
 font-size: 16px;
 }
 div {
 background-color: yellow;
 padding: 10px;
 }
 </style>
 </head>
 <body>
 <h1>Heading with Internal CSS</h1>
 <p>Paragraph with Internal CSS</p>
 <div>
```

```
<p>Div with Internal CSS</p>
</div>
</body>
</html>
```

**Output Description:** How the HTML code above looks in a browser

## Heading with Inline CSS

### Paragraph with Inline CSS

Div with Inline CSS



CSS styles are defined within the `<style>` element in the `<head>` section. These styles apply to HTML elements throughout the document.

### 1.5.3. External Stylesheet

External stylesheets are created in separate CSS files and linked to HTML documents using the `<link>` tag. This method involves creating a separate CSS file (e.g., `styles.css`) and linking it to the HTML document using the `<link>` element in the `<head>` section.

#### Syntax:

```
<link rel="" type="text/css" href="">
```



## Operation Sheet : 1-1

### Title: Creating an HTML document with an external stylesheet

**Objective:** This operation sheet provides step-by-step instructions on how to insert external css

**Instructions:** Write the below codes step-by-step guide to creating an HTML document with an external stylesheet

#### Step 1: Create the HTML File

1. Open a text editor (e.g., Notepad, VS Code, Sublime Text).
2. Create a new file and save it as **index.html**.
3. Type or Copy the below code and press ctrl + S

```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

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

 <title>External CSS Example</title>

 <link rel="stylesheet" href="styles.css">

</head>

<body>

 <h1>Heading with External CSS</h1>

 <p>Paragraph with External CSS</p>

 <div>
```

```
<p>This is a div element styled with external CSS.</p>

</div>

<section>

 <p>This is a section element styled with external CSS.</p>

</section>

</body>

</html>
```

**Step 2:** Create the CSS File

1. Create a new file in the same directory as `index.html` and save it as `styles.css`.

Add the following CSS code:

```
/* styles.css */
```

```
/* Styling for the h1 element */
```

```
h1 {

 color: blue;

 font-family: Arial, sans-serif;

 text-align: center;

}
```

```
/* Styling for the p element */
```

```
p {
```

```
color: green;

font-size: 14px;

line-height: 1.5;

}

/* Styling for the div element */

div {

border: 2px solid black;

padding: 10px;

margin: 10px;

background-color: #f0f0f0;

}

/* Styling for the section element */

section {

border: 1px dashed red;

padding: 20px;

margin: 20px;

background-color: #e0e0e0;

}
```

**Step 3:** Link the CSS File to the HTML File

Make sure the link to the stylesheet in your HTML file is correct:

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

**Step 4:** Open the HTML File in a Browser

1. Open the directory where you saved index.html and styles.css.
2. Double-click on index.html to open it in your web browser.

You should see the HTML content styled according to the rules defined in *styles.css*. The heading will be blue and centered, the paragraphs will be green, and the div and section elements will have their respective borders, padding, margins, and background colors.

Here's an example to illustrate how to insert external CSS.

| HTML file (index.html)                                                                                                                                                                                                                                                                                                                                                                                                                  | CSS file (styles.css)                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;!DOCTYPE html&gt;  &lt;html&gt;  &lt;head&gt;      &lt;title&gt;External CSS Example&lt;/title&gt;      &lt;link rel="stylesheet" type="text/css"         href="styles.css"&gt;  &lt;/head&gt;  &lt;body&gt;      &lt;h1&gt;Heading with External CSS&lt;/h1&gt;      &lt;p&gt;Paragraph with External CSS&lt;/p&gt;      &lt;div&gt;          &lt;p&gt;Div with External CSS&lt;/p&gt;      &lt;/div&gt;  &lt;/body&gt;</pre> | <pre>h1 {      color: red;      font-size: 24px;  }  p {      color: blue;      font-size: 16px;  }  div {      background-color: yellow;      padding: 10px;  }</pre> |

```
</body>

</html>
```

**Output Description:** How the HTML code above looks in a browser

## Heading with External CSS

### Paragraph with External CSS

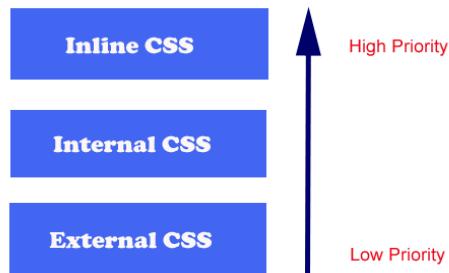
### Div with External CSS

In this example, the CSS styles are defined in the **styles.css** file, and the HTML document (**index.html**) links to this external stylesheet using the **<link>** element. This allows for the separation of concerns and easier maintenance of styles across multiple pages.

#### 1.5.4. Precedence of CSS method /Cascading Order

**Dear Students!** What style will be used when there is more than one style specified for an HTML element?

The browser will decide the priority based on three types of CSS (inline, internal, and external). The below image shows the priority among these three types:



*Figure 1.2: Precedence of CSS method*

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

**Dear Students!** *It is now your opportunity to provide answers to the following self-check questions.*

**Self-check 1-3:**

1. Which method of CSS involves writing the CSS code in a separate file?
2. Where is Internal CSS typically placed within an HTML document?
3. Which CSS method allows for styling individual HTML elements directly within their tags?
4. Which CSS method is best suited for making global styling changes across multiple web pages?
5. What is one advantage of using Internal CSS?
6. Which CSS method is recommended for applying temporary styling changes or quick adjustments?
7. Which CSS method typically leads to better code organization and maintenance?
8. What attribute is used to apply Inline CSS to HTML elements?
9. Which CSS method allows for the quickest styling changes to a specific page?
10. Which CSS method is commonly used for applying unique styles to individual elements?
11. When is External CSS preferred over Internal CSS?
12. How can Inline CSS affect the readability of HTML code?
13. What is the file extension commonly used for External CSS files?
14. Which CSS method allows for the separation of content and presentation?
15. Which CSS method is applied directly to individual HTML elements?

## 1.6. CSS Properties

CSS (Cascading Style Sheets) properties are rules used in conjunction with HTML elements to control the visual presentation of web pages. They define how HTML elements should appear on the screen or in other media. CSS properties encompass a wide range of styling options, including but not limited to:

- **Typography:** Font family, font size, font weight, font style, text alignment, text decoration, line height, letter spacing, text transform, etc.
- **Colors and Backgrounds:** Text color, background color, background image, background position, background repeat, etc.
- **Layout and Box Model:** Width, height, margin, padding, border, display type, position, float, clear, etc.
- **Visual Effects:** Box shadow, border radius, opacity, transitions, transforms, etc.
- **Flexbox and Grid Layout:** Properties for creating flexible and grid-based layouts.
- **Media Queries:** Properties for defining styles based on the characteristics of the device displaying the web page, such as screen size, resolution, etc.

CSS properties are applied to HTML elements using selectors, which target specific elements or groups of elements, and values, which define how the selected elements should be styled.

### 1.6.1. Styling text and fonts

#### 1.6.1.1. Text Properties

Table 1-1: Common text properties in CSS

| CSS Property | Description                           | Example                          |
|--------------|---------------------------------------|----------------------------------|
| color        | Sets the color of the text.           | <code>color: #FF0000;</code>     |
| text-align   | Aligns the text within its container. | <code>text-align: center;</code> |
| line-height  | Sets the height of a line of text.    | <code>line-height: 1.5;</code>   |

| CSS Property    | Description                                           | Example                           |
|-----------------|-------------------------------------------------------|-----------------------------------|
| letter-spacing  | Sets the spacing between characters.                  | letter-spacing: 1px;              |
| text-transform  | Controls the capitalization of text.                  | text-transform: uppercase;        |
| text-decoration | Sets the decoration of the text.                      | text-decoration: underline;       |
| word-spacing    | Sets the spacing between words.                       | word-spacing: 2px;                |
| white-space     | Defines how white space inside an element is handled. | white-space: nowrap;              |
| text-shadow     | Adds shadow to text.                                  | text-shadow: 2px 2px 4px #000000; |

### 1.6.1.2. Font Properties:

Table 1-2: Summarizing common font properties in CSS

| CSS Property | Description                                                       | Example                                   |
|--------------|-------------------------------------------------------------------|-------------------------------------------|
| font-family  | Specifies the font family for text.                               | body { font-family: Arial, sans-serif; }  |
| font-size    | Sets the size of the font.                                        | h1 { font-size: 24px; }                   |
| font-weight  | Sets the thickness of the font.                                   | strong { font-weight: bold; }             |
| font-style   | Sets the style of the font.                                       | em { font-style: italic; }                |
| font-variant | Controls the usage of small caps for lowercase letters.           | .small-caps { font-variant: small-caps; } |
| font-stretch | Selects a normal, condensed, or expanded face from a font family. | p { font-stretch: condensed; }            |

| CSS Property    | Description                          | Example                                   |
|-----------------|--------------------------------------|-------------------------------------------|
| line-height     | Sets the height of a line of text.   | p { line-height: 1.5; }                   |
| letter-spacing  | Sets the spacing between characters. | h1 { letter-spacing: 1px; }               |
| text-transform  | Controls the capitalization of text. | .uppercase { text-transform: uppercase; } |
| text-decoration | Sets the decoration of the text.     | a { text-decoration: none; }              |

**Dear Students!** Let's create a simple HTML file to illustrate the CSS text and fonts exercise along with the answer code provided.



## Operation Sheet : 1-2

**Title:** Extensive practice with CSS text and font style

**Objective:** This operation sheet provides step-by-step instructions on how to apply style text and font properties

**Instruction:** Enhance your CSS Property skills with this exercise that delves into text and fonts. Check out the answer code provided!

**Task:** Create HTML File and CSS file named (index.html) and styles.css and define the following styles:

1. Set the font family of the entire document to Arial, sans-serif.
2. Make all **<h2>** elements uppercase and italic.
3. Set the font size of all paragraphs (**<p>** elements) to 16 pixels.

4. Set the line height of all paragraphs to 1.5 times their font size.
5. Make all **a** elements bold and change their text color to blue. Remove the default underline decoration.
6. Add a 1-pixel solid border around all **blockquote** elements with a gray color.
7. Change the text color of all **blockquote** elements to dark green.

### Solution

1. HTML File (index.html):

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Text and Fonts Exercise</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h2>This is a Heading</h2>

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

<blockquote>This is a blockquote..</blockquote>

<!-- This is a link -->

</body>

</html>
```

The above HTML file includes a heading (**<h2>**), a paragraph (**<p>**), a blockquote (**<blockquote>**), and a link (**<a>**). When you open this HTML file in a web browser, the styles defined in the **styles.css** file will be applied to these elements according to the exercise instructions. You'll see the text styled as per the specifications mentioned in the exercise.

2. Here's the CSS file(**Style.css**)

```
/* 1. Set the font family of the entire document to Arial, sans-serif. */

body {

 font-family: Arial, sans-serif;

}

/* 2. Make all <h2> elements uppercase and italic. */

h2 {

 text-transform: uppercase;

 font-style: italic;

}

/* 3. Set the font size of all paragraphs (<p> elements) to 16 pixels. */

p {

 font-size: 16px;

}

/* 4. Set the line height of all paragraphs to 1.5 times their font size. */

p {

 line-height: 1.5;

}

/* 5. Make all <a> elements bold and change their text color to blue. Remove the default underline decoration. */

a {

 font-weight: bold;

 color: blue;

 text-decoration: none;
```

```

}

/* 6. Add a 1-pixel solid border around all <blockquote> elements with a gray color. */

blockquote {

 border: 1px solid gray;

}

/* 7. Change the text color of all <blockquote> elements to dark green. */

blockquote {

 color: darkgreen;

}

```

Make sure this link tag added or not in HTML file:<link rel="stylesheet" href="styles.css">

4. Open the above HTML file (index.html) in any browser.

## 1.6.2. CSS colors and backgrounds

Here is a brief explanation of CSS colors and backgrounds along with examples in a table:

| Property         | Description                                                         | Example                                 |
|------------------|---------------------------------------------------------------------|-----------------------------------------|
| color            | Sets the text color of an element.                                  | color: blue;                            |
| background-color | Sets the background color of an element.                            | background-color: lightgray;            |
| opacity          | Sets the transparency level of an element.                          | opacity: 0.5;                           |
| rgba()           | Defines a color using red, green, blue, and alpha (opacity) values. | background-color: rgba(255, 0, 0, 0.5); |
| hex              | Defines a color using hexadecimal notation.                         | color: #00FF00;                         |
| hsl()            | Defines a color using hue, saturation, and lightness.               | background-color: hsl(120, 100%, 50%);  |

| Property              | Description                                                                                      | Example                               |
|-----------------------|--------------------------------------------------------------------------------------------------|---------------------------------------|
| background-image      | Sets one or more background images for an element.                                               | background-image: url('example.jpg'); |
| background-repeat     | Sets how a background image is repeated.                                                         | background-repeat: repeat-x;          |
| background-position   | Sets the starting position of a background image.                                                | background-position: center top;      |
| background-size       | Sets the size of the background image.                                                           | background-size: cover;               |
| background-attachment | Sets whether a background image scrolls with the content or remains fixed.                       | background-attachment: fixed;         |
| background-blend-mode | Specifies how the background image or color should blend with the element's content.             | background-blend-mode: multiply;      |
| background-origin     | Sets where the background image or color starts from within the padding, border, or content box. | background-origin: padding-box;       |
| background-clip       | Specifies the painting area of the background.                                                   | background-clip: content-box;         |

**Dear Students!** Let's create a simple CSS file to illustrate the CSS colors and Backgrounds exercise along with the answer code provided



### Operation Sheet : 1-3

**Title:** Extensive practice with CSS colors and backgrounds

**Objective:** This operation sheet provides step-by-step instructions on how to apply Properties colors

and backgrounds

**Instruction:** let's create an exercise to practice CSS colors and backgrounds along with the answer code.

**Task:** Suppose an HTML file is formed. Create the following styles in a CSS file called styles.css.

1. Set the background color of the entire document to light gray.
2. Set the text color of all paragraphs (<p> elements) to dark blue.
3. Set the background color of all <div> elements to light yellow.
4. Set the text color of all <h1> elements to red.
5. Set the background color of the header (<header>) to light green.
6. Set the text color of links (<a> elements) to purple and remove the default underline decoration.
7. Set the background color of all <section> elements to light cyan

**Solution: CSS file(Style.css)**

```
/* 1. Set the background color of the entire document to light gray. */
```

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

```
/* 2. Set the text color of all paragraphs (<p> elements) to dark blue. */
```

```
p {
 color: darkblue;
}
```

```
/* 3. Set the background color of all <div> elements to light yellow. */
```

```
div {
```

```
background-color: lightyellow;
}

/* 4. Set the text color of all <h1> elements to red. */

h1 {
 color: red;
}

/* 5. Set the background color of the header (<header>) to light green. */

header {
 background-color: lightgreen;
}

/* 6. Set the text color of links (<a> elements) to purple and remove the default underline decoration. */

a {
 color: purple;
 text-decoration: none;
}

/* 7. Set the background color of all <section> elements to light cyan. */

section {
 background-color: lightcyan;
}
```

### 1.6.3. Margin and Padding:

CSS properties for margin and padding, along with examples:

| Property       | Description                                                  | Example               |
|----------------|--------------------------------------------------------------|-----------------------|
| margin         | Sets the margin space around an element.                     | margin: 10px;         |
| margin-top     | Sets the margin space above an element.                      | margin-top: 20px;     |
| margin-right   | Sets the margin space to the right of an element.            | margin-right: 15px;   |
| margin-bottom  | Sets the margin space below an element.                      | margin-bottom: 20px;  |
| margin-left    | Sets the margin space to the left of an element.             | margin-left: 15px;    |
| padding        | Sets the padding space inside an element's boundary.         | padding: 10px;        |
| padding-top    | Sets the padding space above an element's content.           | padding-top: 15px;    |
| padding-right  | Sets the padding space to the right of an element's content. | padding-right: 20px;  |
| padding-bottom | Sets the padding space below an element's content.           | padding-bottom: 15px; |
| padding-left   | Sets the padding space to the left of an element's content.  | padding-left: 20px;   |

CSS properties related to borders, width, and height, along with examples:

| Property     | Description                                              | Example                  |
|--------------|----------------------------------------------------------|--------------------------|
| border       | Sets the width, style, and color of an element's border. | border: 1px solid black; |
| border-width | Sets the width of an element's border.                   | border-width: 2px;       |

| Property      | Description                                                  | Example               |
|---------------|--------------------------------------------------------------|-----------------------|
| border-style  | Sets the style of an element's border (e.g., solid, dashed). | border-style: dotted; |
| border-color  | Sets the color of an element's border.                       | border-color: red;    |
| border-radius | Sets the curvature of an element's border corners.           | border-radius: 5px;   |
| width         | Sets the width of an element.                                | width: 200px;         |
| min-width     | Sets the minimum width of an element.                        | min-width: 100px;     |
| max-width     | Sets the maximum width of an element.                        | max-width: 500px;     |
| height        | Sets the height of an element.                               | height: 150px;        |
| min-height    | Sets the minimum height of an element.                       | min-height: 50px;     |
| max-height    | Sets the maximum height of an element.                       | max-height: 300px;    |

**Dear Students!** Let's create a simple CSS file to illustrate the CSS colors and Backgrounds exercise along with the answer code provided.



### Operation Sheet : 1-4

**Title:** Extensive practice with borders, width, height, margin, and padding

**Objective:** This operation sheet provides step-by-step instructions on how to apply Properties colors and backgrounds

**Instruction!** Let's create an exercise to practice CSS properties related to borders, width, height, margin, and padding:

**Task:** Create a CSS file named styles.css and define the following styles:

1. Set the border of all <div> elements to a solid 2-pixel border with a color of dark gray.
2. Set the width of all <img> elements to 100 pixels.
3. Add a margin of 20 pixels to the top and bottom of all <h1> elements and 10 pixels to the left and right.
4. Add a padding of 15 pixels to all sides of all <p> elements.

**Solution: CSS file(Style.css)**

```
/* 1. Set the border of all <div> elements to a solid 2-pixel border with a color of dark gray. */
div {
 border: 2px solid darkgray;
}

/* 2. Set the width of all elements to 100 pixels. */
img {
 width: 100px;
}

/* 4. Add a margin of 20 pixels to the top and bottom of all <h1> elements and 10 pixels to the left and right. */
h1 {
 margin: 20px 10px;
}

/* 5. Add a padding of 15 pixels to all sides of all <p> elements. */
```

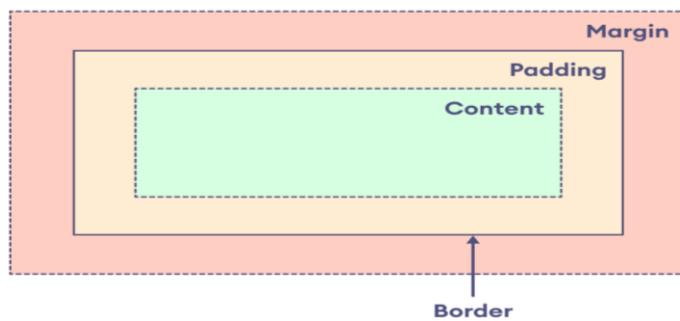
```
p {
 padding: 15px;
}
```

The above exercise covers CSS properties related to borders, width, height, margin, and padding. You can link this CSS file in your HTML file to see the styles applied.

#### 1.6.4. CSS box model

The CSS Box Model is a fundamental concept in web design that describes the structure of an HTML element. It consists of four main components:

- I. **Content:** The actual content of the element, such as text, images, or other HTML elements.
- II. **Padding:** The space between the content and the element's border. Padding helps to create space inside the element and separates the content from the border.
- III. **Border:** The border surrounds the padding and content of the element. It can have different styles (solid, dashed, etc.), widths, and colors, and it acts as a boundary between the padding and the margin.
- IV. **Margin:** The space outside the border of the element. Margins create space between the element's border and other elements on the page, providing separation between elements.



*Figure 1.3: CSS Box Model*

The different CSS box model attributes are listed in detail in the table below, along with some helpful examples.

| Property       | Description                                                  | Example                  |
|----------------|--------------------------------------------------------------|--------------------------|
| width          | Sets the width of an element's content area.                 | width: 300px;            |
| height         | Sets the height of an element's content area.                | height: 200px;           |
| padding        | Sets the padding area around an element's content.           | padding: 20px;           |
| padding-top    | Sets the padding area above an element's content.            | padding-top: 10px;       |
| padding-right  | Sets the padding area to the right of an element's content.  | padding-right: 15px;     |
| padding-bottom | Sets the padding area below an element's content.            | padding-bottom: 10px;    |
| padding-left   | Sets the padding area to the left of an element's content.   | padding-left: 15px;      |
| border         | Sets the width, style, and color of an element's border.     | border: 1px solid black; |
| border-width   | Sets the width of an element's border.                       | border-width: 2px;       |
| border-style   | Sets the style of an element's border (e.g., solid, dashed). | border-style: dotted;    |
| border-color   | Sets the color of an element's border.                       | border-color: red;       |
| margin         | Sets the margin area around an element.                      | margin: 10px;            |
| margin-top     | Sets the margin area above an element.                       | margin-top: 5px;         |
| margin-right   | Sets the margin area to the right of an element.             | margin-right: 5px;       |
| margin-bottom  | Sets the margin area below an element.                       | margin-bottom: 5px;      |
| margin-left    | Sets the margin area to the left of an element.              | margin-left: 5px;        |

**Dear Students!** Let's expand on that explanation by demonstrating how to apply specific top, bottom, left, and right margins, borders, and padding to an element using CSS. We'll also visualize how the content fits inside the padding, border, and margin. By practicing with these exercises, you can better understand how the CSS box model works and how to control the layout and spacing of elements on a webpage.

**Instruction!** Let's create a practical exercise to help you practice and understand the CSS box model.



## Operation Sheet : 1-5

### Title: Extensive practice CSS box model

**Objective:** This exercise will involve creating a simple CSS box model, where you can manipulate margins, borders, and padding to see how they affect the layout.

**Instruction:** Let's engage in a hands-on exercise that will aid you in practicing and comprehending the CSS box model. This exercise will involve the development of a simple webpage that incorporates numerous boxes. By experimenting with the margins, borders, and padding, you will be able to observe the direct impact these attributes have on the overall layout and design of the webpage.

### Project 1:Simple CSS Box Model Practice

**Task1:** Create a HTML file named box.html and write the following code:

```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">
```

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

<title>Detailed CSS Box Model Example</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

 <div class="box">This is the content area</div>

</body>

</html>
```

**Task2:** Create a CSS file named styles.css and Write the following script:

```
/* CSS for the detailed box model example */
```

```
body {

 display: flex;
 justify-content: center;
 align-items: center;
 height: 100vh;
 margin: 0;
 background-color: #f0f0f0;
}
```

```
.box {

 width: 200px;
 height: 100px;
 background-color: lightblue;
```

```
padding-top: 10px; /* Padding inside the element */

padding-right: 20px;

padding-bottom: 30px;

padding-left: 40px;

border-top: 5px solid darkblue; /* Border around the element */

border-right: 10px solid darkgreen;

border-bottom: 15px solid darkred;

border-left: 20px solid darkorange;

margin-top: 10px; /* Margin outside the element */

margin-right: 20px;

margin-bottom: 30px;

margin-left: 40px;

color: black;

font-family: Arial, sans-serif;

text-align: center;

}
```

**Task3:-** In order to improve the appearance and functionality of the css task mentioned above, it is necessary to make adjustments, changes, modifications, and additions to the CSS styles.

1. **Adjust Margins:** Change the margin values and observe how the space around the box changes. For example, try setting **margin-top: 50px;**
2. **Change Borders:** Experiment with different border styles such as **dashed**, **dotted**, and different colors.
3. **Modify Padding:** Change the padding values and observe how the space inside the border changes. For example, try setting **padding-left: 60px;**
4. **Add Another Box:** Add another <div> with the same class **box** to see how the margin affects the spacing between the two boxes. For example:

```
<div class="box">This is the first box</div>

<div class="box">This is the second box</div>
```

5. **Use Box-Sizing:** Apply **box-sizing: border-box;** to see how it changes the calculation of the element's total width and height by including padding and border in the element's dimensions.
- ```
.box { box-sizing: border-box; }
```

Project 2: Simple to multiple Box Model Practice

Activity1: Create a simple webpage with multiple boxes.

Activity2: Manipulate margins, borders, and padding to understand their effects on layout.

Activity3: Adjust, change, modify, and add HTML and CSS styles to understand their effects on layout.

Steps: Setup Your HTML and CSS Files

1. **HTML File (index.html):** Create an **index.html** file with the following basic structure:

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>CSS Box Model Practice</title> <link rel="stylesheet" href="styles.css"> </head> <body> <div class="box box1">Box 1</div> <div class="box box2">Box 2</div> <div class="box box3">Box 3</div> <div class="box box4">Box 4</div> </body> </html>
```

2. **CSS File (styles.css):** Create a **styles.css** file with the following styles:

```
/* Basic body styles for alignment */ body { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; margin: 0; background-color: #f0f0f0; } /* Basic box styles */ .box { width: 200px; height: 100px; margin: 20px; padding: 20px; border: 5px solid black; background-color: lightblue; color: black; font-family: Arial, sans-serif; text-align: center; line-height: 60px; /* Centers text vertically */ } .box1 { margin-top: 30px; border-top: 10px solid red; padding-left: 30px; } .box2 { margin-bottom: 40px; border-right: 10px dashed green; padding-right: 40px; } .box3 { margin-left: 50px; border-bottom: 10px dotted blue; padding-bottom: 50px; } .box4 { width: 250px; height: 150px; margin: 25px; padding: 25px; border: 15px solid navy; background-color: lightgreen; color: darkblue; text-align: center; line-height: 100px; /* Centers text vertically */ }
```

Tasks:

1. Adjust the Margin Values:

- Adjust the **margin-top** of **.box1** to **50px**.
- Adjust the **margin-bottom** of **.box2** to **60px**.
- Adjust the **margin-left** of **.box3** to **70px**.

Code

```
.box1 { margin-top: 50px; /* Adjusted */ border-top: 10px solid red; padding-left: 30px; }  
.box2 { margin-bottom: 60px; /* Adjusted */ border-right: 10px dashed green; padding-right: 40px; } .box3 { margin-left: 70px; /* Adjusted */ border-bottom: 10px dotted blue; padding-bottom: 50px; }
```

2. Change the Border Properties:

- Change the border style of **.box1** to **solid** and color to **purple**.
- Change the border style of **.box2** to **double** and color to **orange**.
- Change the border style of **.box3** to **groove** and color to **pink**.

code

```
.box1 { margin-top: 50px; border-top: 10px solid purple; /* Changed */ padding-left: 30px; }  
.box2 { margin-bottom: 60px; border-right: 10px double orange; /* Changed */ padding-right: 40px; } .box3 { margin-left: 70px; border-bottom: 10px groove pink; /* Changed */ padding-bottom: 50px; }
```

3. Modify the Padding Values:

- Modify the **padding-left** of **.box1** to **50px**.
- Modify the **padding-right** of **.box2** to **60px**.
- Modify the **padding-bottom** of **.box3** to **70px**.

code

```
.box1 { margin-top: 50px; border-top: 10px solid purple; padding-left: 50px; /* Modified */ }  
.box2 { margin-bottom: 60px; border-right: 10px double orange; padding-right: 60px; /* Modified */ } .box3 { margin-left: 70px; border-bottom: 10px groove pink; padding-bottom: 70px; /* Modified */ }
```

4. Add Another Box: Add another **<div>** with a class **box4** to your HTML file and style it differently.

Code

```
<div class="box box4">Box 4</div>
```

The styles for **.box4** were already included in the initial CSS setup

1.7. CSS positioning and layout Properties

Dear Learner! you'll gain practical understanding of how to position and layout elements on a webpage effectively. Feel free to adjust values and experiment further to see how the layout changes.

CSS positioning and layout properties are essential for creating structured and responsive web designs. The main properties include position, display, float, clear, flexbox, and grid.

Overview of Properties

1. Position Property

- static: Default. Elements are positioned according to the normal flow of the document.
- relative: Positioned relative to its normal position.
- absolute: Positioned relative to the nearest positioned ancestor.
- fixed: Positioned relative to the viewport.
- sticky: A hybrid of relative and fixed, depending on the scroll position.

2. Display Property

- block: The element is displayed as a block.
- inline: The element is displayed as an inline element.
- inline-block: The element is displayed as an inline-level block container.
- none: The element is not displayed.

3. Float and Clear Properties

- float: Floats an element to the left or right.
- clear: Specifies whether an element should be next to floating elements or be moved down (none, left, right, both).

4. Flexbox

- display: flex: Creates a flex container.
- Properties like flex-direction, justify-content, align-items, flex-wrap control

the flex layout.

5. Grid

- `display: grid;` Creates a grid container.
- Properties like `grid-template-columns`, `grid-template-rows`, `grid-area`, `grid-gap` control the grid layout.

| | |
|--|--------------------------------|
|  | <h2>Operation Sheet : 1-6</h2> |
| <p>Title: CSS positioning and layout properties</p> <p>Objective: This operation sheet provides step-by-step instructions on how to apply positioning layout</p> <p>Instruction! Create a simple webpage to practice various CSS positioning and layout properties. Step-by-Step Guide</p> <p>1. Setup Your HTML File:</p> <p>Task1. Create an index.html file and add the following structure:</p> <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>CSS Positioning and Layout Practice</title> <link rel="stylesheet" href="styles.css"> </head></pre> | |

```
<body>

    <div class="container">

        <header class="header">Header</header>

        <nav class="nav">Navigation</nav>

        <main class="main">Main Content</main>

        <aside class="aside">Sidebar</aside>

        <footer class="footer">Footer</footer>

    </div>

</body>

</html>
```

2.Create Your CSS File:

Task1: Create a styles.css file and add the following styles:

```
/* Basic body styles for alignment */
```

```
body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

.container {

    display: flex;

    flex-wrap: wrap;
```

```
height: 100vh;  
}  
  
.header, .nav, .main, .aside, .footer {  
    padding: 20px;  
    color: white;  
}  
  
.header {  
    background-color: #1abc9c;  
    flex: 1 100%;  
}  
  
.nav {  
    background-color: #3498db;  
    flex: 1;  
    order: 1;  
}  
  
.main {  
    background-color: #9b59b6;  
    flex: 2;  
    order: 2;  
}  
  
.aside {  
    background-color: #f39c12;
```

```
flex: 1;  
  
order: 3;  
  
}  
  
.footer {  
  
background-color: #e74c3c;  
  
flex: 1 100%;  
  
order: 4;  
  
}
```

3. Tasks and Experimentation

Task 1: Positioning

Apply different position properties to the .header and observe their effects. For example, make the header fixed so it stays at the top.

```
.header {  
  
position: fixed;  
  
top: 0;  
  
width: 100%;  
  
z-index: 1;  
  
}
```

Task 2: Display

Change the display property of the .nav to inline-block and see how it affects the layout.

css

[Copy code](#)

```
.nav {  
    display: inline-block;  
    width: 200px;  
}
```

Task 3: Float and Clear

Use float to float the .aside to the right and use clear to make sure the .footer does not float next to any floated elements.

```
.aside {  
    float: right;  
    width: 200px;  
}  
  
.footer {  
    clear: both;  
}
```

Task 4: Flexbox

Experiment with different flex properties such as flex-direction, justify-content, and align-items on the .container.

```
.container {  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;
```

```
}
```

Task 5: Grid

Convert the layout to a grid layout using display: grid and define grid columns and rows.

```
.container {
```

```
    display: grid;
```

```
    grid-template-areas:
```

```
        'header header'
```

```
        'nav main'
```

```
        'nav aside'
```

```
        'footer footer';
```

```
    grid-gap: 10px;
```

```
    height: 100vh;
```

```
}
```

```
.header {
```

```
    grid-area: header;
```

```
}
```

```
.nav {
```

```
    grid-area: nav;
```

```
}
```

```
.main {
```

```
    grid-area: main;
```

```
}
```

```
.aside {  
    grid-area: aside;  
}  
  
.footer {  
    grid-area: footer;  
}
```



Self-check 1-4:

1. Explain the CSS Box Model and its components. How does the box model influence layout and design in web development?
2. What is the difference between margin and padding in CSS? Provide examples of when you would use each.
3. How do border, padding, and margin affect the total width and height of an element in CSS?
4. What are the different types of display properties in CSS, and how do they affect the rendering of elements on a webpage? Provide examples of display: block, display: inline, display: inline-block, and display: flex.
5. Explain the difference between position: relative, position: absolute, and position: fixed in CSS. How do they affect the positioning of an element on a webpage?

Unit Summary

CSS, or Cascading Style Sheets, is a crucial tool in web development that controls the layout and appearance of a website. It separates content from design, enabling easier changes and updates. CSS is compatible with various devices and screen sizes, making it a versatile choice for responsive web design. However, it can lead to conflicts between different stylesheets and can be challenging for beginners due to its complexity. CSS uses selectors and properties to define style rules for a webpage, targeting specific elements and defining their appearance. To

insert CSS into a webpage, it can be linked to an external stylesheet, embedded directly into HTML code, or used inline styles.

Unit Review Questions

Part I: Multiple Choice:

Instructions: For the following multiple-choice questions, choose the best answer among the options provided.

1. What does CSS stand for in web development?
 - a. A) Computer Style Sheets
 - b. B) Cascading Style Sheets
 - c. C) Content Styling System
 - d. D) Creative Styling Solutions
2. Which HTML element is used to include CSS code directly within an HTML document?
 - a. A) <style>
 - b. B) <link>
 - c. C) <script>
 - d. D) <css>
3. What is the correct syntax to link an external CSS file to an HTML document?
 - a. A) <style src="styles.css">
 - b. B) <css src="styles.css">
 - c. C) <link rel="stylesheet" href="styles.css">
 - d. D) <script src="styles.css">
4. What is the purpose of CSS selectors?
 - a. A) To identify elements in an HTML document
 - b. B) To insert multimedia elements in a webpage
 - c. C) To define the layout of a web page
 - d. D) To create interactivity and animations on a webpage

5. Which CSS selector targets all `<p>` elements in an HTML document?
 - a. A) `.p`
 - b. B) `#p`
 - c. C) `p`
 - d. D) `<p>`
6. How do you add comments in CSS?
 - a. A) `/* This is a comment */`
 - b. B) `<!-- This is a comment -->`
 - c. C) `// This is a comment`
 - d. D) `# This is a comment`
7. What is the primary purpose of CSS comments?
 - a. A) To hide CSS code from the browser
 - b. B) To add notes and explanations for developers
 - c. C) To create visual effects on a webpage
 - d. D) To prevent users from accessing the CSS file
8. Which of the following is the correct way to comment out multiple lines of CSS code?
 - a. A) `/* line 1 line 2 line 3 */`
 - b. B) `<!-- line 1 line 2 line 3 -->`
 - c. C) `// line 1 line 2 line 3 //`
 - d. D) `/* line 1 \n line 2 \n line 3 */`
9. What is the purpose of the CSS id selector?
 - a. A) To select elements based on their class names
 - b. B) To select a single element based on its unique identifier
 - c. C) To group and style multiple elements together
 - d. D) To select elements based on their HTML tags
10. How do you select elements with a specific class name in CSS?
 - a. A) `class_name { }`
 - b. B) `#class_name { }`
 - c. C) `.class_name { }`
 - d. D) `class_name []`

11. Which CSS property is used to change the background color of an element?

- a. A) color
- b. B) background-color
- c. C) bgcolor
- d. D) color-background

12. Which CSS property is used to change the text color of an element?

- a. A) text-color
- b. B) color
- c. C) font-color
- d. D) text-style

13. Which CSS property controls the text size?

- a. A) font-size
- b. B) text-size
- c. C) font-style
- d. D) text-style

14. How do you make each word in a text start with a capital letter?

- a. A) text-transform: capitalize
- b. B) text-style: capital
- c. C) transform: capitalize
- d. D) font-variant: small-caps

15. Which CSS property is used to change the spacing between letters?

- a. A) letter-spacing
- b. B) spacing
- c. C) text-spacing
- d. D) character-spacing

16. Which CSS property is used to change the spacing between lines of text?

- a. A) line-spacing
- b. B) spacing
- c. C) line-height
- d. D) line-style

17. Which CSS property is used to align text?
- A) text-align
 - B) vertical-align
 - C) text-align-vertical
 - D) text-align-horizontal
18. Which CSS property is used to create space between the element's border and inner content?
- A) margin
 - B) border-spacing
 - C) padding
 - D) spacing
19. Which CSS property is used to create space outside the element's border?
- A) margin
 - B) border-spacing
 - C) padding
 - D) spacing
20. Which CSS property is used to specify the width of an element's border?
- A) border-width
 - B) border-height
 - C) border-size
 - D) border-style
21. Which CSS property is used to change the font of an element?
- A) font-style
 - B) font-family
 - C) font-weight
 - D) font-size
22. Which CSS property makes text bold?
- A) font-weight: bold
 - B) font-style: bold
 - C) text-weight: bold

- d. D) text-style: bold
23. Which CSS property is used to control the order of flexible items in a flex container?
- A) order
 - B) flex-order
 - C) item-order
 - D) arrange
24. Which CSS property is used to define a grid layout?
- A) grid-layout
 - B) grid
 - C) display: grid
 - D) layout-grid
25. Which CSS property is used to make an element disappear from the document without affecting the layout?
- A) display: none
 - B) visibility: hidden
 - C) opacity: 0
 - D) position: absolute

Unit 2

Layout and navigation of a web page

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

1. Division and multicolumn layouts in HTML document
2. Navigation Formatting

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- Create layout and navigation of a web page

Key Terms: *Headers, Menu, Navigation, Content and Footer*

2.1. Division and multicolumn layouts

The CSS **multi-column layout** module lets you divide content across multiple columns. By using the properties in this module, you can define the preferred number and width of columns, the gap size between columns, and the visual appearance of the optional column dividing lines (known as column rules). You can also define how content should flow from column to column and how to break content between columns.

A website is often divided into headers, Menu/Navigation, content and a footer:

The following picture summarizes the layout structure:

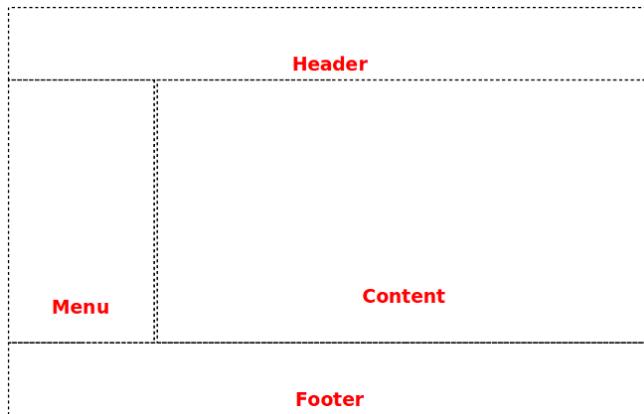


Figure 2.1: Division and multicolumn layouts

Header: The header is located at the top of the webpage and typically contains branding elements such as the website logo, site title, and sometimes a tagline. It may also include navigation links, contact information, or search bars.

Menu/Navigation: The menu or navigation section provides links to different pages or sections within the website. It enables users to navigate between different parts of the website easily.

Content: The content area is where the main information or functionality of the website is presented. It may include text, images, videos, forms, or any other interactive elements. The content section is the main body of the website. The user can divide the content section in an n-column layout. The layout in this section often depends on the target users. The most common layout is one (or combining them) of the following:

- **1-column** (often used for mobile browsers)
- **2-column** (often used for tablets and laptops)
- **3-column layout** (only used for desktops)
-

1 Column Layout: It is mostly used for mobile layout.

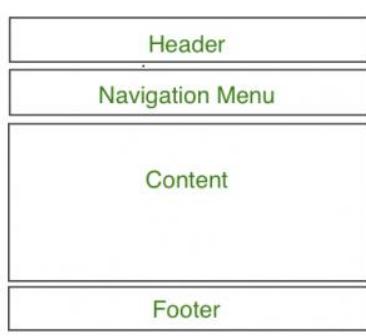


Figure 2.2: One Column Layout

2 Column Layout: This website layout is mostly used for tablets or laptops.

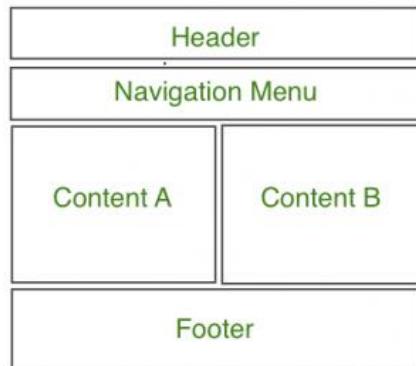


Figure 2.3: Two Column Layout

3 Column Layout: This website layout is mostly used for desktops.

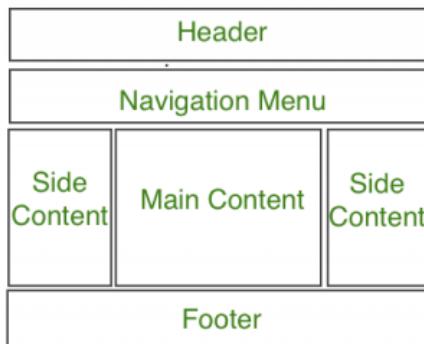


Figure 2.4: Three Column Layout

The user can also create a responsive layout where the layout will get changed as per screen size.

Footer: The footer is located at the bottom of the webpage and typically contains additional information such as copyright notices, legal disclaimers, contact information, or links to related pages.

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

1. CSS framework
2. CSS float property
3. CSS flexbox
4. CSS grid
5. CSS Positioning

CSS Frameworks

If you want to create your layout fast, you can use a CSS framework, like [W3.CSS](#) or [Bootstrap](#).

CSS Float Layout

It is common to do entire web layouts using the CSS **float** property. Float is easy to learn - you just need to remember how the **float** and **clear** properties work. Disadvantages: Floating elements are tied to the document flow, which may harm the flexibility.

CSS Flexbox Layout

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

CSS Grid Layout

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use

CSS Positioning

Positioning property is basically used to alter the element's natural layout flow position on the page and customize it according to our needs. It can be used to fix an element somewhere on the page, give a relative or an absolute position, or no special position at all.

2.2. Layout using HTML div and table tags

Creating Website Layouts

Creating a website layout is the activity of positioning the various elements that make a web page in a well-structured manner and give appealing look to the website. You have seen most websites on the internet usually display their content in multiple rows and columns, formatted like a magazine or newspaper to provide the users a better reading and writing environment. This can be easily achieved by using the HTML tags, such as <table>, <div>, <header>, <footer>, <section>, etc. and adding some CSS styles to them.

2.2.1. Layout using HTML 4 div tags

Using the <div> elements is the most common method of creating layouts in HTML. The <div> (stands for division) element is used for marking out a block of content, or set of other elements inside an HTML document. It can contain further other div elements if required.

The following example uses the div elements to create a multiple column layout. It will produce the same result as in the previous example that uses table element:

Code Description: *The code to create a layouts using div in forms*

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">

<title>HTML Div Layout</title>

<style>

    body {
        font: 14px Arial,sans-serif;
        margin: 0px;
    }

    .header {
        padding: 10px 20px;
        background: #acb3b9;
    }

    .header h1 {
        font-size: 24px;
    }

    .container {
        width: 100%;
        background: #f2f2f2;
    }

    .nav, .section {
        float: left;
        padding: 20px;
        min-height: 170px;
        box-sizing: border-box;
    }


```

```
}
```

```
.nav {
```

```
    width: 20%;
```

```
    background: #d4d7dc;
```

```
}
```

```
.section {
```

```
    width: 80%;
```

```
}
```

```
.nav ul {
```

```
    list-style: none;
```

```
    line-height: 24px;
```

```
    padding: 0px;
```

```
}
```

```
.nav ul li a {
```

```
    color: #333;
```

```
}
```

```
.clearfix:after {
```

```
    content: ".";
```

```
    display: block;
```

```
    height: 0;
```

```
    clear: both;
```

```
    visibility: hidden;
```

```
}
```

```
.footer {  
    background: #acb3b9;  
    text-align: center;  
    padding: 5px;  
}  
  
</style>  
  
</head>  
  
<body>  
  
    <div class="container">  
  
        <div class="header">  
            <h1>Grade 11</h1>  
        </div>  
  
        <div class="wrapper clearfix">  
            <div class="nav">  
                <ul>  
                    <li><a href="#">Home</a></li>  
                    <li><a href="#">About</a></li>  
                    <li><a href="#">Contact</a></li>  
                </ul>  
            </div>  
            <div class="section">  
                <h2>Welcome to our site</h2>  
                <p>Here you will learn how to create websites...</p>  
            </div>  
        </div>  
    </div>
```

```
</div>

</div>

<div class="footer">

    <p>copyright © CTE.com</p>

</div>

</div>

</body>

</html>
```

Output Description: *The code above will produce the following output:*

Grade 11

[Home](#)

[About](#)

[Contact](#)

Welcome to our site

Here you will learn how to create websites...

copyright © CTE.com

We've created this layout using the CSS float techniques, since most web browsers support it. Alternatively, you can also use the CSS3 flexbox to create modern and more flexible layouts. See the tutorial on CSS3 flexible box layouts to learn about flexbox in detail.



Better web page layouts can be created by using the DIV elements and CSS. You can change the layout of all the pages of your website by editing just one CSS file. To learn about CSS in detail, please check out

Better web page layouts can be created by using the DIV elements and CSS. You can change the layout of all the pages of your website by editing just one CSS file. To learn about CSS in detail, please check out our CSS tutorial section.

2.2.2. Layout using table tags

Table provides the simplest way for creating layouts in HTML. Generally, this involves the process of putting the contents such as text, images, and so on into rows and columns. The following layout is created using an HTML table with 3 rows and 2 columns — the first and last row spans both columns using the table's colspan attribute:

Code Description: *The code to create layouts with 3 rows and 2 columns*

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>HTML Table Layout</title>

</head>

<body style="margin:0px;">

    <table style="width:100%; border-collapse:collapse; font:14px Arial,sans-serif;">
```

```
<tr>

    <td colspan="2" style="padding:10px 20px; background-color:#acb3b9;">

        <h1 style="font-size:24px;">Grade 11</h1>

    </td>

</tr>

<tr style="height:170px;">

    <td style="width:20%; padding:20px; background-color:#d4d7dc; vertical-align: top;">

        <ul style="list-style:none; padding:0px; line-height:24px;">

            <li><a href="#" style="color:#333;">Home</a></li>

            <li><a href="#" style="color:#333;">About</a></li>

            <li><a href="#" style="color:#333;">Contact</a></li>

        </ul>

    </td>

    <td style="padding:20px; background-color:#f2f2f2; vertical-align:top;">

        <h2>Welcome to our site</h2>

        <p>Here you will learn how to create websites...</p>

    </td>

</tr>

<tr>

    <td colspan="2" style="padding:5px; background-color:#acb3b9; text-align:center;">

        <p>copyright © CTE.com</p>

    </td>

</tr>
```

```
</table>  
  
</body>  
  
</html>
```

Output Description: *The code above will produce the following output:*

Grade 11

[Home](#)
[About](#)
[Contact](#)

Welcome to our site

Here you will learn how to create websites...

copyright © CTE.com



The method used for creating the layout in the above example is not wrong, but it's not recommended. Avoid tables and inline styles for creating layouts. Layouts created using tables often render very

2.2.3. Layout using Semantic HTML 5 Elements

HTML5 has introduced the new structural elements such as <header>, <footer>, <nav>, <section>, etc. to define the different parts of a web page in a more semantic way. HTML5 introduces several new structural elements to provide a more semantic way to define different parts of a web page.

The following table provides a brief overview of new HTML5 structural elements.

| HTML5 Element | Description |
|---------------|--|
| <header> | Represents a container for introductory content or a set of navigational links. Typically contains headings, logo, authorship information, etc. |
| <footer> | Represents a footer for its nearest sectioning content or sectioning root element. Typically contains information about the author, copyright information, links to related documents, etc. |
| <nav> | Represents a section of a page that links to other pages or to parts within the page: a section with navigation links. |
| <section> | Represents a generic section of a document, typically with a heading. Used for grouping together thematically related content. |
| <article> | Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable, e.g., a blog post, newspaper article, or user-submitted comment. |

| | |
|---------|--|
| <aside> | Represents a section of a page that consists of content that is tangentially related to the content around the aside element (e.g., a sidebar). |
| <main> | Represents the dominant content of the <body> of a document. The content inside the <main> element should be unique to the document and not include content that is repeated across documents such as sidebars, navigation links, etc. |

You can consider these elements as a replacement for commonly used classes such as .header, .footer, .nav, .section, etc. The following example uses the new HTML5 structural elements to create the same layout that we have created in the previous examples.

Code Description: *The code to create layouts with HTML 5*

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>HTML5 Web Page Layout</title>

<style>

body {

    font: 14px Arial,sans-serif;

    margin: 0px;

}

header {

    padding: 10px 20px;
```

```
background: #acb3b9;  
}  
  
header h1 {  
    font-size: 24px;  
}  
  
.container {  
    width: 100%;  
    background: #f2f2f2;  
}  
  
nav, section {  
    float: left;  
    padding: 20px;  
    min-height: 170px;  
    box-sizing: border-box;  
}  
  
section {  
    width: 80%;  
}  
  
nav {  
    width: 20%;  
    background: #d4d7dc;  
}  
  
nav ul {
```

```
list-style: none;  
line-height: 24px;  
padding: 0px;  
}  
  
nav ul li a {  
    color: #333;  
}  
  
.clearfix:after {  
    content: ".";  
    display: block;  
    height: 0;  
    clear: both;  
    visibility: hidden;  
}  
  
footer {  
    background: #acb3b9;  
    text-align: center;  
    padding: 5px;  
}  
  
</style>  
  
</head>  
  
<body>  
    <div class="container">
```

```
<header>

    <h1>Grade 11</h1>

</header>

<div class="wrapper clearfix">

    <nav>

        <ul>

            <li><a href="#">Home</a></li>

            <li><a href="#">About</a></li>

            <li><a href="#">Contact</a></li>

        </ul>

    </nav>

    <section>

        <h2>Welcome to our site</h2>

        <p>Here you will learn how to create websites...</p>

    </section>

</div>

<footer>

    <p>copyright © CTE.com</p>

</footer>

</div>

</body>

</html>
```

Output Description: *The code above will also produce the same output as the previous example*

The screenshot shows a website layout. At the top, a header bar contains the text "Grade 11". Below this, a sidebar on the left side contains three links: "Home", "About", and "Contact", each underlined. The main content area on the right has a title "Welcome to our site" and a subtitle "Here you will learn how to create websites...". At the bottom of the main content area, there is a copyright notice "copyright © CTE.com".

2.3. Navigation Formatting

Navigation formatting refers to the methods and principles used to design and organize the navigation system of a website. A well-formatted navigation system is essential for providing a positive user experience, as it helps users find the information they need quickly and easily.

A navigation bar, or navbar, in HTML, is a user interface section, typically on top of a website, that allows users to explore and access information. The navigation bar comes in a dropdown menu or a collapse navbar that has class names. An application with navigation features may also use HTML and CSS as a programming language to code elements of an application.

Here are some key concepts and best practices for navigation formatting:

Key Concepts:

1. Hierarchy:

- Create a clear hierarchy in your navigation structure, organizing content from general to specific.
- Use primary navigation for main sections and secondary navigation for subsections.

2. Consistency:

- Ensure navigation is consistent across all pages of the website to avoid confusing users.

3. Clarity:

- Use clear and descriptive labels for navigation links.
- Avoid jargon or ambiguous terms.

4. Accessibility:

- Make navigation accessible to all users, including those using screen readers.
- Ensure navigation links are keyboard navigable.

5. Responsiveness:

- Design navigation that adapts to different screen sizes and devices.
- Use techniques like hamburger menus or dropdowns for smaller screens.

Best Practices:**1. Grouping:**

- Group related links together.
- Use dropdown menus or mega menus for large websites with many sections.

2. Visual Design:

- Use visual cues like highlighting the active link, hover effects, or breadcrumb trails.
- Ensure there is sufficient contrast between text and background.

3. Simplicity:

- Avoid overcrowding the navigation menu with too many links.
- Aim for a balance between providing enough options and keeping it uncluttered.

2.3.1. Add Navigation or menu Bar

A navigation bar is a section of a graphical user interface (GUI) that helps users navigate through a website, app, or other software. It is essential for users to quickly and easily navigate to the content they are looking for.

The navigation bar can be horizontal or vertical, that contains links to important pages or features.

Navbars can also contain other elements, such as the logo of the website or app, search bar, or social media icons. Navbars can be styled using CSS to change their appearance.

Uses for Navigation Bar in HTML

2. Organization. Having a navigation menu for your website makes sure that the list items for the dropdown menu are coordinated and logically formulated. Organized navigation links are visually appealing to website visitors.
3. Ease of Use. Navigating an unordered list of links is inconvenient and might turn traffic away from your website. Well-organized navigation bars that are easy to use provide a convenient experience for people that visit your website.
4. Accessibility. Good-looking navigation bars provide easier access to information that allows for a streamlined user experience.

2.3.2 CSS Horizontal Navbar

Following example shows a horizontal navigation bar, which is the most common type of navigation bar displayed across the top of a web page as shown below –

Code Description: *HTML code to create a Horizontal Navbar*

```
<!DOCTYPE html>

<html>

<head>

<style>

    ul {
```

```
background-color: #28bf17;  
overflow: hidden;  
list-style-type: none;  
margin: 0;  
padding: 0;  
}  
  
li {  
float: left;  
}  
  
li a {  
display: block;  
color: #f2f2f2;  
text-align: center;  
padding: 10px;  
text-decoration: none;  
font-size: 17px;  
}  
  
li a:hover {  
background-color: #dd9ede;  
color: black;  
}  
  
.active-link {  
background-color: #f53319;
```

```
color: white;  
}  
</style>  
</head>  
<body>  
<ul>  
<li><a href="#" class="active-link">TutorialsPoint</a></li>  
<li><a href="#">Home</a></li>  
<li><a href="#">Articles</a></li>  
<li><a href="#">Courses</a></li>  
<li><a href="#">eBook</a></li>  
</ul>  
<h1>Welcome to TutorialsPoint</h1>  
<h3>Simple Easy Learning at your fingertips</h3>  
</body>  
</html>
```

Output Description: The code above will also produce the same output as the previous example

TutorialsPoint Home Articles Courses eBook

Welcome to TutorialsPoint

Simple Easy Learning at your fingertips

2.3.3. CSS Vertical Navbar

A vertical navigation bar is also known as a sidebar menu. It is typically positioned on the left or right side of the screen. Here is an example –

Code Description: *HTML code to create a Horizontal Navbar*

```
<html>

<head>

<style>

ul {

background-color: #28bf17;

list-style-type: none;

margin: 0;

padding: 0;

width: 200px;

}

li {

text-align: center;

}

li a {

display: block;

color: #f2f2f2;

text-align: center;

padding: 10px;

text-decoration: none;
```

```
font-size: 17px;  
}  
  
li a:hover {  
  
background-color: #dd9ede;  
  
color: black;  
}  
  
.active-link {  
  
background-color: #f53319;  
  
color: white;  
}  
  
</style>  
  
</head>  
  
<body>  
  
<ul>  
  
<li><a href="#" class="active-link">Tutorialspoint</a></li>  
  
<li><a href="#">Home</a></li>  
  
<li><a href="#">Articles</a></li>  
  
<li><a href="#">Courses</a></li>  
  
<li><a href="#">eBook</a></li>  
  
</ul>  
  
</body>  
  
</html>
```

Output Description: The code above will also produce the same output as the previous example



2.3.4. CSS Dropdown Navbar

A dropdown navbar is a navigation bar with a drop-down menu that appears when a user hovers over a link. Dropdown menus are a way to show a list of related items in a small space. Here is an example –

Code Description: *HTML code to create a dropdown Navbar*

```
<html>
<head>
<style>
.navbar {
    background-color: #28bf17;
    overflow: hidden;
}
.navbar a {
```

```
display: block;  
float: left;  
color: #f2f2f2;  
text-align: center;  
padding: 10px;  
text-decoration: none;  
font-size: 15px;  
}  
  
.navbar a:hover {  
background-color: #dd9ede;  
color: black;  
}  
  
.active-link {  
background-color: #f53319;  
color: white;  
}  
  
.dropdown {  
float: left;  
}  
  
.dropdown .dropbtn {  
border: none;  
color: #f2f2f2;  
padding: 10px;
```

```
background-color: #28bf17;  
}  
  
.dropdown-content {  
display: none;  
position: absolute;  
background-color: #c7385a;  
min-width: 120px;  
}  
  
.dropdown-content a {  
float: none;  
color: #f2f2f2;  
padding: 10px;  
display: block;  
text-align: left;  
}  
  
.dropdown-content a:hover {  
background-color: #dd9ede;  
color: black;  
}  
  
.dropdown:hover .dropdown-content {  
display: block;  
}  
  
</style>
```

```
</head>

<body>

<nav class="navbar">

    <a href="#" class="active-link">Tutorialspoint</a>

    <a href="#">Home</a>

    <div class="dropdown">

        <button class="dropbtn">Articles</button>

        <div class="dropdown-content">

            <a href="#">HTML</a>

            <a href="#">CSS</a>

            <a href="#">Bootstrap</a>

        </div>

    </div>

    <a href="#">Courses</a>

    <a href="#">eBook</a>

</nav>

<h1>Welcome to Tutorialspoint</h1>

<h3>Simple Easy Learning at your fingertips</h3>

</body>

</html>
```

Output Description: *The code above will also produce the same output as the previous example*

Tutorialspoint Home Articles Courses eBook

Welcome to TutorialsPoint

Simple Easy Learning at your fingertips

2.3.5. CSS Fixed Navbar

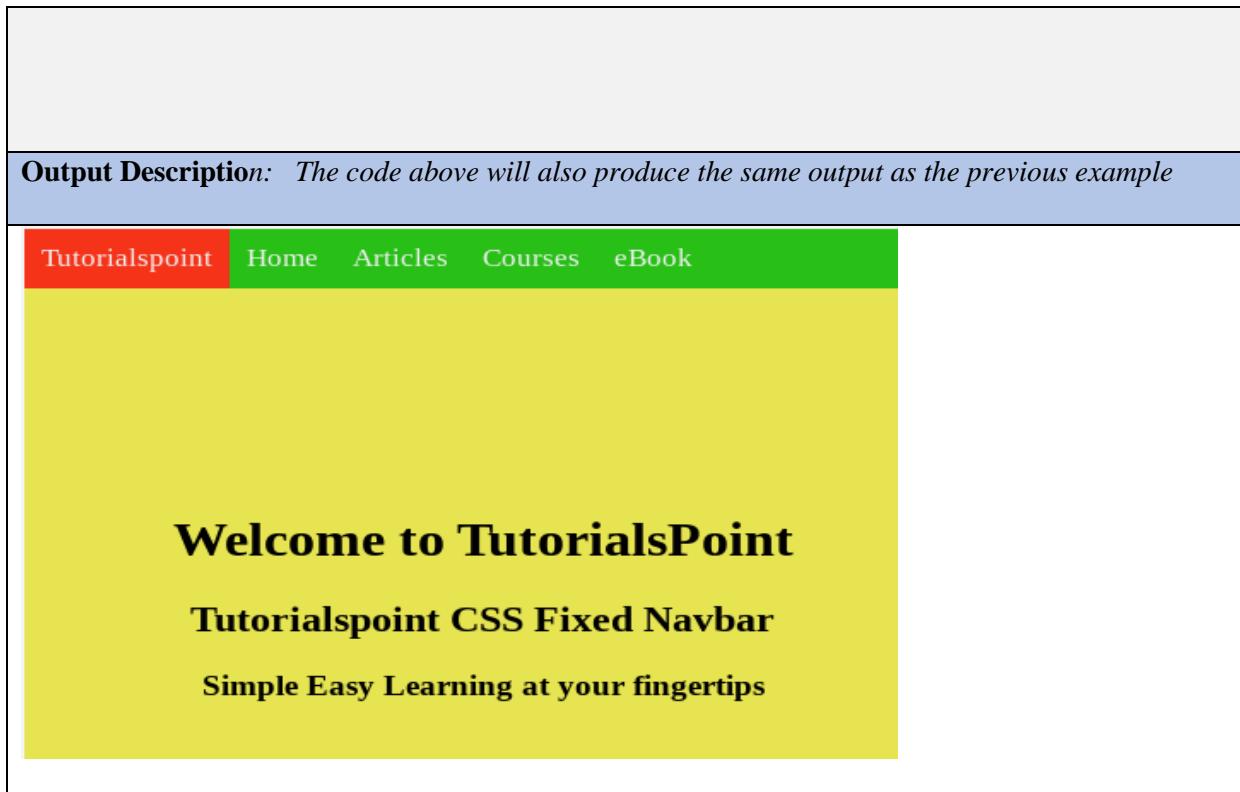
A fixed navbar is a navigation bar that sticks to the top of the screen when the user scrolls down the page. To make a navbar fixed, you can use the position: fixed property. Here is an example –

Code Description: *HTML code to create a Fixed Navbar*

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }
      .navbar {
        position: fixed;
        top: 0;
        width: 100%;
        margin: 0;
      }
    </style>
  </head>
  <body>
    <div class="navbar">
      <h1>Welcome to TutorialsPoint</h1>
      <p>Simple Easy Learning at your fingertips</p>
    </div>
  </body>
</html>
```

```
padding: 0px;  
  
overflow: hidden;  
  
background-color: #28bf17;  
  
}  
  
.navbar a {  
  
display: block;  
  
float: left;  
  
color: #f2f2f2;  
  
text-align: center;  
  
padding: 10px;  
  
text-decoration: none;  
  
font-size: 17px;  
  
}  
  
.navbar a:hover {  
  
background-color: #dd9ede;  
  
color: black;  
  
}  
  
.active-link {  
  
background-color: #f53319;  
  
color: white;  
  
}  
  
.heading {
```

```
padding-top: 170px;  
  
text-align: center;  
  
background-color: #e6e451;  
  
padding-bottom: 300px  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<nav class="navbar">  
  
    <a href="#" class="active-link">TutorialsPoint</a>  
  
    <a href="#">Home</a>  
  
    <a href="#">Articles</a>  
  
    <a href="#">Courses</a>  
  
    <a href="#">eBook</a>  
  
</nav>  
  
<div class="heading">  
  
    <h1>Welcome to TutorialsPoint</h1>  
  
    <h2>TutorialsPoint CSS Fixed Navbar</h2>  
  
    <h3>Simple Easy Learning at your fingertips</h3>  
  
</div>  
  
</body>  
  
</html>
```



2.3.5. CSS Sticky Navbar

You can use the position: sticky property to create a sticky navbar, which will stay at the top of the screen even when the user scrolls down the page. Here is an example –

Code Description: *HTML code to create a Sticky Navbar*

```
<html>  
  
<head>  
  
<style>  
  
body {  
  
    margin: 0;  
  
}  
  
.navbar {  
  
    position:sticky;
```

```
top: 0;  
width: 100%;  
margin: 0;  
padding: 0px;  
overflow: hidden;  
background-color: #28bf17;  
}  
  
.navbar a {  
display: block;  
float: left;  
color: #f2f2f2;  
text-align: center;  
padding: 10px;  
text-decoration: none;  
font-size: 17px;  
}  
  
.navbar a:hover {  
background-color: #dd9ede;  
color: black;  
}  
  
.active-link {  
background-color: #f53319;  
color: white;
```

```
}

.heading {
    padding-top: 170px;
    text-align: center;
    background-color: #e6e451;
    padding-bottom: 300px
}

h2 {
    text-align: center;
}

</style>

</head>

<body>

<h2>Scroll down to see the effect</h1>

<nav class="navbar">

<a href="#" class="active-link">Tutorialspoint</a>

<a href="#">Home</a>

<a href="#">Articles</a>

<a href="#">Courses</a>

<a href="#">eBook</a>

</nav>

<div class="heading">

<h1>Welcome to Tutorialspoint</h1>
```

```
<h2>Tutorialspoint CSS Sticky Navbar</h2>

<h3>Simple Easy Learning at your fingertips</h3>

</div>

</body>

</html>
```

Output Description: *The code above will also produce the same output as the previous example*

Scroll down to see the effect

Tutorialspoint Home Articles Courses eBook

Welcome to TutorialsPoint

Tutorialspoint CSS Sticky Navbar

Simple Easy Learning at your fingertips

2.4. CSS properties for the navigation bar

Styling a navigation bar involves using a variety of CSS properties to control layout, appearance, and interactivity. Here are some essential CSS properties for a navigation bar, along with explanations and examples of how to use them:

I. Background Color (background-color)

Sets the background color of the navigation bar.

```
nav { background-color: #333; }
```

II. Text Color (color)

Sets the color of the text within the navigation bar.

```
nav a {  
    color: white;  
}
```

III. Padding (padding)

Adds space inside the navigation bar, affecting the spacing around the navigation items.

```
nav {  
    padding: 10px 20px;  
}
```

IV. 4. Margin (margin)

Adds space outside the navigation bar, affecting its positioning relative to other elements.

```
nav {  
    margin: 0;  
}
```

V. 5. List Style (list-style-type)

Removes default list styling for navigation items when using an unordered list for navigation.

```
nav ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

VI. Text Alignment (text-align)

Aligns the text within the navigation items horizontally.

```
nav {  
    text-align: center;  
}
```

VII. Text Decoration (text-decoration)

Sets or removes text decorations such as underlines from the navigation links.

```
nav a {  
    text-decoration: none;  
}
```

VIII. Display (display)

Defines how the navigation items are displayed. Common values include flex, block, inline-block, and inline.

```
nav ul {  
    display: flex;  
    justify-content: space-around;  
}
```

IX. Flex Properties (flex, justify-content, align-items)

Used for flexible box layout (Flexbox) to arrange navigation items.

```
nav ul {  
    display: flex;
```

```
justify-content: space-between;  
align-items: center;  
}
```

X. Width and Height (width, height)

Sets the width and height of the navigation bar or individual items.

```
nav {  
    width: 100%;  
    height: 60px;  
}
```

XI. Border (border)

Adds a border around the navigation bar or items within it.

```
nav {  
    border-bottom: 2px solid #444;  
}
```

XII. Box Shadow (box-shadow)

Adds shadow effects to create depth and dimension.

```
nav {  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}
```

XIII. Hover Effects (:hover)

Changes the style of navigation links when hovered over for better interactivity.

```
nav a:hover {  
    background-color: #555;  
}
```

XIV. Positioning (position, top, left, right, bottom)

Controls the positioning of the navigation bar on the page.

```
nav {  
    position: fixed;  
    top: 0;  
    width: 100%;  
}
```

XV. Font Properties (font-size, font-family)

Sets the font size and font family for the navigation links.

```
nav a {  
    font-size: 16px;  
    font-family: Arial, sans-serif;  
}
```

Example CSS for a Navigation Bar

Here is a complete example combining these properties to create a styled navigation bar:

```
/* Navigation bar styles */  
  
nav {  
    background-color: #333;  
    color: white;  
    padding: 10px 0;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    width: 100%;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    position: fixed;  
    top: 0;  
}
```

```
/* Navigation list styles */
nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    display: flex;
}

/* Navigation list item styles */
nav li {
    margin: 0 15px;
}

/* Navigation link styles */
nav a {
    color: white;
    text-decoration: none;
    padding: 14px 20px;
    display: block;
    transition: background-color 0.3s ease;
    font-size: 16px;
    font-family: Arial, sans-serif;
}

/* Hover effect for navigation links */
nav a:hover {
    background-color: #555;
}
```

Dear Students! I'll guide you through the process of creating a simple dropdown menu using HTML and CSS.



Operation Sheet : 2-1

Title: Dropdown /popup menu

Objective: This operation sheet provides step-by-step instructions on how to apply Dropdown /popup menu

Instructions: Create a dropdown /popup menu with HTML and CSS

Project 1: simple dropdown menu

Task 1: Add HTML

We'll start by creating the HTML structure for our dropdown menu. We'll use an unordered list `` to represent the menu items and nested lists to create submenus. Here is how our HTML markup will look like:

```
<nav>  
  
    <ul>  
  
        <li><a href="#">Home</a></li>  
  
        <li>  
  
            <a href="#">Resources</a>  
  
            <ul>  
  
                <li><a href="#">Blog</a></li>
```

```
<li><a href="#">Academy</a></li>

<li><a href="#">Reviews</a></li>

</ul>

</li>

<li>

<a href="#">Newsletters</a>

<ul>

<li><a href="#">Blog</a></li>

<li><a href="#">Socials</a></li>

<li><a href="#">Reviews</a></li>

</ul>

</li>

<li><a href="#">About Us</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>
```

Task 2: Add CSS Styling

Step 1. We'll start by removing the default list styles and resetting the default padding and margin to eliminate any spacing around the list.

Here's the CSS:

```
nav ul {

list-style: none;

margin: 0;
```

```
padding: 0;  
  
background-color: yellow;  
  
}
```

After that, let's style the main menu `` items within the `<nav>` element. We'll set them to display as inline-block elements, which places them horizontally next to each other.

Here's the CSS:

```
nav ul li {  
  
display: inline-block;  
  
position: relative;  
  
}
```

Next, we'll style the `<a>` elements by:

- Setting them to display as block elements. so they occupy the entire width of their parent list items.
- Adding some padding for spacing around the text.
- We'll add some transition to create a smooth color change effect when you hover over the menu items.
- Changing the background-color on hover to make it a little bit visually appealing.

Here's the CSS:

```
nav ul li a {  
  
display: block;  
  
padding: 10px 20px;  
  
text-decoration: none;  
  
color: #333;  
  
transition: background-color 0.3s;
```

```
}
```

```
nav ul li a:hover {  
    background-color: #f0f0f0;  
}
```

Step 2. Now, let's add some styles for the dropdown submenus within the main menu items by:

- Hiding them by default using display: none and give them a position: absolute so they are placed relative to their parent list items.
- we'll also set a hover effect on the main menu , which displays the associated submenu by changing the display property to block.

Here's the CSS:

```
nav ul li ul {  
  
    display: none;  
  
    position: absolute;  
  
    background-color: #fff;  
  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
  
    border: 1px solid #ccc;  
  
}  
  
nav ul li:hover ul {  
  
    display: block;  
  
}
```

Finally, To finish up styling the dropdown menu, we'll add the following:

- Styles the individual items within the dropdown submenu. It ensures they are displayed as block-level elements and prevents wrapping of text using white-space: nowrap.
- Similar to the main menu items, we'll make them display as block-level elements, add padding for spacing and include a transition effect for background-color changes on hover.

Here's the CSS:

```
nav ul li ul li {  
    display: block;  
    white-space: nowrap;  
}  
  
nav ul li ul li a {  
    display: block;  
    padding: 10px 20px;  
    text-decoration: none;  
    color: #333;  
    transition: background-color 0.3s;  
}  
  
nav ul li ul li a:hover {  
    background-color: #f0f0f0;  
}
```

Project 2: create a comprehensive dropdown menu

Instruction: Presented here is a comprehensive HTML document that illustrates the utilization of various concepts and the development of a basic dropdown menu.

```
<style>

nav ul {

    list-style: none;

    margin: 0;

    padding: 0;

    background-color: yellow;

}

nav ul li {

    display: inline-block;

    position: relative;

}

nav ul li a {

    display: block;

    padding: 10px 20px;

    text-decoration: none;

    color: #333;

    transition: background-color 0.3s;

}

nav ul li a:hover {

    background-color: #f0f0f0;

}

nav ul li ul {

    display: none;

}
```

```
position: absolute;  
  
background-color: #fff;  
  
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
  
border: 1px solid #ccc;  
  
}  
  
nav ul li:hover ul {  
  
    display: block;  
  
}  
  
</style>  
  
<nav>  
  
<ul>  
  
<li><a href="#">Home</a></li>  
  
<li>  
  
    <a href="#">Resources</a>  
  
<ul>  
  
<li><a href="#">Blog</a></li>  
  
<li><a href="#">Academy</a></li>  
  
<li><a href="#">Reviews</a></li>  
  
</ul>  
  
</li>  
  
<li>  
  
    <a href="#">Newsletters</a>  
  
<ul>
```

```
<li><a href="#">Blog</a></li>

<li><a href="#">Socials</a></li>

<li><a href="#">Reviews</a></li>

</ul>

</li>

<li><a href="#">About Us</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>
```



Self-check 2-1:

-
1. Explain the role of the `<div>` and `` elements in HTML. How are they typically used in conjunction with CSS for layout purposes?
 2. Describe the differences between using CSS Grid and Flexbox for layout. When would you use one over the other?
 3. Explain the concept of a CSS framework. What are some popular CSS frameworks, and how do they simplify web development?
 4. Discuss the role of position property in creating fixed and sticky navigation bars?
-

Unit Summary

A well-structured web page layout relies on the use of CSS to style `div` and `section` elements, which serve as generic and semantic containers respectively. Effective navigation is crucial for user experience, typically implemented through a navigation bar (navbar) that includes responsive design principles to ensure accessibility and usability across devices. The navbar often features a combination of logos, navigation links, dropdown menus, and sometimes a search bar, facilitating intuitive and efficient navigation throughout the site.

Unit Review Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. Which CSS layout method provides a two-dimensional layout system that allows precise control over both rows and columns?
 - a) Flexbox
 - b) Float Layout
 - c) Grid
2. Which layout technique is commonly used for creating responsive and flexible layouts, allowing items to be dynamically arranged within a container?
 - a) Grid
 - b) Flexbox
 - c) Float Layout
3. What HTML element is typically used to define a logical section within a webpage, representing a thematic grouping of content?
 - a) `<division>`
 - b) `<section>`
 - c) `<container>`
4. Which CSS property is primarily used to define the alignment and positioning of elements within a Flexbox container?
 - a) `justify-content`
 - b) `grid-template-columns`
 - c) `float`

5. Which layout method is commonly used for creating multi-column layouts before the introduction of more modern approaches like Flexbox and Grid?
 - a) Grid
 - b) Float Layout
 - c) Flexbox

6. Which HTML tag is typically used to enclose the main content area of a webpage, excluding headers, footers, and sidebars?
 - a) <content>
 - b) <main>
 - c) <body>

7. Which CSS property is used to define the alignment of items within a Grid container along the inline axis (horizontally)?
 - a) align-items
 - b) justify-content
 - c) align-content

8. What HTML element is commonly used to create a horizontal line or divider between sections of content?
 - a) <hr>
 - b) <divider>
 - c) <line>

9. Which layout method is best suited for creating complex, responsive layouts with multiple rows and columns, especially when the order of elements matters?
 - a) Flexbox
 - b) Grid
 - c) Float Layout

10. Which CSS property is used to specify the size of the columns and rows in a Grid layout?
 - a) grid-template-columns
 - b) flex-direction
 - c) float

11. Which HTML element is typically used to define a group of navigation links within a webpage?
 - a) <navbar>
 - b) <nav>
 - c) <menu>

12. Which layout method is characterized by the ability to align items both horizontally and vertically within a container, making it suitable for centering content?
 - a) Float Layout
 - b) Grid
 - c) Flexbox

13. Which CSS property is used to define the size of Flex items relative to the other items in the Flex container?
 - a) flex-grow
 - b) grid-template-columns
 - c) align-items

14. What HTML element is typically used to enclose content that is tangentially related to the main content of a webpage, such as sidebars or callout boxes?
 - a) <sidebar>
 - b) <aside>
 - c) <section>

15. Which layout method is preferred for creating a simple, one-dimensional layout where elements flow along the document flow?
 - a) Grid
 - b) Flexbox
 - c) Float Layout

16. Which CSS layout method is best suited for creating complex, responsive layouts with multiple columns and rows?
 - a) Grid
 - b) Flexbox
 - c) Float Layout

17. Which layout technique is commonly used for creating navigation menus, headers, and footers that adjust dynamically to different screen sizes?
- Flexbox
 - Grid
 - Float Layout
18. Which layout method was traditionally used for creating multi-column layouts before the advent of more modern approaches like Flexbox and Grid?
- Grid
 - Float Layout
 - Flexbox

Answer key for Self-check questions

Self-Check 2-1: Answer Key

1. `<div>` and `` are generic container elements in HTML.

 - `<div>`: A block-level element used to group larger chunks of content, such as sections of a webpage. It can be styled with CSS to define layout structures, like columns or grids.
 - ``: An inline element used to group small portions of text or other inline elements. It's often used for styling parts of text or adding CSS classes without breaking the flow of content. Example:

```
<div class="container">
  <div class="header">Header</div>
  <div class="content">
    <span class="highlight">Highlighted text</span> in the content.
  </div>
  <div class="footer">Footer</div>
</div>
```
2. A). CSS Grid: Designed for two-dimensional layouts. It allows you to create complex grid-based designs with rows and columns. It's ideal for layout structures that require precise placement of items on both axes.

B). Flexbox: Designed for one-dimensional layouts, either a row or a column. It's excellent for distributing space within a container and aligning items in a single direction (horizontal or vertical).

3. CSS frameworks are pre-prepared libraries that are meant to be used as a foundation for developing responsive and visually consistent web pages. They provide predefined classes for layout, components, and utilities.

Popular CSS Frameworks:

- Bootstrap: Offers a wide range of components and utilities, with a strong grid system and responsive design.
- Foundation: Known for its flexible and customizable components.
- Bulma: A modern CSS framework based on Flexbox.

Advantages:

- Simplify the process of creating responsive layouts.
- Ensure cross-browser compatibility.
- Speed up development by using predefined styles and components.

4. A). Fixed Navigation Bar: Stays in the same position relative to the viewport, even when the page is scrolled.
5. B). Sticky Navigation Bar: Behaves like relative positioning until a certain point in the scroll, after which it becomes fixed.

Fixed Example:

```
nav {  
  
    position: fixed;  
  
    top: 0;  
  
    width: 100%;  
  
    background-color: white;  
}
```

Sticky Example:

```
nav {  
  
    position: -webkit-sticky; /* For Safari */  
    position: sticky;  
    top: 0;  
  
    background-color: white;  
}  
  
0;  
  
color: #fff;  
}
```

References

Module VI

Interactivity To a Website



Contents

| No | Contents | page |
|----------|---|------|
| | Module Description | 572 |
| 1 | UNIT 1: SCRIPTING LANGUAGE | 573 |
| | 1.1 Identify scripting language | 584 |
| | 1.2 Introduction to Javascript (JS) | 585 |
| | Unit Summary | 587 |
| | Unit Review Questions | 588 |
| 2 | UNIT 2: Apply INTERACTIVITY USING JAVASCRIPT | 590 |
| | 2.1 Overview of interactivity using Javascript | 590 |
| | 2.2 JavaScript Syntax and Basic Concepts | 591 |
| | 2.3 Control Structures and Loops | 611 |
| | 2.4 Functions in JavaScript | 622 |
| | 2.5 DOM Manipulation and Event Handling | 636 |
| | 2.6 Document Object Model (DOM) | 636 |
| | Unit Summary | 646 |
| | Unit Review Questions | 646 |
| 3 | UNIT 3: VALIDATING USER INPUT | 649 |
| | 3.1 Introduction to Validation | 649 |
| | Unit Summary | 655 |
| | Unit Review Questions | 655 |
| | References | 663 |

Module Description

The module "Apply interactivity to a website" refers to a specific section or unit of study that focuses on teaching the concepts, techniques, and tools necessary to add dynamic and interactive elements to a website. It typically covers topics related to client-side web development using technologies such as HTML, CSS, and JavaScript.

The module may include the following key aspects:

- Identify scripting language
- Apply interactivity using JavaScript
- Validating User Input

Module Instruction:

Learning Instructions: How to use this Module

For effectively use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what are being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.
5. Complete the ‘Project Work’ sited at the end of the module.

UNIT 1

SCRIPTING LANGUAGE

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 1.1. Introduction to JavaScript (JS)

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Identify scripting language

Key Terms: *JavaScript, syntax, DOM, loops.*

1.1 Identify scripting language

To identify a scripting language, you can look for certain characteristics and features that are commonly associated with scripting languages. Here are a few key indicators:

1. Interpreted execution: Scripting languages are typically interpreted rather than compiled, meaning the code is executed by an interpreter at runtime rather than being compiled into machine code beforehand.
2. High-level syntax: Scripting languages often have a simplified and high-level syntax that is easy to read and write, making them accessible to non-programmers and beginners.
3. Dynamic typing: Scripting languages usually employ dynamic typing, which means you don't need to explicitly declare variable types. Variables can change their type during runtime.
4. Built-in libraries and frameworks: Scripting languages often provide extensive libraries and frameworks that make it easier to perform common tasks and interact with various system resources or APIs.

5. Flexibility and rapid development: Scripting languages are often designed to prioritize quick development and flexibility, allowing developers to write and modify code rapidly.

Examples of popular scripting languages include:

- JavaScript (primarily used for web development but can also be used for server-side programming with Node.js)
- Python, Ruby, PHP, Perl, Bash (shell scripting language), PowerShell, etc.

It's important to note that some languages can fall into both the scripting and general-purpose categories, and the distinction between them can sometimes be blurry. However, these characteristics can help you identify a language as a scripting language in most cases.

1.2. Introduction to JavaScript (JS)

By applying interactivity to a website, you can make it more engaging, user-friendly, and interactive, leading to improved user satisfaction and increased user interaction with the website's content.

"Apply interactivity to a website" refers to the process of adding dynamic and interactive elements to a website to enhance the user experience and engage the visitors. It involves using programming languages like JavaScript to create interactive features, handle user interactions, and manipulate the content of the webpage in real-time.

When a website is interactive, it allows users to actively engage with the content, perform actions, and receive immediate feedback.

1.2.1. Preface of JavaScript:

JavaScript is a high-level, interpreted programming language that was initially created to add interactivity to web pages. It was developed by Brendan Eich in 1995 and has since become one of the most widely used languages for web development. JavaScript allows developers to add functionality to web pages, manipulate the Document Object Model (DOM), and create responsive user interfaces. Unlike HTML and CSS, which focus on structure and styling, JavaScript is a scripting language that enables client-side scripting, making web pages more dynamic.

1.2.2. Role of JavaScript in web development:

JavaScript plays a vital role in web development by enabling dynamic and interactive features on web pages. It runs on the client-side, meaning it executes within the user's web browser, allowing it to manipulate the Document Object Model (DOM) and respond to user interactions. JavaScript is used to enhance user experience, validate form inputs, create interactive elements, handle events, fetch and display data from APIs, and perform various other tasks that make websites more engaging and user-friendly.

With the introduction of technologies like Node.js, JavaScript can also be used for server-side development, allowing developers to build full-stack web applications using a single language.

JavaScript is a cornerstone of modern web development. Its versatility extends beyond just client-side scripting; it's also used on the server side (Node.js) and in mobile app development (React Native, Native Script). JavaScript frameworks and libraries, such as React, Angular, and Vue, have revolutionized how developers build user interfaces. The rise of Single Page Applications (SPAs) and asynchronous programming, facilitated by JavaScript, has enhanced the overall user experience on the web. In essence, JavaScript is an essential language for anyone involved in web development.

- JavaScript is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning JavaScript:
- JavaScript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt JavaScript, it helps you developing great front-end as well as back-end software's using different JavaScript based frameworks like jQuery, Node.JS etc.
- JavaScript is everywhere, it comes installed on every modern web browser and so to learn JavaScript you really do not need any special environment setup. For example, Chrome, Mozilla Firefox, Safari and every browser you know as of today, supports JavaScript.
- JavaScript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.

- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as JavaScript Programmer.
- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills look like in the job market.
- Great thing about JavaScript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

1.2.3. Features of JavaScript:

JavaScript offers several features that make it a powerful and versatile programming language for web development.

Features of JavaScript

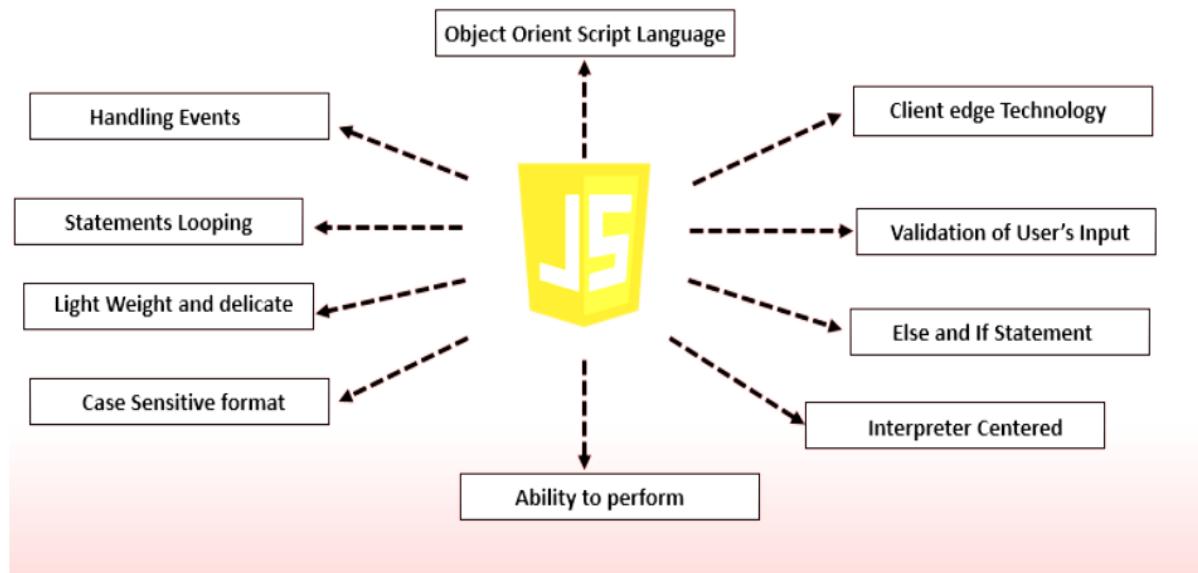


Fig: 1.1.1. Java Script Features

1. Object-Centered Script Language

Object Centered Language features built in the object as Java Script has a window object. Some Common Examples of Object Centered languages are Java Script and Visual Basic etc. The object-centered languages are mostly used for features like Polymorphism which is a quality of taking an object in many forms.

Use of **Polymorphism** within **object-oriented programming** requires whenever we use to represent reference of the parent class to an object of a child class.

2. Client Edge Technology

The client is basically a term used for Web Browser in respective of User. The user gets access to the client through a web browser for surfing and interacting through websites. The client edge technology in Java Script allows the client to have full control over the content which is being updated in servers.

3. Validation of User's Input

Validation of User's Input is most commonly known as form validation, it allows users to interact with client through filling forms through web pages. The details in the form need to be correctly filled where form validation helps the client to validate the details entered by the user.

4. Else and IF Statement

IF and Else Statements are used to perform logical operations.

5. Interpreter Centered

Java Script is built with Interpreter Centered which allows the user to get the output without the use of Compiler. That means the input performed by the user gets rendered directly without the compiling of codes.

6. Ability to Perform in Build Function

Java Script has many In-Built Functions like isNaN (), Number (), parseFloat () and parseInt () etc. **isNaN () Function** is used to identify that input object is correct number format. parseFloat () function is used in the conversion of the object into a number. parseInt () Function is used to analyze strings.

7. Case Sensitive Format

The codes written in Java Script are Case Sensitive which explains that there will be no difference in the output whether the codes are written in Upper Case or Lower-Case Format.

8. Light Weight and delicate

Java Script Features Light Weight and delicate and codes written in JavaScript don't include variables and uses only objects to perform the operations.

9. Statements Looping

The statement looping is used to perform the same operations repeatedly. In this operation the same set of code run in repeat manner for a specific or unspecific set of time.

10. Handling Events

The Java Script has the ability to control operations updated on servers. This is basically controlling the response on the website when the user tries to perform any operation the server handled by the client like clicking on links and options, interaction response over the website, etc.

Some of its notable features include:

- a. **Interactivity:** JavaScript allows developers to add interactivity to websites, enabling users to interact with elements, perform actions, and receive immediate feedback.

- b. **DOM Manipulation:** JavaScript provides methods to access and manipulate the DOM, allowing developers to dynamically modify the content, style, and structure of web pages.
- c. **Event Handling:** JavaScript can handle various user events, such as button clicks, form submissions, mouse movements, or keyboard interactions, and execute specific actions in response.
- d. **Asynchronous Programming:** JavaScript supports asynchronous operations, enabling developers to perform tasks such as making HTTP requests to APIs and updating web page content without blocking the execution of other code.
- e. **Cross-Browser Compatibility:** JavaScript is supported by all major web browsers, ensuring broad compatibility and consistent behavior across different environments.
- f. **Extensibility:** JavaScript can be extended through the use of libraries and frameworks, providing additional functionality and simplifying common web development tasks.
- g. **Dynamic Typing:** JavaScript uses dynamic typing, allowing variables to change data types during runtime, providing flexibility and ease of use.

These features, among others, contribute to JavaScript's popularity and its ability to create interactive and dynamic web experiences.

1.2.4. Application of JavaScript:

JavaScript has a wide range of applications in web development. Some common use cases include:

As mentioned before, Javascript is one of the most widely used programming languages (Front- end as well as Back-end). It has its presence in almost every area of software development. I'm going to list few of them here:

1. **Client-side validation** - This is really important to verify any user input before submitting it to the server and Javascript plays an important role in validating those inputs at front-end itself.
2. **Manipulating HTML Pages** - Javascript helps in manipulating HTML page on the fly.

3. This helps in adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.
4. **User Notifications** - You can use Javascript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.
5. **Back-end Data Loading** - Javascript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
6. **Presentations** - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build web-based slide presentations.
7. **Server Applications** - Node JS is built on Chrome's Javascript runtime for building fast and scalable network applications. This is an event-based library which helps in developing very sophisticated server applications including Web Servers.

Advantages of JavaScript

The merits of using JavaScript are

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

- JavaScript doesn't have any multi-threading or multiprocessor capabilities.
- Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

1.2.5. Setting up the JavaScript development environment:

To set up a JavaScript development environment, you need a text editor or an Integrated Development Environment (IDE) and a web browser. **Here are the basic steps:**

1. **Install a text editor or IDE:** Choose a text editor or IDE that suits your preferences and install it on your computer. Some popular options include Visual Studio Code, Sublime Text, Atom, or WebStorm.
2. **Install a web browser:** JavaScript runs in web browsers, so you need a browser to test your code. Chrome, Firefox, Safari, and Edge are commonly used browsers for JavaScript development.
3. **Create a project directory:** Create a directory on your computer where you'll store your JavaScript files and other project assets.
4. **Create an HTML file:** In your project directory, create an HTML file with a .html extension. This file will serve as the entry point for your JavaScript code.
5. **Link your JavaScript file:** Within the HTML file, use the <script> tag to link your JavaScript file. You can either write your JavaScript code directly within the <script> tags or link an external JavaScript file using the src attribute.
6. **Start coding:** Open your HTML file in a web browser to see the changes made by your JavaScript code. You can use the browser's developer tools to debug and inspect your JavaScript code.

Here, we are going to use JavaScript in developing a web application. So, we must have at least two things, a browser, and an editor to write the JavaScript code.

Although we also need a webserver to run a web application, but we will use a single HTML web page to run our JavaScript code. So, no need to install it for now.

Browser

Mostly, you will have a browser already installed on your PC, Microsoft Edge on the Windows platform, and Safari on Mac OS.

You can also install the following browser as per your preference:

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

IDEs for JavaScript Application Development

You can write JavaScript code using a simple editor like Notepad. However, you can install any open-sourced or licensed IDE (Integrated Development Environment) to get the advantage of IntelliSense support for JavaScript and syntax error/warning highlighter for rapid development.

The followings are some of the well-known JavaScript editors:

- Visual Studio Code (Free, cross-platform)
- Eclipse (Free, cross-platform)
- Atom (Free, cross-platform)
- Notepad++ (Free, Windows)
- Code Lobster (Free, cross-platform)
- WebStorm (Paid, cross-platform)

Online Editors for JavaScript

Use the online editor to quickly execute the JavaScript code without any installation. The followings are free online editors:

- jsfiddle.net
- jsbin.com
- playcode.io

1.2.6. Method to insert scripts in HTML document:

There are a few ways to insert JavaScript code into an HTML document:

Inline script: You can include JavaScript code directly within the HTML file using the `<script>` tag. Place the `<script>` tag within the `<head>` or `<body>` section of the HTML files, and writes your JavaScript code between the opening `<script>` and closing `</script>` tags.

JavaScript string (primitive or String object) includes default properties and methods which you can use for different purposes.

For example:

The HTML script tag `<script>` is used to embed data or executable client-side scripting language in an HTML page. Mostly, JavaScript or JavaScript based API code inside a `<script></script>` tag. The following is an example of an HTML page that contains the JavaScript code in a `<script>` tag.

Example: JavaScript in a `<script>` Tag

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1> JavaScript Tutorials </h1>
    <script>
        //write JavaScript code here.
        alert('Hello, how are you?')
    </script>
</body>
</html>
```

In the above example, a `<script></script>` tag contains the JavaScript alert ('Hello, how are you?') that display a message box.

HTML v4 requires the type attribute to identify the language of script code embedded within script tag. This is specified as MIME type e.g. 'text/javascript', 'text/ecmascript', 'text/vbscript', etc.

HTML v5 page does not require the type attribute because the default script language is 'text/javascript' in a <script> tag.

An HTML page can contain multiple <script> tags in the <head> or <body> tag. The browser executes all the script tags, starting from the first script tag from the beginning.

Scripts without async, defer or type="module" attributes, as well as inline scripts, are fetched and executed immediately, before the browser continues to parse the page. Consider the following page with multiple script tags.

Example: Multiple <script> Tags

```
<!DOCTYPE html>
<html>
<head>
<script>
    alert('Executing JavaScript 1')
</script>
</head>
<body>
    <h1> JavaScript Tutorials </h1>

    <script>
        alert('Executing JavaScript 2')
    </script>

    <p>This page contains multiple script tags. </p>
```

```
<script>
    alert('Executing JavaScript 3')
</script>
</body>
</html>
```

Above, the first `<script>` tag containing `alert ('Executing JavaScript 1')` will be executed first, then `alert ('Executing JavaScript 2')` will be executed, and then `alert('Executing JavaScript 3')` will be executed.

The browser loads all the scripts included in the `<head>` tag before loading and rendering the `<body>` tag elements. So, always include JavaScript files/code in the `<head>` that are going to be used while rendering the UI. All other scripts should be placed before the ending `</body>` tag. This way, you can increase the page loading speed.

Reference the External Script File

A `<script>` tag can also be used to include an external script file to an HTML web page by using the `src` attribute.

If you don't want to write inline JavaScript code in the `<script></script>` tag, then you can also write JavaScript code in a separate file with `.js` extension and include it in a web page using `<script>` tag and reference the file via `src` attribute.

Example: JavaScript in a `<script>` Tag

```
<!DOCTYPE html>
<html>
<head>
    <script src="/MyJavaScriptFile.js" ></script>
</head>
<body>
    <h1> JavaScript Tutorials </h1>
```

```
</body>  
</html>
```

Above, the `<script src="/MyJavaScriptFile.js">` points to the external JavaScript file using the `src="/MyJavaScriptFile.js"` attribute where the value of the `src` attribute is the path or url from which a file needs to be loaded in the browser. Note that you can load the files from your domain as well as other domains.

Global Attributes

The `<script>` can contain the following global attributes:

| Attribute | Usage |
|----------------|--|
| async | <code><script async></code> executes the script asynchronously along with the rest of the page. |
| crossorigin | <code><script crossorigin="anonymous use-credentials"></code> allows error logging for sites which use a separate domain for static media. Value anonymous do not send credentials, whereas use-credentials sends the credentials. |
| defer | <code><script defer></code> executes the script after the document is parsed and before firing DOMContentLoaded event. |
| src | <code><script src="uri\path to resource"></code> specifies the URI/path of an external script file; |
| type | <code><script type="text\javascript"></code> specifies the type of the containing script e.g. text\javascript, text\html, text\plain, application\json, application\pdf, etc. |
| referrerpolicy | <code><script referrerpolicy="no-referrer"></code> specifies which referrer information to send when fetching a script. Values can be no-referrer, no-referrer-when-downgrade, origin, same-origin, strict-origin, etc. |
| integrity | <code><script integrity="sha384-oqVuAfXRKap7fdgc"></code> specifies that a user agent can use to verify that a fetched resource has been delivered free of unexpected manipulation. |

| Attribute | Usage |
|-----------|---|
| nomodule | <script nomodule> specifies that the script should not be executed in browsers supporting ES2015 modules. |

By using these methods, you can insert and execute JavaScript code within an HTML document, allowing you to enhance the functionality and interactivity of your web pages.



Dear Students! Give brief answer for the following review questions:

1. What distinguishes JavaScript from other programming languages?
 2. What are variables and how are they declared in JavaScript?
 3. How is JavaScript included in HTML files?
-

Unit Summary

JavaScript's fundamentals include variables, data types, operators, control flow, functions, objects, arrays, and error handling. You'll use these to manipulate the DOM, handle events, and even fetch data asynchronously. By mastering these concepts, you'll be able to define and use functions, including anonymous ones, work with DOM elements dynamically, and handle user interactions seamlessly.

Unit Review Questions

Identify Scripting Language

Multiple Choices

1. Which scripting language is introduced in the unit?
 - a) HTML
 - b) CSS
 - c) JavaScript (JS)
 - d) Python
2. What is the role of JavaScript in web development?
 - a) Styling web pages
 - b) Adding interactivity and dynamic behavior
 - c) Structuring web content
 - d) Managing databases
3. What are some features of JavaScript?
 - a) Cascading styles and animations
 - b) Modular programming and code reusability
 - c) Dynamic content manipulation and event handling
 - d) Database management and querying
4. What is the application of JavaScript?
 - a) Creating server-side applications
 - b) Writing operating system scripts
 - c) Developing mobile applications
 - d) Adding interactivity to web pages
5. How can you set up the JavaScript development environment?
 - a) By installing a web browser
 - b) By using a code editor with JavaScript support
 - c) By configuring server-side settings
 - d) By adding JavaScript libraries to the website

6. What is the method to insert JavaScript scripts in an HTML document?

- a) Using the <style> tag
- b) Using the <script> tag
- c) Using the <link> tag
- d) Using the <body> tag

UNIT 2

INTERACTIVITY USING JAVASCRIPT

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

- 1.1.JavaScript syntax and basic concepts
- 1.2.Control structures and loops
- 1.3.Functions in JavaScript

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ *Apply interactivity using JavaScript*

Key Terms: *JavaScript, structures, loops and Functions.*

2.1 Overview of interactivity using JavaScript

JavaScript is a programming language that executes on the browser. It turns static HTML web pages into interactive web pages by dynamically updating content, validating form data, controlling multimedia, animate images, and almost everything else on the web pages.

JavaScript is the third most important web technology after HTML and CSS. JavaScript can be used to create web and mobile applications, build web servers, create games, etc.

JavaScript can be used in various activities like data validation, displaying popup messages, handling events of HTML elements, modifying CSS, etc.

In early 1995, Brendan Eich from Netscape designed and implemented a new language for non-Java programmers to give newly added Java support in Netscape navigator. It was initially named Mocha, then LiveScript, and finally JavaScript.

2.2 JavaScript Syntax and Basic Concepts

Nowadays, JavaScript can execute not only on browsers but also on the server or any device with a JavaScript Engine. For example, Node.js is a framework based on JavaScript that executes on the server.

Learn some important characteristics of JavaScript syntax in this section.

As mentioned in the previous chapter, JavaScript code can be written inside HTML Script Tags or in a separate file with .js extension.

Write JavaScript Code

```
<script>  
    //Write javascript code here...  
</script>
```

Character Set

JavaScript uses the unicode character set, so allows almost all characters, punctuations, and symbols.

Case Sensitive

JavaScript is a case-sensitive scripting language. So, name of functions, variables and keywords are case sensitive. For example, myfunction and MyFunction are different, Name is not equal to nAme, etc.

Variables

In JavaScript, a variable is declared with or without the `var` keyword.

Example: JavaScript Statements

```
<script>  
    var name = "Abebe";  
    id = 10;
```

```
</script>
```

Semicolon

JavaScript statements are separated by a semicolon. However, it is not mandatory to end a statement with a semicolon, but it is recommended.

Example: JavaScript Statements

```
<script>
  var one = 1; two = 2; three = 3; //three different statements
  var four = 4; //single statement
  var five = "Five" //single statement without ;
</script>
```

Whitespaces

JavaScript ignores multiple spaces and tabs. The following statements are the same.

Example: Whitespaces in JavaScript

```
<script>
  var one =1;
  var one  =  1;
  var one      =      1;
</script>
```

Code Comments

A comment is single or multiple lines, which give some information about the current program. Comments are not for execution.

Write comment after double slashes // or write multiple lines of comments between /* and */

Example: Comment JavaScript Code

```
<script>
  var one =1; // this is a single line comment
```

```
/* this  
is multi line  
comment*/  
var two = 2;  
var three = 3;  
</script>
```

String

A string is a text in JavaScript. The text content must be enclosed in double or single quotation marks.

Example: String in JavaScript

```
<script>  
  var msg = "Hello World" //JavaScript string in double quotes  
  var msg = 'Hello World' //JavaScript string in single quotes  
</script>
```

Number

JavaScript allows you to work with any type of number like integer, float, hexadecimal etc.

Number must **NOT** be wrapped in quotation marks.

Example: Numbers in JavaScript

```
<script>  
  var num = 100;  
  var flot = 10.5;  
</script>
```

Boolean

As in other languages, JavaScript also includes true and false as a boolean value.

Example: Booleans in JavaScript

```
<script>
```

```
var yes = true;  
var no = false;  
</script>
```

Keywords

Keywords are reserved words in JavaScript, which cannot be used as variable names or function names.

The following table lists some of the keywords used in JavaScript.

| JavaScript Reserved Keywords | | |
|------------------------------|------------|----------|
| Var | function | if |
| Else | do | while |
| For | switch | break |
| Continue | return | try |
| Catch | finally | debugger |
| Case | class | this |
| Default | false | true |
| In | instanceOf | typeOf |
| New | null | throw |
| Void | width | delete |

2.2.1 Variables, Data Types, and Operators:

Understand how to declare and use variables of different data types (numbers, strings, booleans, objects, arrays). Master various operators for performing calculations, comparisons, and logical operations.

2.2.1.1 Variables

Variable means anything that can vary. In JavaScript, a variable stores data that can be changed later on.

Declare a Variable

In JavaScript, a variable can be declared using **var**, **let**, **const** keywords.

- **var** keyword is used to declare variables since JavaScript was created. It is confusing and error-prone when using variables declared using **var**.
- **let** keyword removes the confusion and error of **var**. It is the new and recommended way of declaring variables in JavaScript.
- **const** keyword is used to declare a constant variable that cannot be changed once assigned a value.

Here, we will use the **let** keyword to declare variables. To declare a variable, write the keyword **let** followed by the name of the variable you want to give, as shown below.

Example: Variable Declaration

```
let msg; // declaring a variable without assigning a value
```

In the above example, **var msg;** is a variable declaration. It does not have any value yet. The default value of variables that do not have any value is **undefined**.

You can assign a value to a variable using the **=** operator when you declare it or after the declaration and before accessing it.

Example: Variable Initialization

```
let msg;  
msg = "Hello JavaScript!"; // assigning a string value
```

In the above example, the **msg** variable is declared first and then assigned a string value in the next line.

You can declare a variable and assign a value to it in the same line. Values can be of any datatype such as string, numeric, boolean, etc.

Example: Variable Declaration and Initialization

```
let name = "Abebe"; //assigned string value  
let num = 100; //assigned numeric value  
let isActive = true; //assigned boolean value
```

Multiple variables can be declared in a single line, as shown below.

Example: Multiple Variables

```
let name = "Abebe", num = 100, isActive = true;
```

You can copy the value of one variable to another variable, as shown below.

Example: Copy Variable

```
let num1 = 100;  
let num2 = num1;
```

JavaScript allows multiple white spaces and line breaks when you declare a variables.

Example: Whitespace and Line Breaks

```
let name = "Abebe",  
    num = 100,  
    isActive = true;
```

Variable names are case-sensitive in JavaScript. You cannot declare a duplicate variable using the `let` keyword with the same name and case. JavaScript will throw a syntax error. Although, variables can have the same name if declared with the `var` keyword (this is why it is recommended to use `let`).

Example: Syntax Error

```
let num = 100;  
let num = 200; //syntax error
```

```
var num = 100;  
var num = 200; //Ok
```

JavaScript Variable Naming Conventions

- Variable names are case-sensitive in JavaScript. So, the variable names **msg**, **MSG**, **Msg**, **mSg** are considered separate variables.
- Variable names can contain letters, digits, or the symbols \$ and _.
- A variable name cannot start with a digit 0-9.
- A variable name cannot be a reserved keyword in JavaScript, e.g. var, function, return cannot be variable names.

Dynamic Typing

JavaScript is a loosely typed language. It means that you don't need to specify what data type a variable will contain. You can update the value of any type after initialization. It is also called dynamic typing.

Example: Loosely Typed Variable

```
let myvariable = 1; // numeric value  
myvariable = 'one'; // string value  
myvariable = 1.1; // decimal value  
myvariable = true; // Boolean value  
myvariable = null; // null value
```

Constant Variables in JavaScript

Use **const** keyword to declare a constant variable in JavaScript.

- Constant variables must be declared and initialized at the same time.
- The value of the constant variables can't be changed after initialized them.

Example: Constant Variables

```
const num = 100;  
num = 200; //error
```

```
const name; //error  
name = "Abebe";
```

The value of a constant variable cannot be changed but the content of the value can be changed. For example, if an object is assigned to a const variable then the underlying value of an object can be changed.

Example: Constant Variables

```
const person = { name: 'Abebe' };  
person.name = "Bill";  
alert(person.name); //Bill
```

It is best practice to give constant variable names in capital letters to separate them from other non-constant variables.

Variable Scope

In JavaScript, a variable can be declared either in the global scope or the local scope.

Global Variables

Variables declared out of any function are called global variables. They can be accessed anywhere in the JavaScript code, even inside any function.

Local Variables

Variables declared inside the function are called local variables of that function. They can only be accessed in the function where they are declared but not outside.

The following example includes global and local variables.

Example: Global and Local Variable

```
let greet = "Hello " // global variable  
function myfunction(){  
    let msg = "JavaScript!";
```

```
    alert(greet + msg); //can access global and local variable  
}  
  
myfunction();  
  
alert(greet); //can access global variable  
alert(msg); //error: can't access local variable
```

Learn global and local scope in JavaScript for more information.

Declare Variables without var and let Keywords

Variables can be declared and initialized without the **var** or **let** keywords. However, a value must be assigned to a variable declared without the **var** keyword.

The variables declared without the **var** keyword become global variables, irrespective of where they are declared. Visit Variable Scope in JavaScript to learn about it.

It is recommended to declare variable using the **let** keyword.

Example: Variable Declaration Without var or let Keyword

```
function myfunction(){  
    msg = "Hello JavaScript!";  
}  
  
myfunction();  
  
alert(msg); // msg becomes global variable so can be accessed here
```

Points to Remember

1. Variables can be defined using **let** keyword. Variables defined without **let** or **var** keyword become global variables.
2. Variables should be initialized before accessing it. Unassigned variable has value **undefined**.
3. JavaScript is a loosely-typed language, so a variable can store any type value.
4. Variables can have local or global scope. Local variables cannot be accessed out of the function where they are declared, whereas the global variables can be accessed from anywhere.

2.1.1.2 Comments

Single Line Comments

- Single line comments start with `//`.
- Any text between `//` and the end of the line will be ignored by JavaScript (will not be executed).

This example uses a single-line comment before each code line:

Example

```
// Change heading:  
document.getElementById("myH").innerHTML = "My First Page";  
  
// Change paragraph:  
document.getElementById("myP").innerHTML = "My first paragraph.;"
```

This example uses a single line comment at the end of each line to explain the code:

Example

```
let x = 5; // Declare x, give it the value of 5  
let y = x + 2; // Declare y, give it the value of x + 2
```

Multi-line Comments

- Multi-line comments start with `/*` and end with `*/`.
- Any text between `/*` and `*/` will be ignored by JavaScript.

This example uses a multi-line comment (a comment block) to explain the code:

Example

```
/*  
The code below will change  
the heading with id = "myH"  
and the paragraph with id = "myP"
```

in my web page:

```
/*
document.getElementById("myH").innerHTML = "My First Page";
document.getElementById("myP").innerHTML = "My first paragraph."
```

Using Comments to Prevent Execution

Using comments to prevent execution of code is suitable for code testing.

Adding `//` in front of a code line changes the code lines from an executable line to a comment.

This example uses `//` to prevent execution of one of the code lines:

Example

```
//document.getElementById("myH").innerHTML = "My First Page";
document.getElementById("myP").innerHTML = "My first paragraph.;"
```

This example uses a comment block to prevent execution of multiple lines:

Example

```
/*
document.getElementById("myH").innerHTML = "My First Page";
document.getElementById("myP").innerHTML = "My first paragraph.;"
```

2.1.1.3 Javascript Operators

JavaScript includes operators same as other languages. An operator performs some operation on single or multiple operands (data value) and produces a result. For example, in `1 + 2`, the `+` sign is an operator and `1` is left side operand and `2` is right side operand. The `+` operator performs the addition of two numeric values and returns a result. JavaScript includes following categories of operators.

1. Arithmetic Operators
2. Comparison Operators

3. Logical Operators
4. Assignment Operators
5. Conditional Operators
6. Ternary Operator

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations between numeric operands.

| Operator | Description |
|----------|--|
| + | Adds two numeric operands. |
| - | Subtract right operand from left operand |
| * | Multiply two numeric operands. |
| / | Divide left operand by right operand. |
| % | Modulus operator. Returns remainder of two operands. |
| ++ | Increment operator. Increase operand value by one. |
| -- | Decrement operator. Decrease value by one. |

The following example demonstrates how arithmetic operators perform different tasks on operands.

Example: Arithmetic Operation

```
let x = 5, y = 10;  
let z = x + y; //performs addition and returns 15  
z = y - x; //performs subtraction and returns 5  
z = x * y; //performs multiplication and returns 50  
z = y / x; //performs division and returns 2  
z = x % 2; //returns division remainder 1
```

The `++` and `--` operators are unary operators. It works with either left or right operand only. When used with the left operand, e.g., `x++`, it will increase the value of `x` when the program control goes to the next statement. In the same way, when it is used with the right operand, e.g., `++x`, it will increase the value of `x` there only. Therefore, `x++` is called post-increment, and `++x` is called pre-increment.

Example: Post and Pre Increment/Decrement

```
let x = 5;
x++; //post-increment, x will be 5 here and 6 in the next line
++x; //pre-increment, x will be 7 here
x--; //post-decrement, x will be 7 here and 6 in the next line
--x; //pre-decrement, x will be 5 here
```

String Concatenation

The `+` operator performs concatenation operation when one of the operands is of string type. The following example demonstrates string concatenation even if one of the operands is a string.

Example: + Operator with String

```
let a = 5, b = "Hello ", c = "World!", d = 10;
a + b; //returns "5Hello "
b + c; //returns "Hello World!"
a + d; //returns 15
b + true; //returns "Hello true"
c - b; //returns NaN; - operator can only used with numbers
```

Comparison Operators

JavaScript provides comparison operators that compare two operands and return a boolean value **true** or **false**.

| Operators | Description |
|-----------|-------------|
|-----------|-------------|

| Operators | Description |
|--------------------|---|
| <code>==</code> | Compares the equality of two operands without considering type. |
| <code>=====</code> | Compares equality of two operands with type. |
| <code>!=</code> | Compares inequality of two operands. |
| <code>></code> | Returns a boolean value true if the left-side value is greater than the right-side value; otherwise, returns false. |
| <code><</code> | Returns a boolean value true if the left-side value is less than the right-side value; otherwise, returns false. |
| <code>>=</code> | Returns a boolean value true if the left-side value is greater than or equal to the right-side value; otherwise, returns false. |
| <code><=</code> | Returns a boolean value true if the left-side value is less than or equal to the right-side value; otherwise, returns false. |

The following example demonstrates the comparison operators.

Example: JavaScript Comparison Operators

```
let a = 5, b = 10, c = "5";
let x = a;
a == c; // returns true
a ===== c; // returns false
a == x; // returns true
a != b; // returns true
a > b; // returns false
a < b; // returns true
a >= b; // returns false
a <= b; // returns true
```

Logical Operators

In JavaScript, the logical operators are used to combine two or more conditions. JavaScript provides the following logical operators.

| Operator | Description |
|----------|---|
| && | && is known as AND operator. It checks whether two operands are non-zero or not (0, false, undefined, null or "" are considered as zero). It returns 1 if they are non-zero; otherwise, returns 0. |
| | is known as OR operator. It checks whether any one of the two operands is non-zero or not (0, false, undefined, null or "" is considered as zero). It returns 1 if any one of them is non-zero; otherwise, returns 0. |
| ! | ! is known as NOT operator. It reverses the boolean result of the operand (or condition). !false returns true, and !true returns false. |

Example: Logical Operators

```
let a = 5, b = 10;
(a != b) && (a < b); // returns true
(a > b) || (a == b); // returns false
(a < b) || (a == b); // returns true
!(a < b); // returns false
!(a > b); // returns true
```

Assignment Operators

JavaScript provides the assignment operators to assign values to variables with less key strokes.

| Assignment operators | Description |
|----------------------|--|
| = | Assigns right operand value to the left operand. |
| += | Sums up left and right operand values and assigns the result to the left operand. |
| -= | Subtract right operand value from the left operand value and assigns the result to the left operand. |
| *= | Multiply left and right operand values and assigns the result to the left operand. |
| /= | Divide left operand value by right operand value and assign the result to the left operand. |
| %= | Get the modulus of left operand divide by right operand and assign resulted modulus to the left operand. |

Example: Assignment operators

```
let x = 5, y = 10, z = 15;
x = y; //x would be 10
x += 1; //x would be 6
x -= 1; //x would be 4
x *= 5; //x would be 25
x /= 5; //x would be 1
x %= 2; //x would be 1
```

Ternary Operator

JavaScript provides a special operator called ternary operator `:?` that assigns a value to a variable based on some condition. This is the short form of the if else condition.

Syntax:

```
<Condition>? <value1>: <value2>;
```

The ternary operator starts with conditional expression followed by the? Operator. The second part (after? and before :) will be executed if the condition turns out to be true. Suppose, the condition returns false, then the third part (after :) will be executed.

Example: Ternary operator

```
let a = 10, b = 5;
```

```
let c = a > b? a : b; // value of c would be 10
```

```
let d = a > b? b : a; // value of d would be 5
```

Remember:

1. JavaScript includes operators that perform some operation on single or multiple operands (data value) and produce a result.
2. JavaScript includes various categories of operators: Arithmetic operators, Comparison operators, Logical operators, Assignment operators, Conditional operators.
3. Ternary operator?: is a short form of if-else condition.

2.1.1.4 JavaScript Data Types

In JavaScript, you can assign different types of values (data) to a variable e.g. string, number, boolean, etc.

Example: A Variable with Different Types of Data

```
let myvariable = 1; // numeric value  
myvariable = 'one'; // string value  
myvariable = true; // Boolean value
```

In the above example, different types of values are assigned to the same variable to demonstrate the loosely typed characteristics of JavaScript. Here, 1 is the number type, 'one' is the string type, and true is the boolean type.

JavaScript includes primitive and non-primitive data types as per the latest ECMAScript 5.1 specification.

Primitive Data Types

The primitive data types are the lowest level of the data value in JavaScript. The followings are primitive data types in JavaScript:

| Data Type | Description |
|------------------|---|
| String | String is a textual content wrapped inside ' ' or " " or ` ` (tick sign). Example: 'Hello World!', "This is a string", etc. |
| Number | Number is a numeric value. Example: 100, 4521983, etc. |
| BigInt | BigInt is a numeric value in the arbitrary precision format. Example: 453889879865131n, 200n, etc. |
| Boolean | Boolean is a logical data type that has only two values, true or false. |
| Null | A null value denotes an absense of value. Example: let str = null; |
| Undefined | undefined is the default value of a variable that has not been assigned any value. Example: In the variable declaration, var str;, there is no value assigned to str. So, the type of str can be check using typeof(str) which will return undefined. |

Structural Data Types

The structural data types contain some kind of structure with primitive data.

| Data Type | Description |
|---------------|---|
| Object | An object holds multiple values in terms of properties and methods. |

| Data Type | Description |
|-----------|--|
| | Example: <pre>let person = { firstName: "Aster", lastName: "Aweke", age: 15 };</pre> |
| Date | The Date object represents date & time including days, months, years, hours, minutes, seconds, and milliseconds. Example: <code>let today = new Date("25 July 2021");</code> |
| Array | An array stores multiple values using special syntax. Example: <code>let nums = [1, 2, 3, 4];</code> |

Learn about each data type in detail in the next section.

2.1.2 Statements, Expressions, and Comments:

Differentiate between statements (performing actions) and expressions (producing values). Effectively structure your code using conditional statements (if/else), loops (for/while), and functions. Leverage comments to explain your code and enhance readability.

Example

```
let x, y, z; // Statement 1
x = 5;      // Statement 2
y = 6;      // Statement 3
z = x + y; // Statement 4
```

JavaScript Programs

A **computer program** is a list of "instructions" to be "executed" by a computer.

In a programming language, these programming instructions are called **statements**.

A **JavaScript program** is a list of programming **statements**.

In HTML, JavaScript programs are executed by the web browser.

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo":

Example

```
document.getElementById("demo").innerHTML = "Hello Dolly.;"
```

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

JavaScript programs (and JavaScript statements) are often called JavaScript code.

Semicolons;

Semicolons separate JavaScript statements.

Add a semicolon at the end of each executable statement:

Examples

```
let a, b, c; // Declare 3 variables  
a = 5;      // Assign the value 5 to a  
b = 6;      // Assign the value 6 to b  
c = a + b;  // Assign the sum of a and b to c
```

When separated by semicolons, multiple statements on one line are allowed:

```
a = 5; b = 6; c = a + b;
```

On the web, you might see examples without semicolons.

Ending statements with semicolon is not required, but highly recommended.

JavaScript White Space

JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

The following lines are equivalent:

```
let person = "Hege";  
let person="Hege";
```

A good practice is to put spaces around operators (= + - * /):

```
let x = y + z;
```

2.1.3 Type Conversions:

- Learn how to explicitly convert values between different data types using functions like parseInt() and toString().
- Understand implicit conversions and potential pitfalls.

2.3 Control Structures and Loops

2.3.1 Conditional Statements:

Apply if, else if, and else statements to control the flow of your code based on conditions. Utilize switch statements for complex decision-making with multiple options.

JavaScript includes if/else conditional statements to control the program flow, similar to other programming languages.

JavaScript includes following forms of if-else statements:

1. if Statement
2. if else Statement
3. else if Statement

if Statement

Use if conditional statement if you want to execute something based on some condition.

```
if(boolean expression)
{
    // code to be executed if condition is true
}
```

Example: if condition

```
if( 1 > 0)
{
    alert("1 is greater than 0");
}
if( 1 < 0)
{
    alert("1 is less than 0");
}
```

In the above example, the first if statement contains $1 > 0$ as conditional expression. The conditional expression $1 > 0$ will be evaluated to true, so an alert message "1 is greater than 0" will be displayed, whereas conditional expression in second if statement will be evaluated to false, so "1 is less than 0" alert message will not be displayed.

In the same way, you can use variables in a conditional expression.

Example: if condition

```
var mySal = 1000;
```

```
var yourSal = 500;  
if( mySal > yourSal)  
{  
    alert("My Salary is greater than your salary");  
}
```

Note:

curly braces { } is not required when if block contains only a single line to execute.

Use comparison operators carefully when writing conditional expression. For example, == and === is different.

Example: if condition

```
if(1=="1")  
{  
    alert("== operator does not consider types of operands");  
}  
  
if(1==="1")  
{  
    alert("===" operator considers types of operands");  
}
```

else condition

Use else statement when you want to execute the code every time when if condition evaluates to false.

The else statement must follow **if** or **else if** statement. Multiple else block is NOT allowed.

```
if(condition expression)  
{  
    //Execute this code..
```

```
}
```

```
else{
```

```
    //Execute this code..
```

```
}
```

Example: else condition

```
var mySal = 500;
```

```
var yourSal = 1000;
```

```
if( mySal > yourSal)
```

```
{
```

```
    alert("My Salary is greater than your salary");
```

```
}
```

```
else
```

```
{
```

```
    alert("My Salary is less than or equal to your salary");
```

```
}
```

else if condition

Use "else if" condition when you want to apply second level condition after if statement.

```
if(condition expression)
```

```
{
```

```
    //Execute this code block
```

```
}
```

```
else if(condition expression){
```

```
    //Execute this code block
```

```
}
```

Example: else if condition

```
var mySal = 500;
```

```
var yourSal = 1000;
```

```
if( mySal > yourSal)
```

```
{
```

```
    alert("My Salary is greater than your salary");  
}  
else if(mySal < yourSal)  
{  
    alert("My Salary is less than your salary");  
}
```

JavaScript allows multiple **else if** statements also.

Example: Multiple if else conditions

```
var mySal = 500;  
var yourSal = 1000;  
if( mySal > yourSal)  
{  
    alert("My Salary is greater than your salary");  
}  
else if(mySal < yourSal)  
{  
    alert("My Salary is less than your salary");  
}  
else if(mySal == yourSal)  
{  
    alert("My Salary is equal to your salary");  
}
```

We will learn about switch case in the next section.

Remember:

1. Use if-else conditional statements to control the program flow.
2. JavaScript includes three forms of if condition: if condition, if else condition and else if condition.

3. The if condition must have conditional expression in brackets () followed by single statement or code block wrapped with { }.
4. 'else if' statement must be placed after if condition. It can be used multiple times.
5. 'else' condition must be placed only once at the end. It must come after if or else if statement.

2.3.2 Looping Structures:

Iterate through sequences of data effectively using for, while, and do-while loops. Understand nested loops and their applications.

JavaScript includes for loop like Java or C#. Use for loop to execute code repeatedly.

```
for (initializer; condition; iteration)
{
    // Code to be executed
}
```

The for loop requires following three parts.

- Initializer: Initialize a counter variable to start with
- Condition: specify a condition that must evaluate to true for next iteration
- Iteration: increase or decrease counter

Example: for loop

```
for (var i = 0; i < 5; i++)
{
    console.log(i);
}
```

Output:

```
0 1 2 3 4
```

In the above example, var i = 0 is an initializer statement where we declare a variable **i** with value 0. The second part, **i < 5** is a condition where it checks whether i is less than 5 or not. The third part, **i++** is iteration statement where we use **++** operator to increase the value of i to 1. All these three parts are separated by semicolon;

The for loop can also be used to get the values for an array.

Example: for loop

```
var arr = [10, 11, 12, 13, 14];
for (var i = 0; i < 5; i++)
{
    console.log(arr[i]);
}
```

Output:

```
10 11 12 13 14
```

Please note that it is not mandatory to specify an initializer, condition and increment expression into bracket. You can specify initializer before starting for loop. The condition and increment statements can be included inside the block.

Example: for loop

```
var arr = [10, 11, 12, 13, 14];
var i = 0;
for (; ;) {
    if (i >= 5)
        break;
    console.log(arr[i]);
    i++;
}
```

Output:

```
10 11 12 13 14
```

Learn about while loop in the next section.

Remember:

1. JavaScript for loop is used to execute code repeatedly.
2. for loop includes three parts: initialization, condition and iteration. e.g.`for(initializer; condition; iteration){ ... }`
3. The code block can be wrapped with { } brackets.
4. An initializer can be specified before starting for loop. The condition and increment statements can be included inside the block.

2.3.3 Break and Continue Statements:

Control the flow of loops using break to exit and continue to skip iterations.

JavaScript Line Length and Line Breaks

For best readability, programmers often like to avoid code lines longer than 80 characters.

If a JavaScript statement does not fit on one line, the best place to break it is after an operator:

Example

```
document.getElementById("demo").innerHTML =  
"Hello Dolly!";
```

JavaScript Code Blocks

JavaScript statements can be grouped together in code blocks, inside curly brackets {...}.

The purpose of code blocks is to define statements to be executed together.

One place you will find statements grouped together in blocks, is in JavaScript functions:

Example

```
function myFunction() {  
    document.getElementById("demo1").innerHTML = "Hello Dolly!";  
    document.getElementById("demo2").innerHTML = "How are you?";  
}
```

In this tutorial we use 2 spaces of indentation for code blocks.

You will learn more about functions later in this tutorial.

2.3.4 Error Handling:

Implement try...catch blocks to gracefully handle runtime errors, preventing crashes and improving user experience.

JavaScript is a loosely-typed language. It does not give compile-time errors. So some times you will get a runtime error for accessing an undefined variable or calling undefined function etc.

Try catch block does not handle syntax errors.

JavaScript provides error-handling mechanism to catch runtime errors using try-catch-finally block, similar to other languages like Java or C#.

```
try  
{  
    // code that may throw an error  
}  
catch(ex)  
{  
    // code to be executed if an error occurs  
}  
finally{  
    // code to be executed regardless of an error occurs or not  
}
```

- **Try:** wrap suspicious code that may throw an error in try block.
- **Catch:** write code to do something in catch block when an error occurs. The catch block can have parameters that will give you error information. Generally catch block is used to log an error or display specific messages to the user.
- **Finally:** code in the finally block will always be executed regardless of the occurrence of an error. The finally block can be used to complete the remaining task or reset variables that might have changed before error occurred in try block.

Let's look at simple error handling examples.

Example: Error Handling in JS

```
try
{
    var result = Sum(10, 20); // Sum is not defined yet
}
catch(ex)
{
    document.getElementById("errorMessage").innerHTML = ex;
}
```

In the above example, we are calling function Sum, which is not defined yet. So, try block will throw an error which will be handled by catch block. Ex includes error message that can be displayed.

The finally block executes regardless of whatever happens.

Example: finally Block

```
try
{
    var result = Sum(10, 20); // Sum is not defined yet
}
catch(ex)
{
    document.getElementById("errorMessage").innerHTML = ex;
}
finally{
    document.getElementById("message").innerHTML = "finally block executed";
}
```

Throw

Uses throw keyword to raise a custom error.

Example: throw Error

```
try
{
    throw "Error occurred";
}
catch(ex)
{
    alert(ex);
}
```

You can use JavaScript object for more information about an error.

Example: throw error with error info

```
try
{
    throw {
        number: 101,
        message: "Error occurred"
    };
}
catch (ex) {
    alert(ex.number + "- " + ex.message);
}
```

2.4 Functions in JavaScript

Functions are the basic building block of JavaScript. Functions allow us to encapsulate a block of code and reuse it multiple times.

JavaScript Keywords

JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed.

Our Reserved Words Reference lists all JavaScript keywords.

| Keyword | Description |
|----------|---|
| Var | Declares a variable |
| Let | Declares a block variable |
| const | Declares a block constant |
| If | Marks a block of statements to be executed on a condition |
| switch | Marks a block of statements to be executed in different cases |
| For | Marks a block of statements to be executed in a loop |
| function | Declares a function |
| return | Exits a function |
| try | Implements error handling to a block of statements |

Functions make JavaScript code more readable, organized, reusable, and maintainable.

```
function <function-name>(arg1, arg2, arg3,...)
{
    //write function code here
};
```

In JavaScript, a function can be defined using the `function` keyword, followed by the name of a function and parentheses. Optionally, a list of input parameters can be included within the parentheses. The code block that needs to be executed when the function is called is written within curly braces.

2.4.1 Defining and Invoking Functions:

Write reusable blocks of code with defined functionality using the `function` keyword. Call functions to execute their code and pass necessary arguments.

The following defines a function named `greet` that will display an alert box.

Example: Define a Function

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

The above `greet()` function does not include any input parameters. It contains a single statement that displays an alert message.

Now, you can call or invoke the `greet` function by using the function name followed by the `()` operator, as shown below. When you call a function, JavaScript will execute the codes written inside the calling function.

Example: Calling a Function

```
greet();
```

2.4.2 Function Expressions and Anonymous Functions:

Create functions on the fly using function expressions. Utilize anonymous functions for concise code and event handling.

A function expression in JavaScript is a function that is stored as a value, and can be assigned to a variable or passed as an argument to another function.

Example: Function Expression

```
var add = function (num1, num2) {  
    return num1 + num2;  
};  
var result = add(10, 20); //returns 30
```

Anonymous Function

In JavaScript, you can also create anonymous functions, which are functions without a name. Anonymous functions are often used as arguments to other functions, and are

Anonymous functions are typically used in functional programming e.g. callback function, creating closure or immediately invoked function expression.

Example: Anonymous Function

```
let numbers = [10, 20, 30, 40, 50];
let squareNumbers = numbers.map(function(number) {
    return number * number;
});
```

Arrow Functions

Arrow functions are a shorthand syntax for defining anonymous functions in JavaScript. They have compact syntax compared to anonymous functions. However, they do not have their own `this` value.

Example: Arrow Function

```
let square = num => num * num;
let result = square(5);
console.log(result); //25
```

Nested Functions

In JavaScript, a function can have one or more inner functions. These nested functions are in the scope of outer function. Inner function can access variables and parameters of outer function. However, outer function cannot access variables defined inside inner functions.

Example: Nested Functions

```
function greet(firstName)
{
    function SayHello() {
        alert("Hello " + firstName);
    }
    return SayHello();
}
greet("Abebe");
```

2.4.3 Arrow Functions and Their Syntax:

Learn the modern, concise syntax of arrow functions and their benefits.

Arrow functions were introduced in ES6.

Arrow functions allow us to write shorter function syntax:

```
let myFunction = (a, b) => a * b;
```

Before Arrow:

```
hello = function() {
    return "Hello World!";
}
```

With Arrow Function:

```
hello = () => {
    return "Hello World!";
}
```

It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets *and* the `return` keyword:

Arrow Functions Return Value by Default:

```
hello = () => "Hello World!";
```

Note: This works only if the function has only one statement.

If you have parameters, you pass them inside the parentheses:

Arrow Function with Parameters:

```
hello = (val) => "Hello " + val;
```

In fact, if you have only one parameter, you can skip the parentheses as well:

Arrow Function without Parentheses:

```
hello = val => "Hello " + val;
```

Let us take a look at two examples to understand the difference. Both examples call a method twice, first when the page loads, and once again when the user clicks a button.

The first example uses a regular function, and the second example uses an arrow function.

The result shows that the first example returns two different objects (window and button), and the second example returns the window object twice, because the window object is the "owner" of the function.

Example

With a regular function `this` represents the object that *calls* the function:

```
// Regular Function:  
hello = function() {  
    document.getElementById("demo").innerHTML += this;  
}  
  
// The window object calls the function:  
window.addEventListener("load", hello);  
  
// A button object calls the function:  
document.getElementById("btn").addEventListener("click", hello);
```

Example

With an arrow function `this` represents the *owner* of the function:

```
// Arrow Function:  
hello = () => {  
    document.getElementById("demo").innerHTML += this;  
}  
  
// The window object calls the function:  
window.addEventListener("load", hello);  
  
// A button object calls the function:  
document.getElementById("btn").addEventListener("click", hello);
```

Remember these differences when you are working with functions. Sometimes the behavior of regular functions is what you want, if not, use arrow functions.

2.4.4 Parameters, Arguments, and Default Values:

Accept input data into functions using parameters and provide default values when necessary. Understand how arguments are passed and used within functions.

You can pass values to a function using parameters. A function can have one or more parameters, and the values will be passed by the calling code.

Example: Function Parameters

```
function greet(firstName, lastName) {  
    alert("Hello " + firstName + " " + lastName);  
}  
  
greet("Abebe", "Jobs");
```

JavaScript is a dynamic type scripting language, so a function parameter can have a value of any data type.

Example: Function Parameters

```
function greet(firstName, lastName) {  
    alert("Hello " + firstName + " " + lastName);  
}  
  
greet("Bill", "Gates");  
  
greet(100, 200);
```

You can pass fewer or more arguments while calling a function. If you pass fewer arguments then the rest of the parameters will become undefined. If you pass more arguments then additional arguments will be ignored.

Example: Function Parameters

```
function greet(firstName, lastName) {  
    alert("Hello " + firstName + " " + lastName);  
}  
  
greet("Abebe", "Jobs", "Mr."); // display Hello Abebe Jobs  
greet("Bill"); // display Hello Bill undefined  
greet(); // display Hello undefined undefined
```

You can also use the built-in arguments object to access parameters inside a function.

Return a Value from a Function

A function can return a value to the calling code using the `return` keyword followed by a variable or a value.

The following returns a number 10.

Example: Return a value of a Function

```
function getNumber() {  
    return 10;  
};  
  
let result = getNumber();  
console.log(result); //output: 10
```

Typically, a function returns some calculated value using parameters or an expression from a function. For example, the following `sum` function adds two parameters values using the `+` operator and returns the result of an expression.

Example: Return value from a Function

```
function Sum(num1, num2) {  
    return num1 + num2;  
};  
  
var result = Sum(10,20); // returns 30
```

A function can return another function in JavaScript.

Example: Function Returning a Function

```
function multiple(x) {  
    function fn(y)  
    {  
        return x * y;  
    }  
    return fn;  
}  
  
var triple = multiple(3);  
triple(2); // returns 6  
triple(3); // returns 9
```

In JavaScript, a string is a primitive data type that is used for textual data. JavaScript string must be enclosed in single quotes, double quotes, or backticks. The followings are string literals in JavaScript.

Example: String literals

```
"Hello World"  
'Hello World'  
`Hello World`
```

The string literal can be assigned to a variable using the equal to = operator.

Example: String Variables

```
let str1 = "This is a double quoted string.";  
let str2 = 'This is a single quoted string.';  
let str3 = `This is a template string. `;
```

The template string (using backticks) is used when you want to include the value of a variable or expressions into a string. Use \${variable or expression} inside backticks as shown below.

Example: Template String

```
let amount = 1000, rate = 0.05, duration = 3;  
let result = `Total Amount Payable: ${amount*(1 + rate*duration)}`;
```

The template string can be spanned in multiple lines which is not allowed with a single or double quoted string, as shown below.

Example: Template String

```
let str1 = `This  
    is  
    multi-line  
    string`;  
  
/*let str2 = "This  
    will  
    give  
    error"; */
```

JavaScript string can be treated like a character array. You can access a character in a string using square brackets [index] or using the str.at(pos) method.

Example: String as array

```
let str = 'Hello World';  
  
let ch1 = str[0] // H  
let ch2 = str[1] // e  
let ch3 = str.at(2) // l  
let ch4 = str.at(3) // l  
str[4] = "P"; //error
```

JavaScript strings can be accessed using a for loop, as shown below.

Example: Use for Loops

```
let str = 'Hello World';
for(let i =0; i< str.length; i++)
    console.log(str[i]);
for(let ch of str)
    console.log(ch);
```

Quotes inside String

You can include single quotes in double-quoted string or include double quotes in a single quoted string. However, you cannot include a single quote in single quoted string and double quotes in double-quoted string.

Example: Quotes in string

```
let str1 = "This is 'simple' string";
let str2 = "This is "simple" string";
let str3 = `This is 'simple' and "easy" string`;
```

If you want to include the same quotes in a string value as surrounding quotes then use a backward slash (\) before the quotation mark inside the string value.

Example: Quotes in string

```
let str1 = "This is \"simple\" string";
let str2 = 'This is \'simple\' string';
```

String Concatenation

JavaScript string can be concatenated using the + operator or string.concat() method.

Example: String concatenation

```
let str1 = 'Hello ';
let str2 = "World ";
let str3 = str1 + str2; //Hello World
let str4 = str1.concat(str2); //Hello World
```

String Objects

JavaScript allows you to create a string object using the new keyword, as shown below.

Example: String object

```
let str1 = new String(); //create string object
str1 = 'Hello World'; //assign value
// or
let str2 = new String('Hello World'); //create and assign value
```

String objects and string literals are different. The typeof() method will return the type of a variable. The following distinguished string and string objects.

Example: String object

```
let str1 = new String('Hello World');
let str2 = "Hello World";
typeof(str1); //"object"
typeof(str2); //"string"
```

Strings Comparison

Two strings can be compared using <, >, ==, === operator, and string.localeCompare(string) method.

The mathematical operators < and > compare two strings and return a boolean (true or false) based on the order of the characters in the string.

The == operator compares the content of strings and === compares the reference equality of strings. The localeCompare() method compares two strings in the current locale. It returns 0 if strings are equal, else returns 1.

Example: String Comparison

```
console.log("a" < "b"); //true
console.log("b" < "a"); //false
console.log("Apple" == "Apple"); //true
console.log("Apple" == "apple"); //false
console.log("Apple" === "Apple"); //true
console.log("Apple" === "apple"); //false
console.log("Apple".localeCompare("Apple")); //0
console.log("Apple".localeCompare("apple")); //1
```

Note that the === operator compares the reference of strings objects and not the values.

Example: String Object Comparison

```
let str1 = "Hello";
let str2 = 'Hello';
let str3 = new String('Hello');
console.log(str1 == str2); //true
console.log(str1 === str2); //true
console.log(str1 == str3); //true
console.log(str1 === str3); //false
```

JavaScript comments can be used to explain JavaScript code, and to make it more readable.

JavaScript comments can also be used to prevent execution, when testing alternative code.

2.5 DOM Manipulation and Event Handling

DOM manipulation and event handling are crucial aspects of JavaScript development. The Document Object Model (DOM) represents the structure of a web page, and through DOM manipulation, we can dynamically modify its content, style, and structure. JavaScript provides powerful functions and methods to access, create, and modify DOM elements. Event handling allows us to respond to user interactions such as clicks, mouse movements, and keystrokes. By attaching event listeners to specific elements, we can execute JavaScript code when these events occur. This combination of DOM manipulation and event handling empowers developers to create interactive and dynamic web applications.

2.6 Document Object Model (DOM):

Grasp the tree-like structure of HTML documents represented by the DOM. Navigate and access specific elements in the DOM using their IDs, classes, or names.

We have learned that a variable can hold only one value. We cannot assign multiple values to a single variable. JavaScript array is a special type of variable, which can store multiple values using a special syntax.

The following declares an array with five numeric values.

```
let numArr = [10, 20, 30, 40, 50];
```

In the above array, numArr is the name of an array variable. Multiple values are assigned to it by separating them using a comma inside square brackets as [10, 20, 30, 40, 50]. Thus, the numArr variable stores five numeric values. The numArr array is created using the literal syntax and it is the preferred way of creating arrays.

Another way of creating arrays is using the Array() constructor, as shown below.

```
let numArr = new Array(10, 20, 30, 40, 50);
```

Every value is associated with a numeric index starting with 0. The following figure illustrates how an array stores values.

JavaScript Array Representation

The following are some more examples of arrays that store different types of data.

Example: Array Literal Syntax

```
let stringArray = ["one", "two", "three"];
let numericArray = [1, 2, 3, 4];
let decimalArray = [1.1, 1.2, 1.3];
let booleanArray = [true, false, false, true];
```

It is not required to store the same type of values in an array. It can store values of different types as well.

```
let data = [1, "Abebe", "DC", true, 255000, 5.5];
```

Get Size of an Array

Use the `length` property to get the total number of elements in an array. It changes as and when you add or remove elements from the array.

Example: Get Array Size

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
console.log(cities.length); //4
cities[4] = "Delhi";
console.log(cities.length); //5
```

Accessing and Modifying DOM Elements:

Change element content, styles, attributes, and more using JavaScript methods. Create dynamic and interactive web pages by manipulating the DOM.

You can update the elements of an array at a particular index using `arrayName[index] = new_value` syntax.

Example: Update Array Elements

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
cities[0] = "Delhi";
cities[1] = "Los angeles";
console.log(cities); //["Delhi", "Los angeles", "Paris", "Sydney"]
```

Array elements (values) can be accessed using an index. Specify an index in square brackets with the array name to access the element at a particular index like `arrayName[index]`. Note that the index of an array starts from zero.

Example: Accessing Array Elements

```
let numArr = [10, 20, 30, 40, 50];
console.log(numArr[0]); // 10
console.log(numArr[1]); // 20
console.log(numArr[2]); // 30
console.log(numArr[3]); // 40
console.log(numArr[4]); // 50

let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
console.log(cities[0]); // "Addis_Ababa"
console.log(cities[1]); // "New York"
console.log(cities[2]); // "Paris"
console.log(cities[3]); // "Sydney"
//accessing element from nonexistence index
console.log(cities[4]); // undefined
```

For the new browsers, you can use the `arr.at(pos)` method to get the element from the specified index. This is the same as `arr[index]` except that the `at()` returns an element from the last element if the specified index is negative.

Example: Accessing Array using at()

```
let numArr = [10, 20, 30, 40, 50];
console.log(numArr.at(0)); // 10
console.log(numArr.at(1)); // 20
console.log(numArr.at(2)); // 30
console.log(numArr.at(3)); // 40
console.log(numArr.at(4)); // 50
console.log(numArr.at(5)); // undefined
//passing negative index
console.log(numArr.at(-1)); // 50
console.log(numArr.at(-2)); // 40
console.log(numArr.at(-3)); // 30
console.log(numArr.at(-4)); // 20
console.log(numArr.at(-5)); // 10
console.log(numArr.at(-6)); // undefined
```

You can iterate an array using `Array.forEach()`, `for`, `for-of`, and `for-in` loop, as shown below.

Example: Accessing Array Elements

```
let numArr = [10, 20, 30, 40, 50];
numArr.forEach(i => console.log(i)); //prints all elements
for(let i=0; i<numArr.length; i++)
  console.log(numArr[i]);
for(let i of numArr)
  console.log(i);
for(let i in numArr)
  console.log(numArr[i]);
```

Adding and Removing Elements Dynamically:

Programmatically add or remove elements from the DOM based on user actions or other conditions. Build complex and interactive user interfaces.

Adding New Elements

You can add new elements using `arrayName[index] = new_value` syntax. Just make sure that the index is greater than the last index. If you specify an existing index then it will update the value.

Example: Add Array Elements

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
cities[4] = "Delhi"; //add new element at last
console.log(cities); //["Addis_Ababa", "New York", "Paris", "Sydney", "Delhi"]
cities[cities.length] = "London";//use length property to specify last index
console.log(cities); //["Addis_Ababa", "New York", "Paris", "Sydney", "Delhi", "London"]
cities[9] = "Pune";
console.log(cities); //["Addis_Ababa", "New York", "Paris", "Sydney", "Delhi", "London",
undefined, undefined, undefined, "Pune"]
```

In the above example, `cities [9] = "Pune"` adds "Pune" at 9th index and all other non-declared indexes as undefined.

The recommended way of adding elements at the end is using the `push()` method. It adds an element at the end of an array.

Example: Add Element At Last using push()

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
cities.push("Delhi"); //add new element at last
console.log(cities); //["Addis_Ababa", "New York", "Paris", "Sydney", "Delhi"]
```

Use the `unshift()` method to add an element to the beginning of an array.

Example: Add Element using `unshift()`

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
cities.unshift("Delhi"); //adds new element at the beginning
console.log(cities); //["Delhi", "Addis_Ababa", "New York", "Paris", "Sydney"]
cities.unshift("London", "Pune"); //adds new element at the beginning
console.log(cities); //["London", "Pune", "Delhi", "Addis_Ababa", "New York", "Paris",
"Sydney"]
```

Remove Array Elements

The `pop()` method returns the last element and removes it from the array.

Example: Remove Last Element

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
let removedCity = cities.pop(); //returns and removes the last element
console.log(cities); //["Addis_Ababa", "New York", "Paris"]
```

The `shift()` method returns the first element and removes it from the array.

Example: Remove First Element

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
let removedCity = cities.shift(); //returns first element and removes it from array
console.log(cities); //["New York", "Paris", "Sydney"]
```

You cannot remove middle elements from an array. You will have to create a new array from an existing array without the element you do not want, as shown below.

Example: Remove Middle Elements

```
let cities = ["Addis_Ababa", "New York", "Paris", "Sydney"];
let cityToBeRemoved = "Paris";
let mycities = cities.filter(function(item) {
    return item !== cityToBeRemoved
})
console.log(mycities); //["Addis_Ababa", "New York", "Sydney"]
console.log(cities); //["Addis_Ababa", "New York", "Paris", "Sydney"]
```

Event Handling and Propagation:

Respond to user interactions (clicks, key presses, etc.) by attaching event listeners to DOM elements. Understand event bubbling and capturing for advanced event handling. A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element. To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute:

```
onclick=JavaScript
```

Examples of HTML events:

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

In this example, the content of the `<h1>` element is changed when a user clicks on it:

Example

```
<!DOCTYPE html>
<html>
<body>
<h1 onclick="this.innerHTML = 'Ooops!'">Click on this text!</h1>
</body>
</html>
```

In this example, a function is called from the event handler:

Example

```
<!DOCTYPE html>
<html>
<body>
<h1 onclick="changeText(this)">Click on this text!</h1>
<script>
function changeText(id) {
    id.innerHTML = "Ooops!";
}
</script>
</body>
</html>
```

HTML Event Attributes

To assign events to HTML elements you can use event attributes.

Example

Assign an onclick event to a button element:

```
<button onclick="displayDate()">Try it</button>
```

In the example above, a function named `displayDate` will be executed when the button is clicked.

Assign Events Using the HTML DOM

The HTML DOM allows you to assign events to HTML elements using JavaScript:

Example

Assign an onclick event to a button element:

```
<script>
document.getElementById("myBtn").onclick = displayDate;
</script>
```

In the example above, a function named `displayDate` is assigned to an HTML element with the `id="myBtn"`. The function will be executed when the button is clicked.

The onload and onunload Events

The `onload` and `onunload` events are triggered when the user enters or leaves the page.

The `onload` event can be used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

The `onload` and `onunload` events can be used to deal with cookies.

Example

```
<body onload="checkCookies()">
```

The `oninput` Event

The `oninput` event is often used to some action while the user input data. Below is an example of how to use the `oninput` to change the content of an input field.

Example

```
<input type="text" id="fname" oninput="upperCase()">
```

The `onchange` Event

The `onchange` event is often used in combination with validation of input fields. Below is an example of how to use the `onchange`. The `upperCase()` function will be called when a user changes the content of an input field.

Example

```
<input type="text" id="fname" onchange="upperCase()">
```



Self-check 2-1:

1. What are the basic data types in JavaScript?
2. How do you declare and initialize a variable in JavaScript?
3. What are JavaScript operators? Provide examples of arithmetic, comparison, and logical operators.
4. How do you use if, else if, and else statements in JavaScript?
5. What is a function in JavaScript?
6. How do you declare a function in JavaScript? Provide examples of both function declarations and function expressions.
7. Explain the concept of parameters and arguments in JavaScript functions.

Unit Summary

Unit Two focuses on applying interactivity to web pages using JavaScript. It covers essential concepts and techniques that enable trainees to enhance user experience and add dynamic functionality to their web applications. The unit covers the following content:

- JavaScript syntax and basic concepts: Trainees learn the fundamentals of JavaScript syntax, including variables, data types, operators, and control structures like conditionals and loops.
- Control structures and loops: Trainees explore different control structures, such as if-else statements and switch statements, and learn how to use loops like for, while, and do-while to control program flow.
- Functions in JavaScript: Trainees gain knowledge of defining and invoking functions, understanding function expressions and anonymous functions, and working with parameters, arguments, and default values.
- DOM manipulation and event handling: Trainees delve into the Document Object Model (DOM) and learn how to access and modify DOM elements using JavaScript. They also discover how to dynamically add and remove elements. Additionally, trainees gain an understanding of event handling and event propagation, enabling them to create interactive features.

Unit Review Questions

Here are some overview questions for Unit Two: Apply Interactivity Using JavaScript:

1. What is the syntax of JavaScript, and what are some basic concepts that you should be familiar with?
2. How do control structures and loops work in JavaScript, and what are their uses?
3. What are functions in JavaScript, and how do you define and invoke them?
4. Explain function expressions and anonymous functions in JavaScript.
5. What are parameters, arguments, and default values in JavaScript functions?

6. What is the Document Object Model (DOM) in JavaScript, and how does it relate to web page elements?
7. How do you access and modify DOM elements using JavaScript?
8. Can you explain how to dynamically add and remove elements in the DOM using JavaScript?
9. What is event handling in JavaScript, and how does it facilitate interactivity?
10. How does event propagation work in JavaScript, and what are its implications for handling events?

Multiple choice

1. What are the topics covered in the unit related to JavaScript syntax and basic concepts?
 - a) Variables, data types, and operators
 - b) Statements, expressions, and comments
 - c) Type conversions
 - d) All of the above
2. What is the purpose of control structures and loops in JavaScript?
 - a) Manipulating the Document Object Model (DOM)
 - b) Adding interactivity to web pages
 - c) Handling errors and exceptions
 - d) Defining and invoking functions
3. What are conditional statements used for in JavaScript?
 - a) Modifying the Document Object Model (DOM)
 - b) Controlling the flow of program execution based on conditions
 - c) Accessing and modifying DOM elements
 - d) Defining and invoking functions
4. What is the purpose of error handling in JavaScript?
 - a) Modifying the Document Object Model (DOM)
 - b) Adding interactivity to web pages
 - c) Handling errors and exceptions in code execution
 - d) Accessing and modifying DOM elements

5. What is the purpose of functions in JavaScript?
 - a) Defining and invoking functions
 - b) Accessing and modifying DOM elements
 - c) Controlling the flow of program execution
 - d) Handling errors and exceptions
6. What is the Document Object Model (DOM) in JavaScript?
 - a) A programming language used for server-side development
 - b) A model that represents the structure of an HTML document
 - c) A library for accessing and modifying databases
 - d) A tool for debugging JavaScript code
7. How can you access and modify DOM elements in JavaScript?
 - a) By using CSS selectors
 - b) By using JavaScript methods and properties
 - c) By using jQuery library functions
 - d) By using HTML tags directly

UNIT 3

VALIDATING USER INPUT

Unit Coverage

This unit is designed to provide you the necessary information and practice regarding the following content coverage:

3.1. Validation

Unit Learning Outcomes

This unit will also assist you to attain the following learning outcomes. Specifically, upon completion of this unit, you will be able to:

- ✓ Apply Validation
- ✓ Apply Form validation

Key Terms: *Validation, Form Validation and web form.*

3.1. Validating User Input

Dear learners! *"What is form validation and why is it important in web applications?"*

Form validation is a crucial aspect of user input validation in web applications. It ensures that the data entered by users in a form meets the required criteria and is valid before it is processed or stored. By validating user input, you can prevent errors, improve data quality, enhance security, and provide a better user experience.

Here are some details about form validation for trainees:

3.1.1 Introduction to Validation:

Validation is the process of checking if user input conforms to specific rules or constraints.

- It helps ensure that the data entered is accurate, complete, and appropriate for the intended purpose.

- Validation can be performed on various types of user input, such as text fields, checkboxes, radio buttons, dropdowns, and file uploads.

3.1.2 Form Validation:

Form validation in JavaScript is a technique used to ensure that user input in web forms meets certain criteria or requirements before it is submitted to the server. It helps validate the data entered by the user and prevents the submission of incorrect or incomplete form data.

Here's an **example 1:-** of how you can perform form validation in JavaScript:

HTML:

html

```
<form id="myForm" onsubmit="validateForm(event)">
  <label for="name">Name:</label>
  <input type="text" id="name" required>
  <label for="email">Email:</label>
  <input type="email" id="email" required>
  <input type="submit" value="Submit">
</form>
```

JavaScript: javascript

```
function validateForm(event) {
  event.preventDefault(); // Prevent form submission
  // Get form inputs
  var nameInput = document.getElementById("name");
  var emailInput = document.getElementById("email");
  // Validate name
  if (nameInput.value === "") {
    alert("Please enter your name.");
    return false; // Stop form submission
  }
  // Validate email
  if (emailInput.value === "") {
    alert("Please enter your email.");
```

```
    return false; // Stop form submission
}

// Additional validation logic for email format, password requirements, etc.

// Form is valid, submit it

document.getElementById("myForm").submit();
}
```

In this example, the validateForm function is called when the form is submitted. It first prevents the default form submission using event.preventDefault(). Then, it retrieves the input values from the form fields and performs the desired validation checks.

If any validation condition fails, an alert is shown with an appropriate error message, and return false stops the form submission. If all validation conditions pass, the form is submitted using document.getElementById("myForm").submit().

Example 2: Form validation in JavaScript using HTML5 involves implementing validation rules on form fields to ensure that user input meets specific criteria before it is submitted. Here's an example of how you can perform form validation using HTML5 attributes and JavaScript:

1. HTML Form Structure:

html

```
<form id="myForm" action="submitForm.php" method="POST">

<label for="name">Name:</label>
<input type="text" id="name" name="name" required>

<label for="email">Email:</label>
<input type="email" id="email" name="email" required>

<label for="password">Password:</label>
<input type="password" id="password" name="password" required>

<input type="submit" value="Submit">
</form>
```

In this example, we have a simple form with three fields: name, email, and password. The "required" attribute is added to ensure that these fields are mandatory.

2. JavaScript Validation:

Javascript

```
var form = document.getElementById("myForm");

form.addEventListener("submit", function(event) {
  if (!form.checkValidity()) {
    event.preventDefault(); // Prevent form submission if validation fails
  }
});
```

In this JavaScript code, we get a reference to the form element using its ID. We then add an event listener for the form's "submit" event. Inside the event handler function, we check if the form is valid using the `checkValidity()` method. If the form is not valid, we use `event.preventDefault()` to prevent the form submission.

With this setup, HTML5's built-in form validation will handle basic validation, such as checking if required fields are filled and if the email field has a valid email format. Additionally, you can use JavaScript to implement custom validation logic by checking specific conditions or patterns using regular expressions or other techniques.

It's important to note that HTML5 form validation provides a convenient way to perform basic client-side validation. However, server-side validation should always be implemented to ensure data integrity and security.

Example 3: Here's an additional example of form validation in JavaScript using HTML5:

HTML Form Structure:

html

```
<form id="myForm" action="submitForm.php" method="POST">  
    <label for="username">Username:</label>  
    <input type="text" id="username" name="username" required minlength="4" maxlength="10">  
  
    <label for="password">Password:</label>  
    <input type="password" id="password" name="password" required minlength="6">  
  
    <label for="confirmPassword">Confirm Password:</label>  
    <input type="password" id="confirmPassword" name="confirmPassword" required>  
  
    <input type="submit" value="Submit">  
</form>
```

In this example, we have a form with three fields: username, password, and confirmPassword. The "required" attribute is added to ensure that these fields are mandatory. Additionally, we use the "minlength" and "maxlength" attributes to specify the minimum and maximum length for the username field and the minimum length for the password field.

JavaScript Validation:

javascript

```
var form = document.getElementById("myForm");  
var password = document.getElementById("password");  
var confirmPassword = document.getElementById("confirmPassword");  
form.addEventListener("submit", function(event) {  
    if (!form.checkValidity()) {  
        event.preventDefault(); // Prevent form submission if validation fails  
    } else if (password.value !== confirmPassword.value) {
```

```
event.preventDefault(); // Prevent form submission if password and confirm password do not  
match  
alert("Passwords do not match!");  
}  
});
```

In this JavaScript code, we get references to the form element, the password field, and the confirmPassword field using their respective IDs. We add an event listener for the form's "submit" event. Inside the event handler function, we first check if the form is valid using the checkValidity() method. If the form is not valid, we use event.preventDefault() to prevent the form submission.

Additionally, we check if the entered passwords match by comparing the values of the password and confirmPassword fields. If they don't match, we prevent the form submission and display an alert message to the user.

This example demonstrates how to implement custom validation logic using JavaScript in addition to HTML5's built-in form validation. By combining both approaches, you can create robust form validation to ensure that user input meets specific criteria before submitting the form.

You can add more validation logic based on your specific requirements, such as checking for the correct email format, password strength, or validating specific fields.

Trainees can practice implementing form validation using HTML, CSS, JavaScript, and server-side languages/frameworks like PHP, Python (Django/Flask), Ruby (Ruby on Rails), or .NET (ASP.NET) depending on their learning objectives and the technologies used in their training environment. They can start with simple form validation scenarios and gradually progress to more complex validation requirements based on real-world examples. It's also beneficial to explore existing libraries and frameworks that provide validation utilities to streamline the development process.

**Self-check 3-1:**

-
1. What is the purpose of form validation in web applications?
 2. What are the two main types of validation in form validation?
-

Unit Summary

Form validation is a critical process in web application development that ensures user input meets specific criteria before processing or storing it. By validating user input, developers can improve data accuracy, prevent errors, enhance security, and provide a better user experience. Form validation can be performed using client-side or server-side techniques, or a combination of both. Client-side validation, carried out using JavaScript or HTML5 validation attributes, provides immediate feedback to users. Server-side validation, performed on the server after form submission, offers a more robust and secure validation process.

Common validation techniques include checking for required fields, validating data formats, ensuring numeric or alphanumeric inputs, and implementing custom validation based on business rules. Trainees should understand the importance of form validation, potential security concerns, and various validation techniques to effectively implement validation in web applications.

Unit Review Questions

1. What is the purpose of validation?
 - a) To secure user input
 - b) To ensure accurate data entry
 - c) To prevent submission of incorrect or incomplete data
 - d) All of the above

2. What is form validation?
 - a) Verifying the authenticity of a web form
 - b) Checking the data entered by the user against predefined rules
 - c) Encrypting form data before submission
 - d) Converting form data into JSON format
3. Which of the following is true about validation?
 - a) It can be performed on the client-side using JavaScript
 - b) It can be performed on the server-side using a backend language
 - c) Both a) and b)
 - d) None of the above
4. How can form validation be implemented in JavaScript?
 - a) By using if-else statements to check input values
 - b) By defining rules for each form field and validating them
 - c) By using regular expressions to match patterns in input
 - d) All of the above
5. What are some common validation checks that can be performed on user input?

| | |
|---------------------------------|--------------------------------|
| a) Checking for required fields | c) Verifying password strength |
| b) Validating email addresses | d) All of the above |
6. What is the purpose of preventing form submission in validation?

| | |
|--|---|
| a) To display error messages to the user | c) To redirect the user to another page |
| b) To clear the form fields | d) None of the above |

Project work

Project Overview: first select one organization and based on business logic, Develop an interactive website for students that provides a user-friendly interface, helpful resources, and interactive features to enhance their learning experience. Task should perform as follows:

1. **Requirements Gathering:** Identify the required features and functionalities.
2. **Planning and Design:** Create a simple sitemap to outline the website's structure and navigation and planning
3. **Development:** including both back end and front end
4. **Testing and Quality Assurance:** Conduct testing to ensure the website functions correctly across different browsers and devices. Verify that interactive features, forms, and data submission work properly. Address any bugs or errors discovered during testing.

RB: Students should submit and present for the class and graded after completing Grade 11 courses

Answers for Self-Check Questions

Answers for Self-check 1-1

1. Distinguishing JavaScript from other programming languages:

- JavaScript is primarily used for scripting web pages and runs on the client side within a web browser. It's known for its asynchronous programming model, which allows for dynamic updates without reloading the entire page. Unlike languages like Java or Python, JavaScript's syntax and execution model are tailored for manipulating web page elements and handling events.

□2. Variables in JavaScript:

- Variables in JavaScript are containers for storing data values. They are declared using var, let, or const keywords followed by the variable name. Example:

javascript

Copy code

// Using var

```
var count = 10;
```

// Using let (preferred for block-scoped variables)

```
let name = 'Alice';
```

// Using const (for constants)

```
const PI = 3.14;
```

□ 3. Including JavaScript in HTML files:

- JavaScript can be included in HTML files using the <script> tag. It can be placed in the <head> or <body> section of the HTML document. Example:

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Example</title>
    <script src="script.js"></script> <!-- External JavaScript file -->
</head>
<body>
    <h1>Hello, world!</h1>
    <script>
        // Inline JavaScript code
        console.log('This is inline JavaScript');
    </script>
</body>
</html>
```

- JavaScript can also be embedded directly within HTML using `<script>` tags, either inline or by linking to an external .js file using the `src` attribute.

Answers for Self-check 2-1

1. Basic data types in JavaScript:

- JavaScript has primitive data types such as number, string, boolean, null, undefined, as well as object and symbol.

2. Declaring and initializing a variable in JavaScript:

- Using `var`, `let`, or `const` followed by the variable name and optionally initializing it with a value.

javascript
Copy code
// Using var
var age = 30;

// Using let (block-scoped)

```
let name = 'John';  
  
// Using const (immutable)  
const PI = 3.14;
```

3. JavaScript operators:

- **Arithmetic operators:** + (addition), - (subtraction), * (multiplication), / (division), % (remainder).
- **Comparison operators:** == (equal to), != (not equal to), === (strict equal to), !== (strict not equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to).
- **Logical operators:** && (logical AND), || (logical OR), ! (logical NOT).

```
javascript  
Copy code  
// Examples  
  
let x = 5;  
let y = 3;  
  
// Arithmetic  
  
let sum = x + y; // 8  
  
// Comparison  
console.log(x > y); // true  
  
// Logical  
let condition = (x > 0) && (y < 0); // false
```

4. Using if, else if, and else statements:

- Conditional statements are used to execute different blocks of code based on different conditions.

```
javascript  
Copy code  
let num = 10;  
  
if (num > 0) {  
    console.log('Number is positive');
```

```
    } else if (num < 0) {
        console.log('Number is negative');
    } else {
        console.log('Number is zero');
    }
```

5. Function in JavaScript:

- A function in JavaScript is a block of reusable code designed to perform a particular task.

6. Declaring a function in JavaScript:

- **Function Declaration:**

```
javascript
Copy code
function greet(name) {
    console.log(`Hello, ${name}!`);
}
greet('Alice'); // Hello, Alice!
```

- **Function Expression:**

```
javascript
Copy code
let square = function(x) {
    return x * x;
};
console.log(square(5)); // 25
```

7. Parameters and arguments in JavaScript functions:

- **Parameters** are variables listed as part of the function definition.
- **Arguments** are the values passed to the function when it is invoked.

```
javascript
Copy code
function add(a, b) { // a and b are parameters
    return a + b;
}
```

```
let result = add(3, 5); // 3 and 5 are arguments
console.log(result); // 8
```

Answers for Self-check 3-1

1. Purpose of form validation in web applications:

- Form validation ensures that user input submitted through web forms is correct, complete, and meets specified criteria before it is processed or sent to a server. It helps prevent erroneous or malicious data from being submitted, improves data quality, and enhances user experience by providing feedback to users on how to correct input errors.

2. Two main types of validation in form validation:

- **Client-side validation:** Validation that occurs in the user's browser using JavaScript or HTML5 attributes. It provides immediate feedback to users but can be bypassed or manipulated by users.
- **Server-side validation:** Validation that occurs on the server after the form data has been submitted. It ensures data integrity and security, as it cannot be bypassed by users.

References

1. Introduction to JavaScript (JS): Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media. Retrieved from [https://www.oreilly.com/library/view/javascript-the-good/9780596517748/JavaScript_Events_Examples_\(w3schools.com\)](https://www.oreilly.com/library/view/javascript-the-good/9780596517748/JavaScript_Events_Examples_(w3schools.com))
2. JavaScript Syntax and Basic Concepts: Haverbeke, M. (2018). Eloquent JavaScript: A Modern Introduction to Programming. No Starch Press. Retrieved from <https://eloquentjavascript.net/> for loop in JavaScript (tutorialsteacher.com)
3. Control Structures and Loops: Flanagan, D. (2011). JavaScript: The Definitive Guide. O'Reilly Media. Retrieved from <https://www.oreilly.com/library/view/javascript-the-definitive/9780596805531/>
4. Functions in JavaScript: Simpson, K. (2015). You Don't Know JS: Up & Going. O'Reilly Media. Retrieved from <https://github.com/getify/You-Dont-Know-JS/tree/2nd-ed/up%20%26%20going>
5. DOM Manipulation and Event Handling: Keith, J. (2005). DOM Scripting: Web Design with JavaScript and the Document Object Model. Apress. Retrieved from <https://www.apress.com/gp/book/9781590595336>
6. Validating User Input: Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media (Chapter 8). Retrieved from <https://www.oreilly.com/library/view/javascript-the-good/9780596517748/>

Module VII

Basic Kaizen

Contents

| No | Contents | page |
|----|--|------|
| | Module Description | 664 |
| 1 | Unit 1: Basic Kaizen Concepts | 665 |
| | 1.1 Introduction to Kaizen | 665 |
| | 1.2 Key Concepts of Continuous Improvement | 669 |
| | 1.3 Benefits of Implementing Kaizen: | 682 |
| | Unit Summary | 684 |
| | Unit Review Questions | 684 |
| 2 | Unit 2: Identify and Eliminate Waste | 689 |
| | 2.1 Introduction to Waste | 689 |
| | 2.2 Types of Waste | 690 |
| | 2.3 Understanding Waste in the Workplace | 695 |
| | 2.4 Identifying and Eliminating Muda | 701 |
| | 2.5 Continuous Improvement Mindset in Waste Reduction | 704 |
| | Unit Summary | 706 |
| | Unit Review Questions | 706 |
| 3 | Unit 3: Basic Kaizen | 709 |
| | 3.1 Meaning of 5S | 709 |
| | 3.2 Benefit of Implementing 5S | 711 |
| | 3.3 The 5S Phases | 711 |
| | Unit Summary | 717 |
| | Unit Review Questions | 717 |
| | References | 721 |

Module Description

This module equips students with the necessary knowledge, skills and mindset to understand and apply Kaizen principles effectively. It covers basic principles and importance of Kaizen, waste elimination, and the 5S methodology for workplace efficiency. Through practical exercises, students learn to identify and eradicate waste, ultimately preparing them to drive continuous improvement in future endeavors.

Module Instruction:

Learning Instructions: How to use this Module

For effectively use this module you are expected to follow the following module instructions:

1. Read the learning outcomes of this module.
2. Learn study lessons in the module. Try to understand what is being discussed.
3. Accomplish the “Self-checks” which are placed following each topic. Then you are to get the answer key at the end of the module to correct your answer only after you have finished answering the Self-checks.
4. Accomplish unit review questions and practical activities which are placed at the end of each unit. Then ask from your teacher/trainer the key to correction (answers key) or you can request your teacher/trainer to correct your work.

UNIT 1

BASIC KAIZEN CONCEPTS

Unit Learning Outcomes

At the end of this unit, students will be able to:

- Understand meaning of Kaizen and its basic concept
- Understand origin of Kaizen
- Performing basic Kaizen Principles
- Recognize the benefit of Kaizen

Key terms: *Kaizen, PDCA, Improvement*

Unit Overview

Kaizen, originating from Japan, embodies the philosophy of continuous improvement. It emphasizes the relentless pursuit of small, incremental changes in processes, systems, and behaviors to achieve greater efficiency, quality, and effectiveness. The term "Kaizen" itself translates to "change" (kai) for the better (zen), reflecting its core principle of ongoing improvement. At its essence, Kaizen is not a one-time event or a grand overhaul but rather a daily practice ingrained in the culture of an organization. It involves the collective efforts of all employees, from top management to frontline workers, who are encouraged to identify problems, propose solutions, and implement improvements in their respective areas of work. Kaizen fosters a culture of innovation, empowerment, and continuous learning, where even the smallest improvements contribute to significant long-term gains. By embracing Kaizen, organizations strive to stay agile, adaptive, and competitive in a rapidly evolving world.

1.1. Introduction to Kaizen

Kaizen is a Japanese term that translates to "change for better" or "continuous improvement." It represents a philosophy and a methodology focused on incremental improvements in processes, products, or services, involving all employees from the CEO to the frontline workers. Originating in Japan, Kaizen gained widespread recognition through its application in manufacturing, particularly at Toyota, and has since been adopted across various industries, including software development, healthcare, and service industries. It is a philosophy of continuous improvement that emphasizes

small, incremental changes involving all employees to enhance processes, increase efficiency, and improve product quality. When applied to website development or production, Kaizen can significantly streamline workflows, reduce waste, and create a more productive and collaborative environment.

Implementing Kaizen in website development promotes a culture of continuous improvement, engaging all team members in the process of making small, meaningful changes that enhance efficiency, quality, and user satisfaction. By focusing on incremental improvements and standardizing best practices, teams can achieve significant long-term benefits and maintain a competitive edge.

1.1.1. The meaning of Kaizen

Kaizen is a Japanese term that translates to "continuous improvement" or "change for the better." It's a philosophy or methodology focused on making incremental improvements in processes, products, or services over time. Rather than seeking major, revolutionary changes, kaizen emphasizes small, gradual improvements that can be implemented by everyone in an organization. This approach encourages employees to constantly look for ways to streamline processes, reduce waste, and enhance quality, resulting in overall efficiency gains and a culture of continuous improvement within the organization. Kaizen is widely used in various industries around the world as a key element of lean manufacturing and management practices. Kaizen is continuous improvement that is based on certain guiding principles:

1.1.2. Origin of Kaizen

Kaizen traces its roots back to post-World War II Japan, a time when the country faced economic devastation and reconstruction challenges. In this context, Japanese industries sought innovative approaches to rebuild their economy swiftly and sustainably. It was during this period that the foundations of Kaizen were laid by visionary leaders and forward-thinking organizations, notably Toyota. Toyota's production system, often referred to as the Toyota Production System (TPS), became synonymous with Kaizen principles. TPS revolutionized manufacturing by introducing concepts such as Just-in-Time (JIT) production, Total Quality Management (TQM), and continuous improvement. These practices enabled Toyota to achieve exceptional levels of efficiency, quality, and flexibility, setting a benchmark for industries worldwide. Over time, the success of TPS and

Kaizen philosophy transcended borders, inspiring organizations across diverse sectors and geographies to adopt similar principles. Today, Kaizen has evolved into a globally recognized management philosophy and a cornerstone of operational excellence, driving innovation and improvement across industries and disciplines.

1.1.3. Principles of Kaizen philosophy

The principles of Kaizen philosophy encompass a set of fundamental beliefs and values that guide organizations in their pursuit of continuous improvement, drive operational excellence and achieve long-term success in today's dynamic and competitive business landscape.

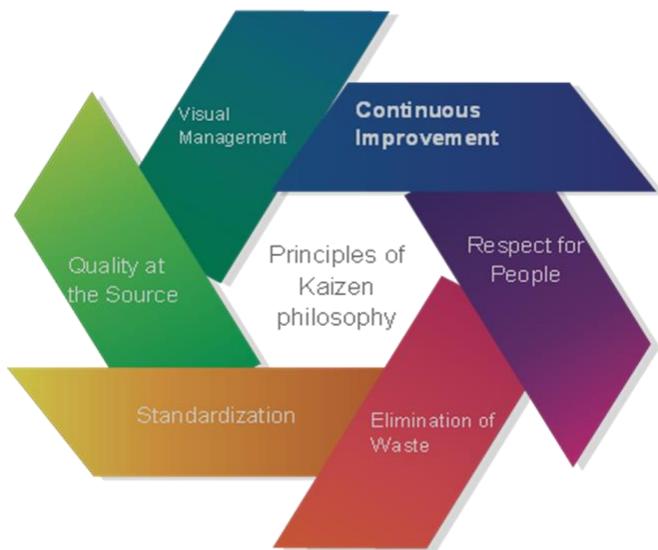


Figure 1.1: Key principles of kaizen

These principles form the cornerstone of Kaizen implementation and are essential for fostering a culture of innovation, collaboration, and sustained growth. Here are the key principles of Kaizen:

- **Continuous Improvement:** Kaizen emphasizes the concept of continuous, incremental improvement in all aspects of operations, processes, and systems. It encourages organizations to constantly seek opportunities for enhancement, no matter how small, to achieve greater efficiency, quality, and customer satisfaction over time.
- **Respect for People:** Central to Kaizen is the belief that every individual within the organization possesses valuable insights, knowledge, and expertise. Therefore, Kaizen

promotes a culture of mutual respect, trust, and empowerment, where employees are actively engaged, encouraged to voice their ideas, and given the autonomy to implement improvements.

- **Elimination of Waste:** Kaizen advocates for the identification and elimination of waste in all forms, including overproduction, waiting times, unnecessary motion, defects, and excess inventory. By reducing waste, organizations can streamline processes, optimize resource utilization, and enhance overall efficiency and productivity.
- **Standardization:** Standardization is essential for ensuring consistency, reliability, and repeatability in processes and operations. Kaizen encourages the establishment of standardized work procedures, protocols, and guidelines to maintain quality, reduce variability, and facilitate continuous improvement efforts.
- **Quality at the Source:** Kaizen promotes the principle of "quality at the source," which means identifying and addressing quality issues at their origin rather than detecting and correcting defects downstream. By instilling a culture of accountability and responsibility for quality among all employees, organizations can prevent errors, defects, and rework, thereby improving product and service quality.
- **Visual Management:** Visual management techniques, such as visual controls, signage, and displays, play a crucial role in Kaizen implementation by making information, processes, and performance indicators easily accessible and understandable to employees. Visual management tools help create transparency, facilitate communication, and support continuous monitoring and improvement efforts.



Self-check 1-1:

1. Describe the significance of continuous improvement in the context of Kaizen philosophy. How does it contribute to organizational success?
2. Identify and discuss three key principles of Kaizen philosophy outlined in the text. How do these principles contribute to fostering a culture of continuous improvement within organizations?

-
3. How does Kaizen promote employee empowerment and engagement in the improvement process? Provide examples of how employees can actively participate in Kaizen initiatives.
-

1.2. Key Concepts of Continuous Improvement

Continuous Improvement (CI) is an ongoing effort to improve products, services, or processes incrementally over time. It is a fundamental principle in various management philosophies, such as Total Quality Management (TQM) and Lean Management. Key concepts of continuous improvement include: Incremental changes vs. radical improvements, The PDCA Cycle, Employee involvement and empowerment, and Quality Circles and teamwork.

1.2.1. Incremental Changes vs. Radical Improvements (Kaizen)

1.2.1.1. Incremental Changes

Incremental changes involve making small, continuous improvements to a website over time. This approach aligns closely with the Kaizen philosophy of continuous improvement, where minor enhancements accumulate to create significant benefits.

Advantages of Incremental Changes

- Reduced Risk: Small changes are less likely to disrupt the website's overall functionality and user experience.
- Continuous Feedback: Frequent updates allow for regular feedback from users, enabling ongoing refinement and adjustment.
- Easier Implementation: Smaller changes are typically easier to implement and test, requiring less time and resources.
- Sustainable Progress: Gradual improvements help maintain a steady pace of development and innovation.
- Quick Wins: Achieving small, quick wins can boost team morale and demonstrate progress to stakeholders.

For examples of Incremental Changes in Website Design and Development

- Performance Optimization: Regularly optimizing images, scripts, and server configurations to improve load times.
- UI/UX Tweaks: Making minor adjustments to the layout, color scheme, or typography to enhance user experience.

- Content Updates: Frequently updating and refining content to keep it relevant and engaging.
- Feature Enhancements: Gradually adding new features or improving existing ones based on user feedback and analytics.
- SEO Adjustments: Continually updating SEO practices to align with the latest search engine algorithms and trends.

Example 1: Incremental Changes based on optimization

|  | Steps: Performance Optimization |
|---|--|
| | <p>Step 1: Identify slow-loading pages using analytics tools.</p> <p>Step 2: Optimize images on one page to reduce load time.</p> <p>Step 3: Measure the impact and roll out similar optimizations across other pages.</p> <p>Step 4: Continuously monitor and refine based on ongoing performance data.</p> |

Example 2: Incremental Changes based on user interface

|  | Steps: UI/UX Improvements |
|--|--|
| | <p>Step 1: Collect user feedback indicating navigation issues.</p> <p>Step 2: Test a new navigation menu on a small segment of users.</p> <p>Step 3: Analyze user interaction data to assess improvement.</p> <p>Step 4: Implement the new navigation site-wide if successful.</p> |

1.2.1.2. Radical Improvements

Radical improvements involve making substantial, often transformative changes to a website. This approach can lead to significant leaps in performance, functionality, or user experience but carries higher risks and requires more resources.

Advantages of Radical Improvements

- Significant Impact: Major changes can dramatically enhance the website's capabilities and user experience.
- Competitive Advantage: Large-scale improvements can differentiate a website from competitors, providing a significant edge.

- Innovation: Radical changes can drive innovation, introducing new technologies or approaches that revolutionize the website.
- Addressing Major Issues: Comprehensive changes can address fundamental issues that incremental improvements might not fully resolve.

Examples of Radical Improvements in Website Design and Development

- Complete Redesign: Overhauling the entire website's design and structure to modernize its appearance and improve usability.
- Platform Migration: Moving the website to a new platform or content management system (CMS) to enhance performance and scalability.
- Feature Overhaul: Introducing major new features or reworking existing ones to offer significantly improved functionality.
- Brand Refresh: Revamping the website to align with a new brand identity, including changes to the logo, color scheme, and messaging.
- Architecture Changes: Redesigning the underlying architecture, such as adopting a new framework or microservices architecture, to improve performance and scalability.

Example: Radical Improvements based on redesign of Website

|  | Steps: Complete Redesign |
|---|---|
| | <p>Step 1: Conduct user research and competitive analysis to inform the new design.</p> <p>Step 2: Develop wireframes and prototypes for the redesigned website.</p> <p>Step 3: Implement the redesign in a staging environment.</p> <p>Step 4: Test the redesign with a select group of users, gather feedback, and make necessary adjustments.</p> <p>Step 5: Launch the redesigned website and monitor user feedback and performance metrics for further refinements.</p> |

1.2.1.3. Changes vs Radical improvements

Table 1-1: compare and contrast incremental changes and Radical improvements

| criteria | Kaizen (Incremental Changes) | Innovation (Radical Improvements) |
|------------------------|--|--|
| Effect | Long term not breakthrough | Short-term breakthrough |
| Steps | Small steps | Big steps |
| teamwork | Permanente action with gradual rising effects | Incidental action with immediate effect |
| Change | gradual and permanent | Sudden and single |
| engagement | All | Chosen leaders |
| Approach | Team effect, process approach | Individual ideas and actions |
| Work method | Maintenance and improvement | Extinguishing and rebuild |
| Ideas | Conventional knowhow and traditional technology | Usage of technology, breakthrough, new innovation and theory |
| Practical requirements | Small investment, big effort | Big investment, small effort |
| Orientation | For people | For technology |
| Assessment criteria | Process and engagement in achieving of better result | Result directly affect on profits |

1.2.2. The PDCA Cycle: A Blueprint for Continuous Improvement

The Plan-Do-Check-Act Cycle (PDCA Cycle) is a four-step model for systematic problem solving and continuous improvement. It offers a simple and structured way for resolving business-related

issues and creating positive change. This framework is widely recognized as the basis for enhancing the quality of processes, products, and services by following a logical sequence of four steps: Plan, Do, Check, and Act.

The above steps represent the PDCA cycle in action

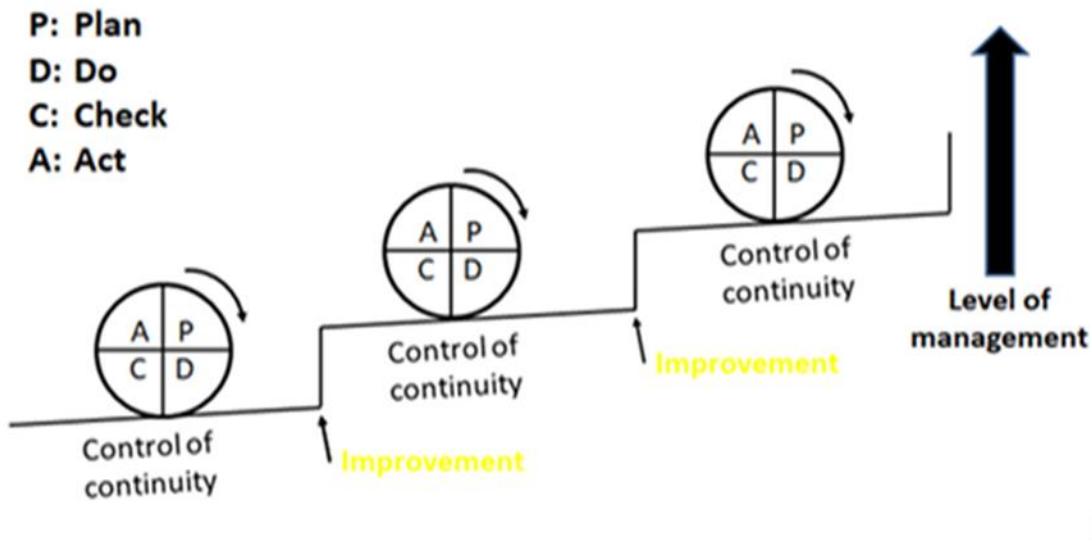


Figure 1.2: The PDCA Cycle Model

The Four Phases of the PDCA Cycle

The PDCA cycle begins with the Planning phase which involves the identification of the problem and objectives. Continuous Improvement is applying the PDCA cycle to website development and improvement ensures a systematic approach to enhancing performance, user experience, and overall effectiveness. By continuously planning, doing, checking, and acting, web development teams can make data-driven decisions that lead to sustained and meaningful improvements.

1.2.2.1. Plan-Do-Check-Act (PDCA) Cycle for Website Development and Improvement

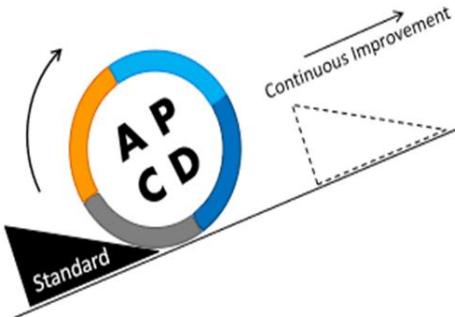


Figure 1.3: PDCA and Standardization

Kaizen, can be effectively applied to website development to enhance the quality, performance, and user experience of a website. Here's a basic guide on how to implement Kaizen principles in website development:

1. Plan (Identify and Plan Improvements)

- Goal Setting: Define clear, measurable goals for your website (e.g., increased user engagement, reduced bounce rate, faster load times).
- Benchmarking: Assess the current performance of your website using analytics tools to identify areas that need improvement.
- User Feedback: Gather feedback from users through surveys, usability tests, and direct comments to understand their needs and pain points.

In the planning phase, the goal is to identify opportunities for improvement and develop a detailed plan of action.

| | |
|--|--|
| | Steps: <ol style="list-style-type: none"> 1. Identify Areas for Improvement: Use analytics, user feedback, and performance data to pinpoint issues or areas that need enhancement. Examples include slow page load times, high bounce rates, or low conversion rates. 2. Set Objectives: Define clear, measurable goals for the improvement. For instance, reducing page load times by 20%, increasing conversion rates by 15%, or improving user |
|--|--|

- engagement by 10%.
3. **Develop Hypotheses:** Formulate hypotheses about what changes might lead to the desired improvements. For example, "Reducing the number of form fields will increase the registration completion rate."
 4. **Create an Action Plan:** Outline the steps needed to implement the changes, assign responsibilities, and set a timeline. This might include designing new elements, writing code, or setting up A/B tests.

Example:

- Objective: Increase the conversion rate of the landing page by 20%.
- Hypothesis: Simplifying the call-to-action (CTA) and reducing the number of form fields will lead to higher conversion rates.
- Action Plan: Redesign the landing page with a simplified CTA and a shorter form. Set up A/B testing to compare the new design with the current one.

2. Do (Implement Changes)

- Small Changes: Implement small, incremental changes rather than large, disruptive ones. This can include tweaking the layout, updating content, or optimizing images.
- A/B Testing: Test changes through A/B testing to see which variations perform better.
- Code Reviews: Regularly review code to ensure it is clean, efficient, and follows best practices.

In the doing phase, you implement the changes outlined in the plan on a small scale.

|  | Steps: |
|---|--|
| | <ol style="list-style-type: none">1. Implement Changes: Make the changes as per the action plan. This could involve updating the website design, modifying content, optimizing code, etc.2. Develop Test Variants: For A/B testing, create different versions (variants) of the webpage or element to test against the control.3. Deploy Changes: Launch the changes on a subset of users or in a controlled environment. Ensure that the changes are tracked for later analysis. |

Example:

- Implement the new landing page design.
- Set up A/B testing with the current landing page as the control (Version A) and the new design as the variant (Version B).
- Ensure analytics are in place to track user interactions and conversions.

3. Check (Evaluate the Results)

- Performance Monitoring: Use analytics tools (like Google Analytics) to track the impact of the changes on your website's performance.
- User Feedback: Collect feedback post-implementation to see if the changes have positively affected the user experience.
- Error Tracking: Use error monitoring tools to track and fix any issues that arise after the changes.

In the checking phase, you analyze the results of the changes to see if they met the objectives.

|  | Steps: |
|---|---|
| | <ol style="list-style-type: none">1. Collect Data: Use web analytics tools (e.g., Google Analytics) to gather data on user behavior, performance metrics, and conversion rates.2. Analyze Results: Compare the performance of the test variants against the control. Assess whether the changes led to the desired improvements.3. Review Feedback: Gather qualitative feedback from users, if available, to understand their experience with the changes. |

Example:

- Data Collection: Monitor metrics such as conversion rates, bounce rates, and user engagement for both the control and variant landing pages.
- Analysis: Compare the data to see if the new design improved the conversion rate by the targeted 20%.

- Feedback Review: Check for user feedback through surveys or direct comments to understand their experience with the new landing page.

4. Act (Standardize and Optimize)

- Standardize Successes: If a change proves successful, standardize it across the website.
- Document Processes: Keep thorough documentation of changes and processes for future reference and to maintain consistency.
- Iterate: Continuously look for new areas to improve, repeating the Kaizen cycle.

In the acting phase, you decide on the next steps based on the results of the check phase.

|  | Steps: |
|---|---|
| | <ol style="list-style-type: none">1. Implement Successful Changes: If the changes achieved the desired outcomes, roll them out on a larger scale or permanently.2. Iterate or Adjust: If the changes did not meet expectations, use the insights gained to adjust the approach and run another PDCA cycle. Refine the hypotheses and action plan based on the findings. <p>Document and Share Learnings: Document the process, results, and lessons learned. Share these with the team to inform future improvement efforts</p> |

Example:

- Success: If the new landing page variant significantly increased conversion rates, implement it permanently across the site.
- Iteration: If the results were inconclusive or did not meet the target, analyze what might have gone wrong. Perhaps the form reduction was not sufficient, or the CTA needs further refinement. Plan a new PDCA cycle to test these new hypotheses.
- Documentation: Record the results, what worked, what didn't, and why. Share this with the team to guide future improvements.

1.2.2.2. Continuous PDCA Cycle

The PDCA cycle is not a one-time process but a continuous loop. Each cycle builds on the previous one, fostering a culture of ongoing improvement.

Example of Ongoing PDCA Cycles:

- **Cycle 1:** Improve landing page conversion rates.
 - Plan: Identify issues, set objectives, hypothesize changes.
 - Do: Implement changes and run A/B tests.
 - Check: Analyze results and user feedback.
 - Act: Implement successful changes or iterate.
- **Cycle 2:** Enhance page load speed.
 - Plan: Identify slow-loading pages, set speed improvement goals.
 - Do: Optimize images, leverage browser caching, and minimize code.
 - Check: Measure load times before and after changes.
 - Act: Apply successful optimizations site-wide or continue refining.
- **Cycle 3:** Boost user engagement.
 - Plan: Identify engagement metrics, set improvement targets.
 - Do: Add interactive elements, improve content relevance.
 - Check: Track user interactions and engagement metrics.
 - Act: Implement successful strategies or refine based on data.

1.2.3. Employee involvement and empowerment

Continuous improvement relies heavily on the participation and engagement of employees at all levels of the organization. Employees are encouraged to identify problems, suggest solutions, and actively participate in improvement initiatives. Empowerment involves giving employees the authority, autonomy, and resources to make decisions and implement changes in their work areas. This fosters a culture of ownership and accountability for improvement.

1.2.4. Quality Circles and teamwork

Quality circles are small groups of employees who voluntarily come together to identify, analyze, and solve work-related problems within their area of expertise. These circles promote teamwork, collaboration, and shared responsibility for quality and improvement. They provide a forum for

employees to exchange ideas, share best practices, and contribute to organizational goals. Continuous improvement is not a one-time initiative but rather a continuous journey towards excellence. By embracing these key concepts and methodologies, organizations can foster a culture of innovation, efficiency, and quality that drives sustainable growth and competitiveness.



Figure 1.4: Quality Circle

1.2.4.1. Quality Circles or Website Development (Kaizen)

In the context of website development, these circles can be highly effective in fostering continuous improvement and addressing specific challenges.

Key Concepts of Quality Circles in Web Development

- Voluntary Participation: Team members participate voluntarily, promoting engagement and ownership of the improvement process.
- Regular Meetings: Circles meet regularly to identify problems, brainstorm solutions, and implement changes.
- Diverse Perspectives: Include team members from different disciplines (developers, designers, content creators, marketers) to bring varied perspectives and expertise.
- Focus on Improvement: The primary goal is to improve processes, enhance quality, and increase efficiency.
- Empowerment: Empower team members to suggest and implement changes, fostering a culture of continuous improvement.

Steps to Implement Quality Circles in Web Development

1. Formation: Create small groups (4-10 members) within the web development team.
2. Training: Provide training on problem-solving techniques, data analysis, and Kaizen principles.
3. Regular Meetings: Schedule regular meetings (weekly or bi-weekly) for Quality Circles to discuss issues and propose solutions.
4. Problem Identification: Use tools like root cause analysis, Pareto charts, and fishbone diagrams to identify and analyze problems.
5. Solution Development: Brainstorm and develop potential solutions, prioritizing those that can be implemented quickly with minimal resources.
6. Implementation: Implement the agreed-upon solutions and monitor their effectiveness.
7. Review and Feedback: Review the outcomes, gather feedback, and refine the solutions as necessary.

Example of Quality Circles in Action

1. **Issue Identification:** A Quality Circle identifies that users frequently abandon their shopping carts during the checkout process.
2. **Analysis:** The team uses analytics to understand the abandonment points and gathers user feedback through surveys.
3. **Solution Development:** They brainstorm ideas such as simplifying the checkout process, reducing the number of steps, and offering guest checkout options.
4. **Implementation:** The team implements a simplified checkout process on a test basis.
5. **Review:** They monitor the impact on cart abandonment rates and gather further user feedback.
6. **Refinement:** Based on the results, they make additional tweaks to optimize the process.

1.2.5. Teamwork in Kaizen for Website Development

Effective teamwork is essential for implementing Kaizen in website development. It involves collaboration, communication, and a shared commitment to continuous improvement.

Key Elements of Teamwork in Kaizen

1. **Cross-Functional Teams:** Assemble teams with members from different disciplines to leverage diverse skills and perspectives.
2. **Open Communication:** Encourage open and transparent communication to facilitate idea sharing and problem-solving.
3. **Shared Goals:** Define and align team goals with the overall objectives of the website development project.
4. **Collaborative Culture:** Foster a culture of collaboration where team members support each other and work together towards common goals.
5. **Continuous Learning:** Promote continuous learning and skill development to keep the team updated with the latest trends and technologies.

Steps to Foster Teamwork in Kaizen

1. **Define Roles and Responsibilities:** Clearly define each team member's role and responsibilities to ensure accountability and effective collaboration.
2. **Set Common Goals:** Establish common goals that align with the project objectives and motivate the team to work together.
3. **Encourage Participation:** Create an environment where all team members feel valued and encouraged to contribute their ideas.
4. **Provide Tools and Resources:** Equip the team with the necessary tools and resources to collaborate effectively, such as project management software, communication platforms, and development tools.
5. **Celebrate Successes:** Recognize and celebrate the team's achievements to build morale and reinforce the importance of collaboration.
6. **Continuous Feedback:** Implement regular feedback sessions to address any issues, provide constructive feedback, and continuously improve team dynamics.

Example of Effective Teamwork in Kaizen

1. **Project Kickoff:** A new project is initiated to improve the website's mobile performance.
2. **Cross-Functional Team:** A team is formed with developers, designers, and UX specialists.
3. **Goal Setting:** The team sets a common goal to reduce mobile load times by 30% within three months.

4. **Collaborative Planning:** They hold planning sessions to outline tasks, assign responsibilities, and establish milestones.
5. **Implementation:** The team collaborates to optimize images, streamline code, and enhance mobile usability.
6. **Feedback and Iteration:** Regular feedback sessions are held to assess progress, address challenges, and iterate on the improvements.
7. **Celebration:** Upon achieving the goal, the team's success is celebrated, reinforcing the value of teamwork and collaboration.

1.3. Benefits of Implementing Kaizen in Web Design and Development

Implementing Kaizen, a Japanese philosophy of continuous improvement, offers a range of benefits to organizations.

- Benefits of Kaizen Enhanced User Experience: Continuous improvements lead to a better user experience and higher customer satisfaction.
- Increased Efficiency: Streamlined processes and elimination of waste improve efficiency and reduce development time.
- Higher Quality: Ongoing quality control ensures that the website remains robust, secure, and high-performing.
- Employee Engagement: Involving employees in the improvement process fosters a culture of collaboration and innovation.
- Adaptability: A commitment to continuous improvement allows the website to adapt to changing user needs and technological advancements.

Applying the Kaizen philosophy to website design and development fosters a culture of continuous improvement, leading to better user experiences, higher efficiency, and sustained success. By focusing on incremental changes and involving all team members in the process, organizations can create dynamic, high-quality websites that evolve with the needs of their users and the market.



Self-check 1-2:

-
1. What are the main differences between incremental changes and radical improvements in the context of continuous improvement?
 2. Describe the four phases of the PDCA Cycle briefly.
 3. How does employee involvement contribute to continuous improvement efforts within an organization?
 4. What are quality circles, and how do they support teamwork and improvement initiatives?
 5. Name two benefits of implementing Kaizen related to increased productivity and efficiency.
-

Examples of Successful Kaizen Practices

Case 1. Toyota Production System (TPS) Toyota is renowned for its implementation of Kaizen principles in its production processes. TPS emphasizes continuous improvement, employee empowerment, and waste reduction. For example, Toyota implemented "Just-in-Time" production, where parts are only ordered and used when needed, reducing inventory costs and improving efficiency.

Case 2. Amazon applies Kaizen principles to enhance its operations and customer service. Key practices include: Data-Driven Decision Making: Using data and metrics to continuously monitor and improve processes, Automation: Implementing automation in warehouses to improve efficiency and reduce errors and Customer Feedback Loops: Actively collecting and analyzing customer feedback to drive product and service improvements. Amazon continually applies Kaizen principles to its website development, ensuring a seamless and optimized user experience. Key practices include: Continuous A/B Testing: Amazon constantly runs A/B tests to optimize various elements of its website, from page layout to checkout processes. These tests involve small changes to see which versions perform better in terms of user engagement and conversions and Incremental Performance Improvements: Amazon's development team focuses on reducing load times and improving site speed through incremental optimizations such as image compression, code magnification, and efficient use of caching.

Unit Summary

Kaizen is a Japanese philosophy that emphasizes continuous improvement through small, incremental changes. Originating in post-World War II Japan, it has been adopted by industries like Toyota. Kaizen is built on several key principles, including continuous improvement, respect for people, waste elimination, standardization, and quality at the source. Employee involvement and empowerment are crucial for continuous improvement, fostering responsibility and accountability. The PDCA cycle, a four-step model for systematic problem-solving, enhances the quality of processes, products, and services. Quality circles, small groups of employees, promote teamwork, engagement, and continuous improvement. By embracing Kaizen principles and using tools like the Kaizen board and PDCA cycle, organizations can create a culture of continuous improvement that benefits both employees and customers. Overall, Kaizen is a powerful tool for organizations to improve efficiency, quality, and effectiveness by promoting continuous improvement and employee involvement.

Unit Review Questions

Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What is the primary emphasis of Kaizen philosophy?
 - A) Large-scale changes
 - B) Continuous improvement
 - C) Rapid production
 - D) Employee turnover
2. Which principle of Kaizen focuses on empowering employees and fostering responsibility?
 - A) Waste elimination
 - B) Quality at the source
 - C) Respect for people
 - D) Standardization

3. What is one of the key tools used in Kaizen for systematic problem-solving?
 - A) PDCA cycle
 - B) Six Sigma
 - C) Kanban
 - D) SWOT analysis
4. How do quality circles contribute to Kaizen philosophy?
 - A) By increasing production speed
 - B) By promoting teamwork and continuous improvement
 - C) By reducing employee engagement
 - D) By eliminating waste
5. Which industry is often credited with popularizing Kaizen philosophy?
 - A) Automotive (e.g., Toyota)
 - B) Software development
 - C) Fashion
 - D) Agriculture
6. What is a key benefit of using the PDCA cycle in Kaizen?
 - A) Reduces employee involvement
 - B) Enhances quality and efficiency
 - C) Decreases customer satisfaction
 - D) Increases waste
7. What role does waste elimination play in Kaizen?
 - A) Decreases productivity
 - B) Increases costs
 - C) Improves efficiency and effectiveness
 - D) Reduces employee satisfaction
8. How does Kaizen promote quality at the source?
 - A) By increasing defects
 - B) By empowering employees to identify and solve problems
 - C) By reducing employee involvement
 - D) By implementing strict regulations

9. Which tool is used to visually manage improvement projects in Kaizen?
 - A) SWOT analysis
 - B) Kanban board
 - C) Six Sigma
 - D) ISO 9001
10. What is a primary goal of implementing Kaizen principles in organizations?
 - A) Decreasing efficiency
 - B) Increasing waste
 - C) Improving efficiency, quality, and effectiveness
 - D) Reducing employee satisfaction

Answer key for Self-check questions

Self-check 1-1: Answer Key

1. Significance of Continuous Improvement in Kaizen Philosophy:

- Continuous improvement in Kaizen focuses on ongoing, incremental enhancements, leading to increased efficiency, reduced waste, and improved quality. This approach helps organizations adapt swiftly to market changes, enhancing customer satisfaction and providing a competitive edge.

2. Three Key Principles of Kaizen Philosophy:

- **Customer Focus:** Ensures all improvements add value to the customer, enhancing satisfaction and loyalty.
- **Incremental Improvements:** Promotes small, manageable changes, fostering a habit of regular progress and reducing resistance to change.
- **Employee Involvement:** Engages all employees in the improvement process, leveraging their insights for more effective solutions and fostering a sense of ownership.

3. Kaizen Promotes Employee Empowerment and Engagement:

- **Empowerment:** Employees are given the authority to identify and solve problems, boosting their confidence and job satisfaction.
- **Engagement:** Active participation in decision-making and problem-solving makes employees feel valued.

Examples of Participation: Suggestion Systems, Quality Circles and Kaizen Events

Self-check 1-2: Answer Key

1. Differences Between Incremental Changes and Radical Improvements:

- **Incremental Changes:** These are small, gradual adjustments made regularly to improve processes, products, or services. They focus on optimizing existing systems and methods, resulting in steady progress and minimal disruption.
- **Radical Improvements:** These involve significant, often disruptive changes that fundamentally alter processes, products, or services. They aim for breakthrough advancements and substantial improvements in performance, efficiency, or innovation.

2. Four Phases of the PDCA Cycle:

- **Plan:** Identify a problem or opportunity for improvement, analyze it, and develop a plan to address it, including setting objectives and processes necessary to deliver results.
- **Do:** Implement the plan on a small scale to test its effectiveness. This phase involves executing the planned actions and collecting data for analysis.
- **Check:** Review and evaluate the results of the implementation. Compare the outcomes against the objectives and analyze any deviations or unexpected results.
- **Act:** Based on the evaluation, decide whether to adopt, modify, or abandon the plan. If successful, implement the changes on a larger scale. If not, refine the plan and repeat the cycle.

3. Employee Involvement in Continuous Improvement:

- Employee involvement is crucial for continuous improvement as it leverages the knowledge, skills, and insights of those directly involved in the processes. Employees can identify problems, suggest solutions, and implement changes effectively. Their engagement fosters a culture of ownership, accountability, and proactive problem-solving, leading to sustained improvement and innovation.

4. Quality Circles:

- Quality circles are small groups of employees who regularly meet to identify, analyze, and solve work-related problems. They promote teamwork by encouraging collaborative problem-solving and sharing of ideas. Quality circles support improvement initiatives by fostering a culture of continuous learning and mutual support, leading to enhanced productivity and quality.

5. Benefits of Implementing Kaizen:

- **Increased Productivity:** Kaizen focuses on continuous, incremental improvements in processes, which streamline workflows, reduce waste, and enhance efficiency. This results in higher productivity and more effective use of resources.
- **Enhanced Efficiency:** By involving all employees in identifying and solving problems, Kaizen creates a culture of ongoing improvement. This collective effort leads to optimized processes, reduced downtime, and better resource utilization, contributing to overall organizational efficiency.

UNIT 2

WASTE IDENTIFICATION AND ELIMINATION

Unit Learning Outcomes

At the end of this unit, students will be able to:

- Understand the concept waste
- Identify various types of waste
- Develop strategies to eliminate waste

Key terms: Waste / Muda

2.1. Introduction to Waste

Muda, a term in Japanese, refers to waste that doesn't add value to the final product or service from the customer's perspective. In a Kaizen culture, waste is anything that consumes resources, increases costs, and doesn't contribute value to the customer. Waste analysis is a core principle of kaizen thinking, involving the identification, quantification, elimination, and prevention of waste. Waste can take various forms and is universally applicable across various industries. The Eight Wastes model is a widely adopted framework, categorizing waste into eight forms, making it easier to identify and prioritize actions.

2.1.1. What is Muda/waste?

Waste refers to actions or steps in a process that do not add value to the customer. The original seven wastes (Muda) was developed by Taiichi Ohno, with the Eight Wastes model being widely used today. There are two types of wastes: obvious and hidden. Obvious wastes take the shape of an iceberg, with the visible bulk containing hidden wastes. According to website development, waste can manifest in various forms, such as unnecessary features, output and input, inefficient processes, and redundant tasks. It can be in the form of materials, stocks, equipment, facilities, manhours, utilities, documents, expenses, motion, and other activities that do not add value.



Self-check 2-1:

-
1. Define waste according to the context of Kaizen philosophy.
 2. Explain the difference between obvious wastes and hidden wastes.

2.2 Types of Waste in Website Development

Waste is one of the biggest enemies of manufacturing efficiency and profitability. It can increase your costs, lower your quality, and reduce your customer satisfaction. **Muda** is a Japanese word that means waste or futility, and it refers to any activity that does not add value to the product or service. According to the lean philosophy, there are seven types of Muda that you need to eliminate or minimize in manufacturing/Service processes.

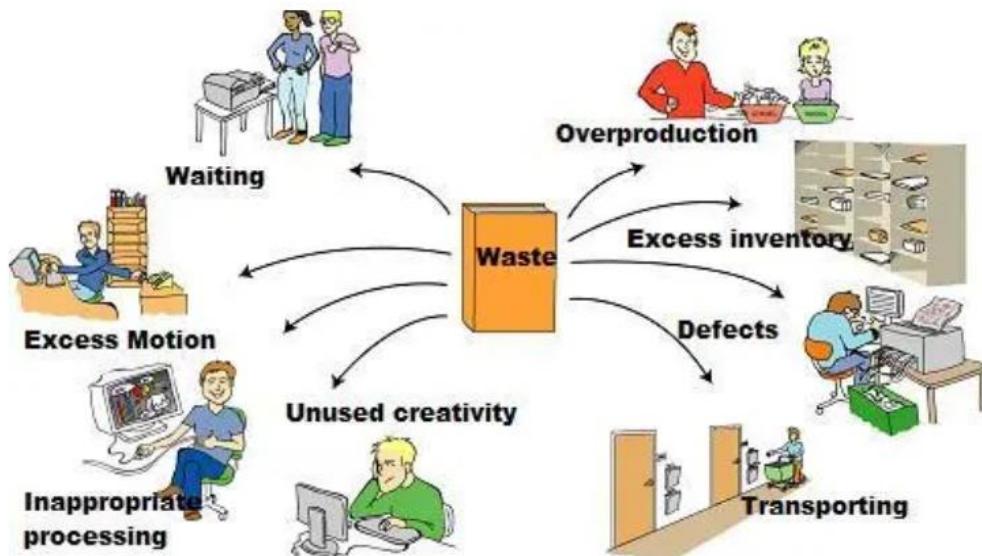


Figure 2.1: The 8 wastes/Muda

Identifying and eliminating waste in website development is essential for improving efficiency, reducing costs, and delivering higher quality products. In this topic, we will explain the 8 types of waste with each their some practical examples, causes and Improvement Ideas. Following are types of Waste in Website Development:

1. **Overproduction:** Creating more features or content than necessary or before they are needed. It producing greater quantities or making more of something than what is actually demanded or required by the customer.

Examples:

- Developing multiple features simultaneously without user demand. In other words, developing complex functionalities that users do not use or creating extensive content that is not frequently accessed.
- Creating extensive content that is rarely accessed by users.

Causes:

- Lack of user research and validation.
- Pressure to add more features for competitive advantage without clear user need.

Improvement Ideas:

- Conduct thorough user research to understand user needs and prioritize features accordingly.
- Implement a Minimum Viable Product (MVP) approach to release only essential features initially and expand based on user feedback.
- Use Agile methodologies to develop features incrementally and validate their value before full-scale development.

2. **Waiting:** Idle time when developers, designers, or other team members are waiting for inputs, approvals, or resources.

Examples:

- Developers waiting for design approvals or content inputs. Whereas developers waiting for design mockups or content creators waiting for feedback from stakeholders.
- Delays in deployment due to dependency on external teams or resources.

Causes:

- Inefficient workflow and approval processes.
- Dependency on external resources or teams.

Improvement Ideas:

- Streamline approval processes by defining clear guidelines and responsibilities.
- Implement parallel workflows where possible, allowing different teams to work concurrently.
- Use Agile sprints to ensure regular and timely delivery of inputs and approvals.

3. **Overprocessing:** Performing more work or adding more features than what is required by the users.

Examples:

- Adding excessive animations or overly complex design elements that do not enhance user experience.
- Implementing redundant functionalities that are rarely used.

Causes:

- Misunderstanding user needs and preferences.
- Lack of clear requirements and scope definition.

Improvement Ideas:

- Simplify design and functionality to focus on user experience and essential features.
- Conduct user testing to validate the necessity and impact of design elements and features.
- Regularly review and refine the project scope to avoid feature creep.

4. Defects: Errors and bugs that lead to rework and corrections.**Examples:**

- Code Bugs and errors in the code, broken links, or incorrect content that needs to be fixed after deployment or require rework
- Broken links or incorrect content that need correction.

Causes:

- Inadequate testing and quality assurance processes.
- Rushed development cycles leading to oversight and errors.

Improvement Ideas:

- Integrate automated testing and continuous integration practices to catch defects early.
- Allocate sufficient time for thorough testing and quality assurance.
- Foster a culture of code reviews and peer feedback to identify and fix issues before deployment.

5. Unused Talent: Not fully utilizing the skills and capabilities of team members.**Examples:**

- Developers only coding and not contributing to design or UX improvements where they have expertise. A developer with UX skills being limited to backend coding without contributing to UI/UX improvements.
- Team members with valuable skills being underutilized in their roles.

Causes:

- Rigid role definitions and lack of cross-functional opportunities.
- Poor understanding of team members' full skill sets.

Improvement Ideas:

- Encourage cross-functional roles and collaboration to leverage the full range of team members' skills.
- Conduct regular skill assessments and training to utilize untapped potential.
- Promote a culture of continuous learning and skill development.

6. Transportation: Unnecessary movement of information or materials.**Examples:**

- Inefficient handoffs of design files between designers and developers. Inefficient handoffs between teams, such as moving design files back and forth between designers and developers
- Moving data between multiple systems or platforms unnecessarily.

Causes:

- Lack of integrated tools and platforms for collaboration.
- Poorly defined workflow processes.

Improvement Ideas:

- Adopt integrated collaboration tools that streamline file sharing and communication.
- Define clear workflow processes to minimize unnecessary handoffs and data transfers.
- Standardize tools and platforms across teams to reduce friction.

7. Inventory: Excess materials or work-in-progress that are not immediately needed.**Examples:**

- Accumulating a large backlog of features and tasks without clear prioritization. Stockpiling unimplemented features or accumulating backlog items without a clear implementation plan.
- Stockpiling unimplemented design elements or content.

Causes:

- Poor backlog management and prioritization practices.
- Lack of regular review and refinement of tasks and features.

Improvement Ideas:

- Implement regular backlog grooming sessions to prioritize and refine tasks.
- Use Kanban or similar visual management tools to manage and limit work-in-progress.
- Focus on delivering high-value features first and regularly reassess the need for remaining items.

8. **Motion:** Unnecessary movements or actions by people that do not add value.

Examples:

- Developers spending excessive time searching for documentation or code snippets.
Developers searching for information in poorly organized documentation or navigating through a cluttered codebase.
- Repeatedly switching between different tools and platforms.

Causes:

- Poor organization of documentation and resources.
- Lack of standardization in tools and processes.

Improvement Ideas:

- Organize documentation and code repositories in a searchable and accessible manner.
- Standardize tools and platforms across teams to reduce unnecessary switching.
- Implement best practices for documentation and code management to facilitate easy access.

**Self-check 2-2:**

1. What is Muda, and why is it crucial in both manufacturing and service sectors?
 2. Provide examples of overproduction waste in both manufacturing and service industries, and discuss its negative impacts.
 3. How does excess inventory affect both manufacturing and service processes, and what strategies can be employed to minimize it?
 4. Define transportation waste in the context of both manufacturing and service, and suggest practical approaches to mitigate it.
 5. Explain waiting-time waste and its implications in both manufacturing and service operations, along with methods for effective reduction.
-
-

2.3 Understanding Waste in the Workplace

2.3.1 Value-added vs. non-value-added activities:

- Value-added activities: These are activities that directly contribute to meeting customer needs and requirements. They enhance the quality, functionality, or features of the final product or service and are perceived as valuable by the customer.
- Non-value-added activities: Non-value-added activities are those that do not contribute to meeting customer needs and requirements. They include tasks, processes, or steps that add no value to the final product or service and are considered wasteful. Identifying and eliminating non-value-added activities is essential for improving efficiency and reducing costs in the workplace.

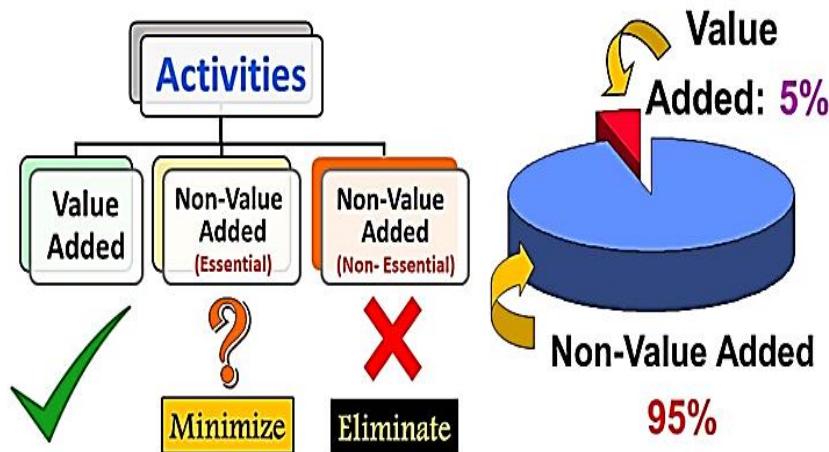


Figure 2.2: Value adding vs non-value adding activities

2.3.2 Sources and Effects of Waste in Production Processes:

Waste in production processes can arise from various sources, including overproduction, waiting times, transportation inefficiencies, unnecessary motion, over processing, excess inventory, and defects or errors in products or services. Eliminating these sources of Muda is the primary goal of lean manufacturing to improve efficiency, quality and productivity.

The 5M+S framework is a useful tool for identifying sources of Muda (waste) in production processes. In website development, waste can stem from various sources, often categorized under the 5M+S framework. The 5M stands for: an, Machine, Material, Method, Measurement, and

Surroundings. Identifying these sources is crucial for optimizing processes and enhancing efficiency.

1. **Man (People):-**Underutilized talent, lack of training, inefficient team collaboration, Untrained workers performing unnecessary motions, and poor ergonomics. For examples Source of Waste: Developers not being involved in the design process despite having valuable UX insights, Delays due to waiting for approvals or inputs from team members and Errors due to lack of training or knowledge about best practices.
2. **Machine (Tools and Technology):-**Outdated tools causing frequent breakdowns and defects, Inefficient , poor integration, and lack of automation. For examples Source of Waste: Slow or outdated development environments that hinder coding speed, Poor integration between content management systems and development tools, causing repeated manual updates and Manual testing processes that could be automated to save time.
3. **Material:-**Excess inventory of obsolete parts, redundant resources, inefficient use of materials and poor quality materials. For examples Source of Waste: Accumulating a large backlog of features that are not prioritized or aligned with current user needs, Redundant or outdated code snippets that complicate maintenance and Unused design assets created during initial brainstorming but never implemented.
4. **Method (Processes):-**Inefficient workflows, lack of standardization, poorly defined processes, Unclear work instructions leading to over processing, and inefficient processes. For examples Source of Waste: Lengthy and complicated approval processes causing project delays, Inconsistent coding standards leading to bugs and rework and Inefficient project management practices resulting in missed deadlines.
5. **Measurement:-** Lack of process control data to detect defects early, Inaccurate or insufficient data, lack of performance metrics, poor feedback mechanisms. For examples Source of Waste: Lack of tracking project progress, causing unrecognized delays, Insufficient user feedback mechanisms leading to features that do not meet user needs and Inconsistent performance metrics that do not provide actionable insights.
6. **Surroundings (Environment):-**Poor working environment, distractions, inadequate resources, Poor layout, lighting, noise, temperature causing worker fatigue and errors. For examples Source of Waste:Open office layouts causing frequent interruptions and

distractions for developers, Limited access to necessary hardware or software tools and Poorly designed workspaces leading to ergonomic issues and inefficiencies.

Effects of waste: Waste negatively impacts productivity, efficiency, and profitability in the workplace. The effects of Muda, or waste, in production processes can be detrimental to organizations in various ways:

- Increased Costs: Waste leads to unnecessary consumption of resources such as materials, labor, energy, and time. This results in higher production costs, reducing profitability and competitiveness.
- Reduced Productivity: Waste creates inefficiencies in production workflows, leading to delays, bottlenecks, and idle time. This reduces the overall productivity of the organization and limits its capacity to meet customer demand.
- Poor Quality: Muda often leads to defects, errors, and rework in the production process. This results in lower product quality, increased customer complaints, and higher costs associated with warranty claims and returns.
- Excess Inventory: Overproduction and inefficient inventory management practices result in excess inventory levels. This ties up capital, occupies valuable storage space, and increases the risk of obsolescence or spoilage.
- Long Lead Times: Waste such as waiting, transportation delays, and inefficient processes contribute to longer lead times for delivering products to customers. This reduces responsiveness to customer needs and increases the risk of losing business to competitors.
- Employee Frustration: Waste in the form of inefficient processes, poor communication, and underutilization of talent can lead to frustration and disengagement among employees. This affects morale, job satisfaction, and retention rates.
- Environmental Impact: Some forms of waste, such as excessive energy consumption, pollution, and waste generation, have negative environmental consequences. This can harm the organization's reputation, lead to regulatory compliance issues, and incur additional costs for environmental remediation.

- **Loss of Competitive Advantage:** Organizations that fail to address waste effectively risk losing their competitive advantage in the market. Competitors who can produce higher quality products more efficiently are likely to capture market share and outperform them.

2.3.3 Visual Management Tools for Waste Identification:

Visual management is a business management approach that communicates important information in a visual and real-time manner. It is a system of labels, signs, markings, information displays, and visual guides instead of written instructions. Kaizen organizations rely significantly on visual management to detect abnormalities, reinforce standards, and ensure stability and safety are maintained in the workplace. Visual management is particularly important during the early phase of Lean implementation.



Figure 2.2: *Visual Board*

A good illustration that demonstrates the power of visual management is found in road signs, traffic lights, and lane markers on the road. The messages they convey are so clear that when you see a traffic light for example, you know exactly what you should be doing. Visual management includes a wide range of visual controls that help making all workplace elements and processes more visually apparent. These visual controls can be:

- Informative to show identity, directions, strategic goals, customer expectations and compliance requirements.
- Instructional to communicate SOPs, work-related information, and workplace organization and maintenance activities.

- Result-oriented to display the status of processes, projects, production, productivity and performance.

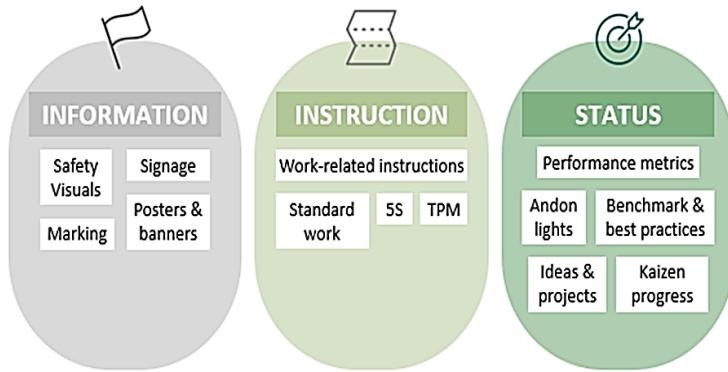


Figure 2.3: Visual Management techniques

Kaizen board

Kaizen board is a bulletin board set up at a workplace or in a publicly accessible place in the factory or the company in order to disseminate information about the Kaizen activities at the workplace and the company. Information put up on the board includes various Kaizen- related news and announcements, either company-wide one or particular workplace related.

In general, a Kaizen board is important for:

- Continuous Kaizen activity in a company or organization.
- Participation of all employees during Kaizen activity through Suggestion system.
- Employees including management to know about Kaizen performance in their work area or organization.
- Employees and management to know about production plans and performance.

The size of a Kaizen board should be 2 times a flip chart paper. It has four corners or parts depending on the information displayed.

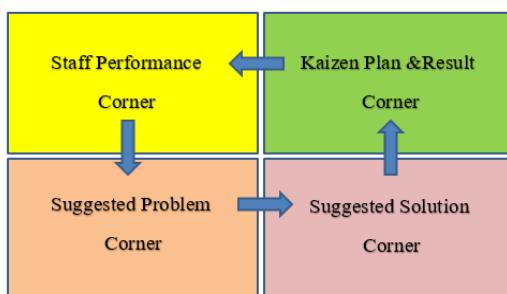


Figure 2.4: Kaizen Board

- The “staff performance corner” shows the actual performance of the staff and the gaps and training needs. The staff performance can be shown on the Kaizen board using different colors such as red for low performer, blue for average performer, and green for best performer.
- The “Kaizen plans & results corner” shows the results generated from implementing Kaizen activities. Improvement graphs can be displayed and should be updated regularly at least on weekly basis. If the results are below the planned target, the team has to discuss and find the root cause and implement solution
- The “suggested problems corner” is the place where every team member’s ideas or identified problems are posted. The posted problems have to be discussed and solved by the teams and the solutions should be displayed on the next corner i.e. “suggested solutions corner”.
- The “suggested solutions corner” displays the solutions suggested for known problems. And the solution ideas have to be implemented and the results achieved have to be shown on the “Kaizen results corner”.

In website development, Visual management tools are essential for identifying and addressing waste. These tools help teams visualize processes, track progress, and highlight areas for improvement, making it easier to implement Kaizen principles effectively. Following are tools in Website Development:

- **Kanban Boards:** A Kanban board is a visual tool used to manage workflow and tasks. It typically consists of columns representing different stages of work (e.g., To Do, In Progress, Done). It Provides a clear overview of work in progress, helps identify bottlenecks and areas of waste and facilitates continuous workflow and limits work-in-progress.
- **Value Stream Mapping (VSM):** A value Stream Mapping is a tool used to visualize the steps involved in delivering a product or service. It helps identify non-value-adding activities and areas of waste. It provides a comprehensive view of the entire development process, identifies delays, redundancies, and inefficiencies, facilitates targeted improvements and waste elimination.

- **Gantt Charts:** A Gantt chart is a bar chart that represents a project schedule. It shows the start and finish dates of project elements and the dependencies between tasks. It visualizes project timelines and task dependencies, identifies potential delays and overlapping tasks and helps in resource allocation and scheduling.
- **Pareto Charts:** It is a bar chart that shows the frequency of defects, problems, or causes in descending order. It helps identify the most significant factors contributing to waste.
- **Root Cause Analysis (RCA) with Fishbone Diagrams:** It involves identifying the root causes of problems or defects. Fishbone diagrams (also known as Ishikawa diagrams) are used to visually map out the causes of a specific problem.
- **Process Flow Diagrams:** It is a visual representation of the steps in a process. It helps in understanding the sequence of activities and identifying inefficiencies.

2.4 Identifying and Eliminating Muda

2.4.1 Gemba Walks and Observation Techniques

Gemba walks involve going to the actual workplace, observing operations, and engaging with employees to understand work processes, identify inefficiencies, and uncover sources of waste. By being on the "gemba" (the actual place where work is done), managers and leaders gain valuable insights into waste and opportunities for improvement.



Figure 2.5: Gemba Walks and Observation Techniques

Gemba walks are a powerful tool for understanding and improving processes by directly observing the actual work being done. Here are some key techniques for conducting effective Gemba walks and observations:

Prepare for the Walk

- Define a clear purpose and scope for the walk
- Prepare questions, checklists, and tools to guide the observation
- Inform the team ahead of time to put them at ease

Observe and Engage

- Start at the end of the process and work backwards
- Observe the process as a whole, not just individual steps
- Look for value-added vs non-value-added activities, wastes, and inefficiencies
- Trace the flow of materials, information, and people
- Compare the actual process to the standard process
- Interview employees with open-ended questions to gain their perspective
- Avoid disrupting the process or providing solutions on the spot

Document Findings

- Record observations, ideas, and findings
- Take photos or videos to document the process
- Avoid preconceived notions and keep an open mind

Follow Up

- Discuss learnings with the team and determine improvement opportunities
- Implement changes and return to the Gemba to verify results
- Regularly conduct Gemba walks as part of continuous improvement

The key is to go to the actual place of work, observe the process, engage employees, and identify opportunities to streamline operations and eliminate waste. Proper preparation, an open mindset, and follow-through are essential for Gemba walks to be effective.

2.4.2 Root Cause Analysis for Waste:

Root Cause Analysis (RCA) is a systematic method used to identify and address the underlying causes of problems or waste in a process. The goal of RCA is to eliminate the root cause of the problem, rather than just treating its symptoms.



Figure 2: Root cause vs symptom

Here's a step-by-step guide to conducting a Root Cause Analysis for waste identification and elimination:

|  Steps: Root Cause Analysis |
|---|
| Step 1: Define the Problem <ul style="list-style-type: none">Identify the specific problem or waste that needs to be addressed.Clearly define the problem statement, including the scope, impact, and any relevant metrics. |
| Step 2: Gather Data <ul style="list-style-type: none">Collect relevant data and information related to the problem.Review documents, records, and reports to understand the process and identify potential causes.Conduct interviews with stakeholders, including employees, customers, and suppliers. |
| Step 3: Identify Potential Causes <ul style="list-style-type: none">Brainstorm potential causes of the problem using tools such as:Fishbone diagrams (Ishikawa diagrams)Cause-and-effect diagramsPareto chartsFlowchartsIdentify potential causes based on the data and information gathered. |
| Step 4: Analyze Causes |

- Evaluate each potential cause using tools such as:
- Failure Mode and Effects Analysis (FMEA)
- Fault Tree Analysis (FTA)
- Systemic Failure Analysis
- Identify the most likely cause of the problem.

Step 5: Identify Root Causes

- Use the analysis results to identify the root cause of the problem.
- A root cause is defined as the underlying cause that has led to the problem, rather than a symptom or immediate cause.
- Ensure that the root cause is specific, measurable, achievable, relevant, and time-bound (SMART).

Step 6: Develop Action Plan

- Based on the identified root cause, develop an action plan to address it.
- Identify specific steps to eliminate or mitigate the root cause.
- Assign responsibilities and establish timelines for implementation.

Step 7: Implement and Monitor

- Implement the action plan and monitor its effectiveness.
- Track progress and adjust the plan as needed.
- Continuously evaluate and improve the process.

2.5 Continuous Improvement Mindset in Waste Reduction:

Adopting a continuous improvement mindset is essential for sustaining waste reduction efforts over time. The concept of a continuous improvement mindset in waste reduction emphasizes the ongoing effort to identify, address, and eliminate waste within an organization. It involves fostering a culture of innovation, learning, and proactive problem-solving where employees at all levels are engaged in seeking opportunities for improvement. Here's how this mindset contributes to waste reduction:

- Proactive Identification of Waste: Employees are encouraged to continuously observe and analyze processes to identify sources of waste. By being proactive in waste identification,

organizations can address inefficiencies before they escalate and negatively impact productivity and quality.

- **Empowerment and Involvement:** Cultivating a continuous improvement mindset empowers employees to take ownership of waste reduction initiatives. When employees feel empowered and involved in the decision-making process, they are more likely to actively participate in waste reduction efforts and contribute innovative solutions.
- **Iterative Problem-Solving:** Embracing a continuous improvement mindset involves adopting an iterative approach to problem-solving. Instead of viewing waste reduction as a one-time effort, organizations continuously seek opportunities for improvement, implement solutions, evaluate outcomes, and make further refinements.
- **Learning Culture:** Organizations with a continuous improvement mindset prioritize learning and knowledge sharing. They encourage employees to experiment with new ideas, learn from both successes and failures, and share best practices across teams and departments. This culture of learning fosters creativity, innovation, and continuous improvement.
- **Data-Driven Decision-Making:** A continuous improvement mindset emphasizes the importance of data-driven decision-making in waste reduction efforts. By collecting and analyzing data related to process performance, organizations can identify trends, root causes of waste, and opportunities for improvement, enabling more informed decision-making.
- **Leadership Support and Alignment:** Leadership plays a crucial role in fostering a continuous improvement mindset. Leaders should actively support and promote waste reduction initiatives, allocate resources, and provide guidance to ensure alignment with organizational goals and objectives.
- **Sustained Focus on Improvement:** Finally, a continuous improvement mindset ensures that waste reduction efforts remain a priority over time. Organizations regularly revisit and review their waste reduction strategies, adapt to changing circumstances, and continuously strive for excellence in all aspects of operations.

Unit Summary

Muda, or waste, is a key principle in Kaizen, a continuous improvement philosophy. It refers to any activity or process that doesn't add value to the final product or service from the customer's perspective. Identifying and eliminating waste is essential for improving efficiency, reducing costs, and enhancing quality. There are eight types of waste in the workplace: overproduction, excess inventory, transportation waste, motion waste, waiting-time waste, over processing, defects, and unused human skills.

Eliminating these types of waste can lead to significant improvements in efficiency and productivity. For instance, eliminating overproduction can reduce inventory costs and improve cash flow, while reducing motion waste can improve employee safety and ergonomic hazards. Adopting a culture of continuous improvement and eliminating waste can lead to sustainable growth and competitiveness, ensuring high-quality products and services for customers.

Unit Review Questions

Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What does Muda refer to in the context of Kaizen?
 - A) Efficient processes
 - B) Value-added activities
 - C) Waste activities
 - D) Customer satisfaction
2. Which of the following is NOT one of the eight types of waste in the workplace?
 - A) Overproduction
 - B) Excess inventory
 - C) Employee satisfaction
 - D) Transportation waste

3. How does eliminating overproduction benefit an organization?
 - A) Reduces inventory costs and improves cash flow
 - B) Increases customer complaints
 - C) Enhances employee satisfaction
 - D) Expedites production processes
4. Which type of waste involves unnecessary movement of people or equipment?
 - A) Overproduction
 - B) Motion waste
 - C) Over processing
 - D) Defects
5. What is one benefit of reducing motion waste in the workplace?
 - A) Reduces ergonomic hazards
 - B) Increases overproduction
 - C) Improves inventory management
 - D) Decreases customer satisfaction
6. Why is adopting a culture of continuous improvement important in eliminating waste?
 - A) To increase overproduction
 - B) To reduce customer complaints
 - C) To sustain growth and competitiveness
 - D) To lower employee satisfaction
7. What does Kaizen aim to achieve through the elimination of waste?
 - A) High employee turnover
 - B) Sustainable growth
 - C) Excessive inventory
 - D) Reduced production speed
8. Which type of waste involves defects or errors that require rework?
 - A) Overproduction
 - B) Waiting-time waste
 - C) Motion waste
 - D) Defects

9. How does eliminating defects contribute to improved efficiency?

- A) Reduces transportation waste
- B) Decreases employee training costs
- C) Reduces rework and enhances quality
- D) Increases over processing

10. What is a key benefit of eliminating waste in Kaizen philosophy?

- A) Decreases employee engagement
- B) Improves efficiency and productivity
- C) Increases overproduction
- D) Reduces customer satisfaction

Answer Key for Self-check Questions

Self-check 2-1: Answer Key

1. Waste, in the context of Kaizen, refers to any activity or process that does not add value to the final product or service from the customer's perspective.
2. Obvious wastes are easily identifiable, while hidden wastes are more challenging to uncover but often have a larger impact.

Self-check 2-2: Answer Key

1. "Muda" means waste in Japanese, and it's any activity that doesn't add value to the final product or service.
2. Overproduction is making more than what's needed, causing resource waste and delays in both manufacturing and services.
3. Excess inventory means having more materials than necessary, leading to space, cost, and working capital issues in manufacturing and services.
4. Transportation waste is unnecessary movement of materials or information, resulting in increased time, costs, and risk in both manufacturing and services.
5. Waiting-time waste is idle time due to delays, which increases lead times and costs without adding value in manufacturing and service

UNIT 3

5s PROCEDURE

Unit Learning Outcomes

At the end of this unit, students will be able to:

- Understand basic concepts of 5s procedures
- Perform 5S activities
- Promote and maintain 5s culture

Key terms: *5s, Sort, Set-order , Shine, Standardize, Sustain and Red gate*

3.1. Meaning of 5S

The 5S methodology is an improvement tool for organizing and maintaining a disciplined and productive workplace. It facilitates the creation of a better working environment by reducing waste while improving efficiency, safety and quality. 5S represents five fundamental practices that starts with the letter ‘S’. It is commonly applied by manufacturing facilities in production lines, storage areas, maintenance areas, and offices. Rooted in Japanese management principles, 5S was originally developed by Toyota as an integral component of their Lean production system. Many companies start their Lean transformation journey with 5S because it is one of the easiest Lean techniques and exposes some of the most visible examples of waste. Many experts believe that you need to be successful with 5S so you don’t struggle with the other techniques during Lean implementation.

Details of 5S approach

5S: Sort- Set- Shine- Standardize- Sustain

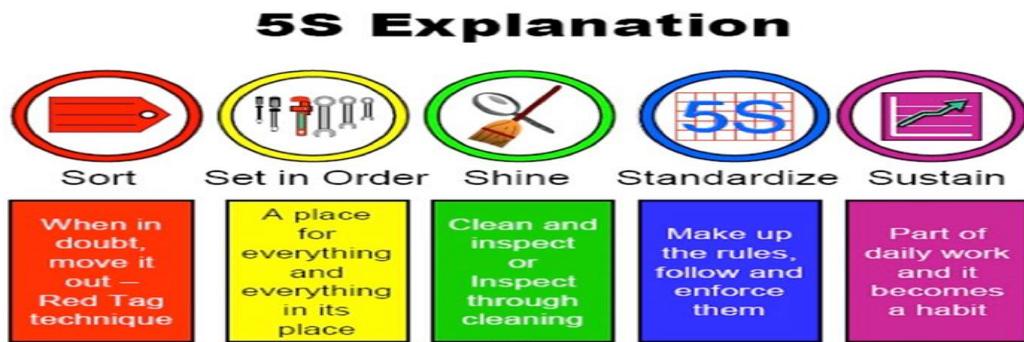


Figure 3.1: 5S

Table 3.1: The 5Ss are: listed by Japanese, English and Amharic language

| Japanese | English | Amharic | |
|----------|--------------|---------|--|
| Seiri | Sort | 整理 | The first step in 5S is to eliminate all the things in the workspace that are not being used and store them away. If a tool or material is not used on a daily basis, eliminate it from the workstation. |
| Seiton | Set in Order | 整頓 | The second step is to arrange the items used on a daily basis so that they can be easily accessed and quickly stored. Your goal is to make eliminate any unnecessary movements and actions by the worker to make his process as efficient as possible. |
| Seiso | Shine | 整掃 | Next is to get everything cleaned and functioning properly. The goal is to remove all the dirt and the grime and to keep it that way on daily basis. You want to get it clean and keep it clean. |
| Seiketsu | Standardize | 整規 | The fourth step is to develop a routine for sorting, setting and shining. Standardize creates a system of tasks and procedures that will ensure that the principles of 5S are performed on a daily basis |
| Shitsuke | Sustain | 整守 | In the last step, you want to create a culture that will follow the steps on a daily basis. The chief objective of sustain is to give your staff the commitment and motivation to follow each step, day in and day out. |

3.2. Benefit of Implementing 5S

- Improves safety and ergonomics
- Promotes flow
- Reduces searching
- Reduces unplanned downtime
- Improve quality
- Enhances teamwork
- Tackles waste
- Improves productivity
- Eliminates distractions
- Reduces inventory and space
- Instills the discipline to follow standard work
- Encourages visual control
- Exposes problems
- Enhances self-management

3.3 The 5S Phases

The term 5S is an abbreviation for five Japanese words: **seiri**, **seiton**, **seisou**, **seiketsu**, and **shitsuke**. These five words are often translated into English as: **sorting**, **setting in order**, **shining**, **standardizing**, and **sustaining**.



Figure 3.2: The 5S Phases

3.3.1 Sort/seiri

Separate required tools, materials, and instructions from those that are not needed. Remove everything that is not necessary from the work area. An easy rule is to remove anything that will not be used within the next 30 days. Follow the Steps to implement:

- **Identify:** Evaluate items in the workspace and determine what is needed for current operations.
- **Categorize:** Classify items as necessary or unnecessary based on current use and frequency.

- **Red Tagging:** Tag items that are not immediately necessary for further review or removal.
- **Dispose/Eliminate:** Remove items that are obsolete, redundant, or not essential to the workflow and remove unnecessary items from the workspace to reduce clutter and improve efficiency.



Figure 3.3: Example of sorting activity

For Example: In the context of web development, conduct a content audit to identify and remove outdated, unused and redundant pages, content, images, libraries, directories, files user interface (UI) and old blog posts that are no longer available. This declutters the website, making it easier for users to find relevant information. Simplify navigation menus by categorizing and prioritizing essential links and pages for better user experience.

Red Tagging in the Sort (Seiri)

A "red tag" is typically a tag, label or sticker used to mark items that are defective, out of service, or require attention. It serves as a visual indicator to personnel that the items are no longer needed in a particular workplace or workshop. Red tags are attached to unneeded tools, equipment, and supplies.



Figure 3.4: Example of Red tags

3.3.2 Set in Order/ seiton:

Arrange necessary items in a logical and efficient manner so they are easy to find, use, and put away. Maintain a system where finding an item takes less than 30 seconds. Follow the Steps to implement:

- Organize:** Designate specific locations for tools, equipment, and materials based on frequency of use and workflow.
- Standardize:** Establish guidelines for how items should be stored and ensure consistency across the workspace.
- Visual Management:** Use labels, color coding, and clear signage to indicate where items belong and to maintain order.



Figure 3.5: Example of Set in order

For Example: Regarding web development, Arrange or create your files and folders logically within your project directory. Use meaningful names such as "assets/img", "scripts/", and "styles/" to easily locate and manage resources. keep all CSS files in a "styles" folder,

JavaScript files in a "scripts" folder, and images in an "images" folder. This organization helps developers quickly locate and navigate through project files.

3.3.3 Shine/ Seiso

Clean and inspect the workspace regularly to ensure it is maintained in a clean and orderly condition. Follow the Steps to implement:

- **Clean:** Regularly clean work surfaces, equipment, and storage areas to remove dirt, dust, and debris.
- **Inspect:** Check for signs of wear, damage, or potential safety hazards during cleaning.
- **Preventive Maintenance:** Address minor repairs and maintenance issues promptly to prevent larger problems.



Figure 3.6: Example of Shine

For Example: Within the realm of web development, regularly review or clean contents and codes to eliminate unused variables, functions, loops or outdated comments. Optimize images and multimedia content to reduce file sizes and improve loading speeds/times without compromising quality. Conduct usability testing to identify and fix UI/UX issues that may impact user satisfaction. Ensure all interactive elements (buttons, forms) are functioning correctly across various browsers and devices.

3.3.4 Standardize

Engage the workforce to systematically perform steps 1, 2, and 3 above daily to maintain the workplace in perfect condition as a standard process. Establish and maintain standardized work practices to sustain the improvements made during the Sort, Set in Order, and Shine phases. Follow the Steps to implement:

- **Develop Procedures:** Document and communicate standard operating procedures (SOPs) for organizing, cleaning, and maintaining the workspace.
- **Training:** Train employees on the established procedures and ensure they understand their roles in maintaining standards.
- **Audit and Review:** Regularly audit the workspace to ensure adherence to standards and identify opportunities for further improvement.



Figure 3.7: Example of Standardize

For Example: In the sphere of web development, Establish and enforce coding standards, best practices and design guidelines across different pages or modules of a website for all projects to maintain uniformity in code structure, naming conventions, HTML, CSS, JavaScript, and other languages you use. Create a style guide documenting typography, color palette, button styles, and UI components. Apply responsive web design principles to ensure the website adapts seamlessly to different devices and screen sizes.

3.3.5 Sustain

Create a culture where the 5S principles are practiced consistently and continuously improved upon. Follow the Steps to implement:

Employee Engagement: Foster ownership and responsibility among employees for maintaining the workspace.

Recognition and Rewards: Recognize and reward individuals or teams that demonstrate commitment to 5S principles.

Continuous Improvement: Encourage ongoing evaluation and refinement of processes to adapt to changing needs and improve efficiency.



Before

After

Figure 3.8: Example of Sustain

For Examples: With respect to web development, Schedule Maintain and continuously improve website performance and design standards and keep content and technology up to date. Implement regular maintenance routines, monitor website analytics, keep the website up-to-date, functioning smoothly and gather user feedback for ongoing improvements. Ensure that design and development standards are adhered to over time and adapt to changing needs and technologies.



Self-check 3-1:

1. What are the main objectives of the "Sort" phase (Seiri) in the 5S methodology?
2. Explain the importance of the "Set in Order" phase (Seiton) and how it is implemented.
3. Outline the strategies for maintaining cleanliness and organization in the "Shine" phase (Seiso).

4. What is the purpose of the "Standardize" phase in 5S, and how are standardized work procedures developed?
 5. Discuss the challenges and strategies for sustaining the 5S program in the "Sustain" phase.
-

Unit Summary

The 5S (Process, System, Service) is a systematic approach to creating and maintaining an organized, clean, and high-performance workplace. It focuses on sorting and removing unnecessary items, setting in order, and Shine to maintain cleanliness and order. Standardization is achieved through standardized work procedures and visual controls. Sustaining the practices involves regular audits, performance monitoring, and continuous improvement initiatives. Leadership commitment is crucial for prioritizing and sustaining 5S practices.

Unit Review Questions

Multiple Choice Questions

Instruction: Take the necessary time to carefully read the questions and choose the most suitable answer from the multiple choices provided.

1. What does the "Sort" phase in 5S primarily focus on?
 - A. Enhancing workplace safety
 - B. Standardizing work procedures
 - C. Removing unnecessary items
 - D. Implementing visual controls
2. Which phase of 5S aims to arrange necessary items for optimal efficiency?
 - A. Sort
 - B. Set in Order
 - C. Shine
 - D. Standardize

3. What is the main objective of the "Shine" phase in 5S?
 - A. Sorting and organizing items
 - B. Standardizing work procedures
 - C. Maintaining cleanliness
 - D. Implementing visual controls
4. How does 5S achieve standardization of work procedures?
 - A. Through regular audits
 - B. By implementing Kaizen events
 - C. Using visual controls and standardized work instructions
 - D. Conducting employee training sessions
5. What is a key strategy for sustaining 5S practices in the workplace?
 - A. Implementing Kaizen events
 - B. Conducting regular audits
 - C. Designating team leaders
 - D. Reducing employee workload
6. Why is leadership commitment crucial in implementing 5S?
 - A. To increase employee workload
 - B. To delegate responsibilities effectively
 - C. To prioritize and sustain 5S practices
 - D. To eliminate unnecessary items
7. Which concept is closely associated with continuous improvement in the context of 5S?
 - A. Standardization
 - B. Set in Order
 - C. Kaizen
 - D. Sorting
8. What role do visual controls play in the 5S methodology?
 - A. Ensuring workplace safety
 - B. Standardizing work procedures
 - C. Sustaining cleanliness
 - D. Providing visual guidance for organization and efficiency

9. What does the "Sustain" phase of 5S primarily focus on?
 - A. Conducting regular audits
 - B. Removing unnecessary items
 - C. Continuous improvement
 - D. Establishing standardized work procedures
10. Which phase of 5S involves arranging items in a way that eliminates waste and improves efficiency?
 - A. Sort
 - B. Set in Order
 - C. Shine
 - D. Standardize

Answer Key for Self-check Questions

Self-check 3-1: Answer Key

1. The main objectives of the "Sort" phase (Seiri) in the 5S methodology are to eliminate unnecessary items, organize essential items, enhance safety, and improve workflow.
2. The "Set in Order" phase (Seiton) is important because it arranges necessary items for optimal efficiency and accessibility, reducing time spent searching for tools and materials. It is implemented by designating specific locations for each item, using labeling and color-coding, and organizing the workspace logically to ensure everything is easy to find and return.
3. Strategies for maintaining cleanliness and organization in the "Shine" phase (Seiso) include:
 - Regular Cleaning Schedules: Establish routine cleaning tasks.
 - Assign Responsibilities: Designate specific cleaning duties to team members.
 - Conduct Inspections: Perform regular checks to ensure cleanliness standards are maintained.

4. Purpose of the "Standardize" Phase in 5S:
 - Ensure consistency and reliability by establishing uniform work procedures.
 - Document clear guidelines, provide training, and use visual aids.
5. Challenges and Strategies for Sustaining the 5S Program in the "Sustain" Phase:
 - Challenges: Maintaining employee engagement and preventing regression.
 - Strategies: Regular audits, leadership support, and continuous employee involvement.

References

- "5S Procedure," n.d.). Retrieved from <https://www.google.com/search?client=firefox-b-d&q=5s+procedure>
- Imai, M. (2012). Gemba Kaizen. The McGraw-Hill Companies.
- Japan International Cooperation Agency. (n.d.). Retrieved from <https://www.jica.go.jp/Resource/activities/issues/health/>
- Kaufman Global. (n.d.). 7 Types of Waste. Retrieved from <https://www.kaufmanglobal.com/glossary/7-types-waste/>
- Liker, J. K., & Meier, D. (2006). The Toyota Way Fieldbook. The McGraw-Hill Companies.
- Ministry of Labor and Skill Development (MOLSE). (2014). Apply 5S and Prevent and Eliminate Muda, Addis Ababa.
- National Productivity Corporation. (2005). Petaling Jaya, Malaysia: National Productivity Corporation.
- National Productivity Corporation. (2005). Petaling Jaya, Malaysia: National Productivity Corporation.
- No specific author provided. ("7 Wastes of Lean Production," n.d.). Retrieved from <https://www.quality-assurance-solutions.com/7-wastes.html#ixzz8O9DWOMne>
- PTC. (n.d.). 7 Wastes of Lean Production. Retrieved from <https://www.ptc.com/en/blogs/iiot/7-wastes-of-lean-production>