


1. UNDERSTANDING THE INTERNET



The Internet is a global network of interconnected computers **2** that use standardized communication protocols to exchange information. It enables users to access web pages, send emails, stream media, and much more.

Key Features:

- **Global Connectivity:** Links computers worldwide.
- **Decentralized Structure:** No central governing authority.
- **Uses IP Addresses & DNS:** Identifies devices and translates domain names to IP addresses.
- **Protocols:** Uses HTTP, HTTPS, FTP, TCP/IP, etc.

- **World Wide Web**

- The World Wide Web (abbreviated WWW or the Web) is an information space³ where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland.
- WWW is another example of client/server computing. Each time a link is followed, the client is requesting a document (or graphic or sound file) from a server (also called a Web server) that's part of the World Wide Web that "serves" up the document. The server uses a protocol called HTTP or Hyper Text Transport Protocol. The standard for creating hypertext documents for the WWW is Hyper Text Markup Language or HTML. HTML essentially codes plain text documents so they can be viewed on the Web.

Notes on Internet Protocols

Internet Protocols are a set of rules governing how data is sent, received, and processed across networks, including the Internet. Below is an overview of essential protocols and their characteristics: Internet protocols are vital for communication over networks. They define how data is formatted, transmitted, routed, and received, ensuring seamless connectivity and secure data transfer across diverse applications.

1. Internet Protocol (IP)

Purpose: Governs the addressing and routing of packets across networks.

Key Versions:

- **IPv4:** Uses 32-bit addresses (e.g., 192.168.1.1). Supports ~4.3 billion addresses.
- **IPv6:** Uses 128-bit addresses (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Supports a vastly larger address space.

Functions:

- **Addressing:** Assigns unique IP addresses to devices.
- **Routing:** Ensures data reaches its destination across multiple networks.

Connection Type: Connectionless, meaning no session is established before data is sent.

2. Transmission Control Protocol (TCP)

Purpose: Ensures reliable delivery of data packets.

5

Characteristics:

- **Connection-oriented:** Establishes a connection before data transmission (e.g., via a handshake process).
- **Reliable:** Ensures all packets are delivered and reassembled in order.
- **Error detection and correction.**

Applications: Web browsing (HTTP/HTTPS), email (SMTP), file transfer (FTP).

3. User Datagram Protocol (UDP)

Purpose: Provides faster but less reliable data delivery than TCP.

Characteristics:

Connectionless: No handshake or session establishment.

Lightweight: Fewer overheads compared to TCP.

No guarantees for delivery, order, or error correction.

Applications: Real-time applications like video streaming, online gaming, and Voice over IP (VoIP).

4. HyperText Transfer Protocol (HTTP)

Purpose: Facilitates the transfer of hypertext (web pages) between web browsers and servers.

Characteristics:

Stateless protocol: Each request is independent.

Versions:

HTTP/1.1: Persistent connections and chunked transfers.

HTTP/2: Improved performance via multiplexing and compression.

HTTP/3: Uses QUIC for better speed and reliability.

Secure Variant: HTTPS, which uses SSL/TLS for encryption.

5. File Transfer Protocol (FTP)

Purpose: Used to transfer files between a client and a server.

Modes:

Active: Client opens a port for the server to connect.

Passive: Server opens a port for the client to connect.

Secure Variant: SFTP (Secure File Transfer Protocol), which encrypts file transfers.

6. Simple Mail Transfer Protocol (SMTP)

7

Purpose: Used to send emails between mail servers and from clients to mail servers.

Related Protocols:

POP3: Downloads emails from the server for offline access.

IMAP: Synchronizes emails between the server and multiple devices.

7. Domain Name System (DNS)

Purpose: Translates human-readable domain names (e.g., example.com) into IP addresses.

How It Works:

- Query types: Recursive, iterative.
- Hierarchy: Root servers, TLD servers, and authoritative servers.

Security Extension: DNSSEC ensures the authenticity of DNS responses.

Dynamic Host Configuration Protocol (DHCP)

Purpose: Dynamically assigns IP addresses and other network configurations (e.g., gateway, DNS server) to devices. **8**

Benefits: Simplifies network management and reduces manual configuration errors.

9. Secure Shell (SSH)

Purpose: Provides secure, encrypted communication for remote login and command execution.

Applications: Remote administration of servers, secure file transfers (SCP, SFTP).

10. Voice over IP (VoIP) Protocols

Purpose: Transmit voice data over IP networks.

Examples:

SIP (Session Initiation Protocol): Establishes and terminates VoIP sessions.

RTP (Real-Time Transport Protocol): Transports audio and video in real-time.

11. Simple Network Management Protocol (SNMP)

Purpose: Monitors and manages network devices like routers, switches, and servers.

Components:

- Managed devices.

- Agents (software on devices).

- Network management system (NMS).

12. Internet Control Message Protocol (ICMP)

Purpose: Used for diagnostics and error reporting.

Applications:

- Ping:** Checks connectivity between devices.

- Traceroute:** Identifies the path taken by packets.

13. Transport Layer Security (TLS)

Purpose: Secures communication over the Internet by encrypting data.

Uses: HTTPS, email encryption, and VPNs.

Successor: Replaces SSL (Secure Sockets Layer).

14. WebSocket Protocol

Purpose: Enables full-duplex communication between a client and server.

Applications: Real-time chat applications, live updates, and online gaming.

2. DIFFERENCE BETWEEN A WEBSITE AND AN APPLICATION SERVER

Website:

- A collection of web pages hosted on a server.
- Primarily static (informational content, blogs, news, etc.).

Uses HTML, CSS, and JavaScript to structure and design pages. Comes under client side scripting

Server

A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data center, the physical computer that a server program runs on is also frequently referred to as a server.

In the client-server model, a server program fulfills requests from client programs, which might be running on the same computer or other computers. A given application in a computer system might function as a client that makes requests for services from other programs and as a server fulfilling requests from other programs. Comes under server side scripting.

Types of servers

Web Server Database Server Email Server Web Proxy Server DNS Server FTP Server File Server DHCP Server Cloud Server Application Server Print Server NTP Server Radius Server Syslog Server Physical Server

Application Server:

- A platform that processes requests from a client and delivers dynamic content.
- Supports business logic and data processing.
- Works with databases, APIs, and backend frameworks (e.g., Node.js, Django, Spring Boot).
- An **Application Server** is a software framework that provides an environment for running web applications and services. It acts as a **middleware** between client requests and backend databases, ensuring smooth execution of applications.

- **Web Server:**

Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).

Basically, web servers are computers used to store HTTP files which makes a website and when a client requests a certain website, it delivers the requested website to the client. For example, you want to open Facebook on your laptop and enter the URL in the search bar of google. Now, the laptop will send an HTTP request to view the facebook webpage to another computer known as the webserver. This computer (webserver) contains all the files (usually in HTTP format) which make up the website like text, images, gif files, etc. After processing the request, the webserver will send the requested website-related files to your computer and then you can reach the website.

- Different websites can be stored on the same or different web servers but that doesn't affect the actual website that you are seeing in your computer. The web server can be any software or hardware but is usually a software running on a computer. One web server can handle multiple users at any given time which is a necessity otherwise there had to be a web server for each user and considering the current world population, is nearly close to impossible. A web server is never disconnected from the internet because if it was, then it won't be able to receive any requests, and therefore cannot process them.

Application Server vs Web Server

16

Feature

Purpose

Processing

Examples

Application Server

Runs business logic and manages application components

Executes complex backend logic

JBoss, WebSphere, WebLogic

Web Server

Serves static and dynamic web pages

Mainly processes HTTP requests

Apache HTTP Server, Nginx, Microsoft IIS

Popular Application Servers

Application Server

JBoss/WildFly

IBM WebSphere

Oracle WebLogic

Microsoft IIS

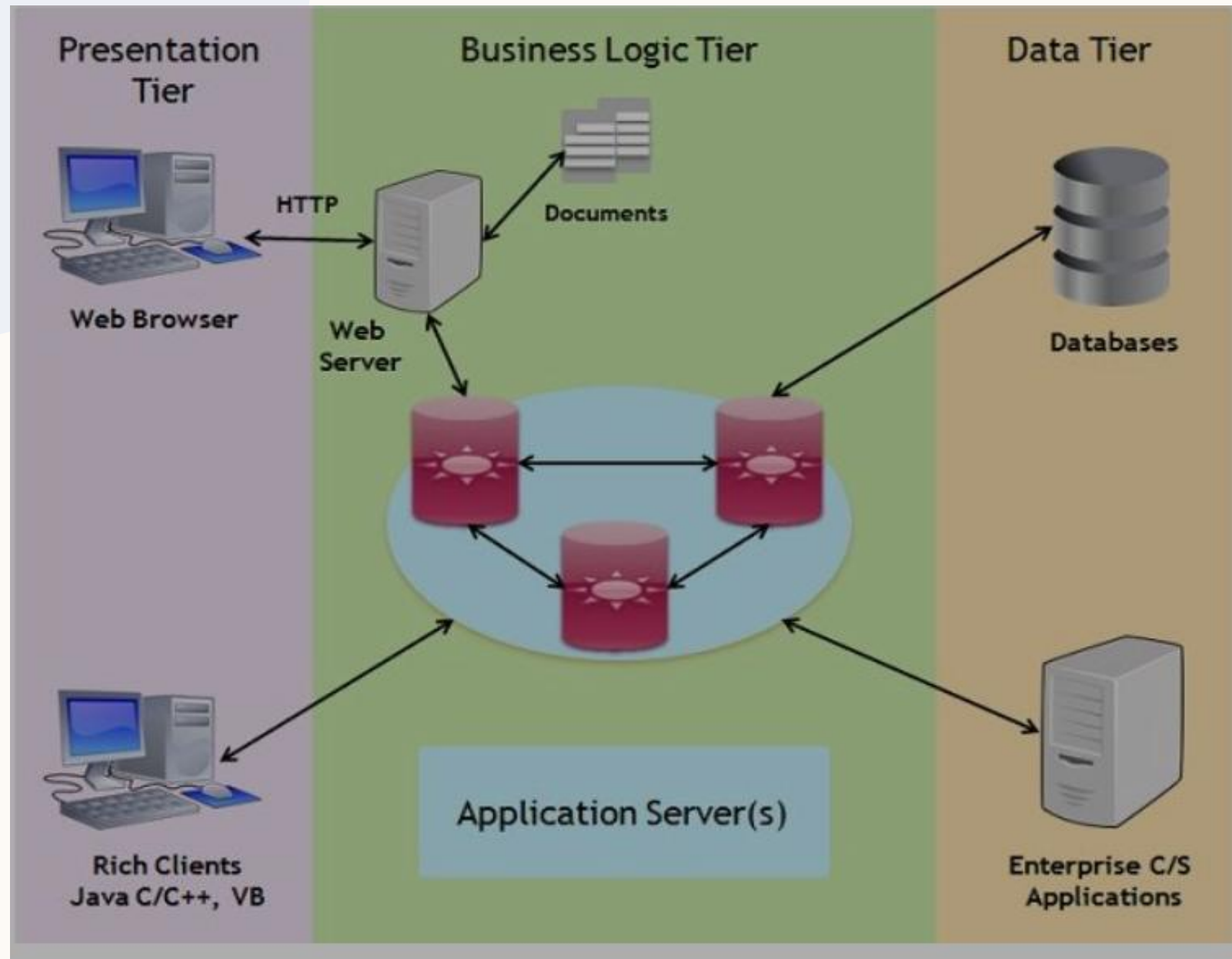
Description

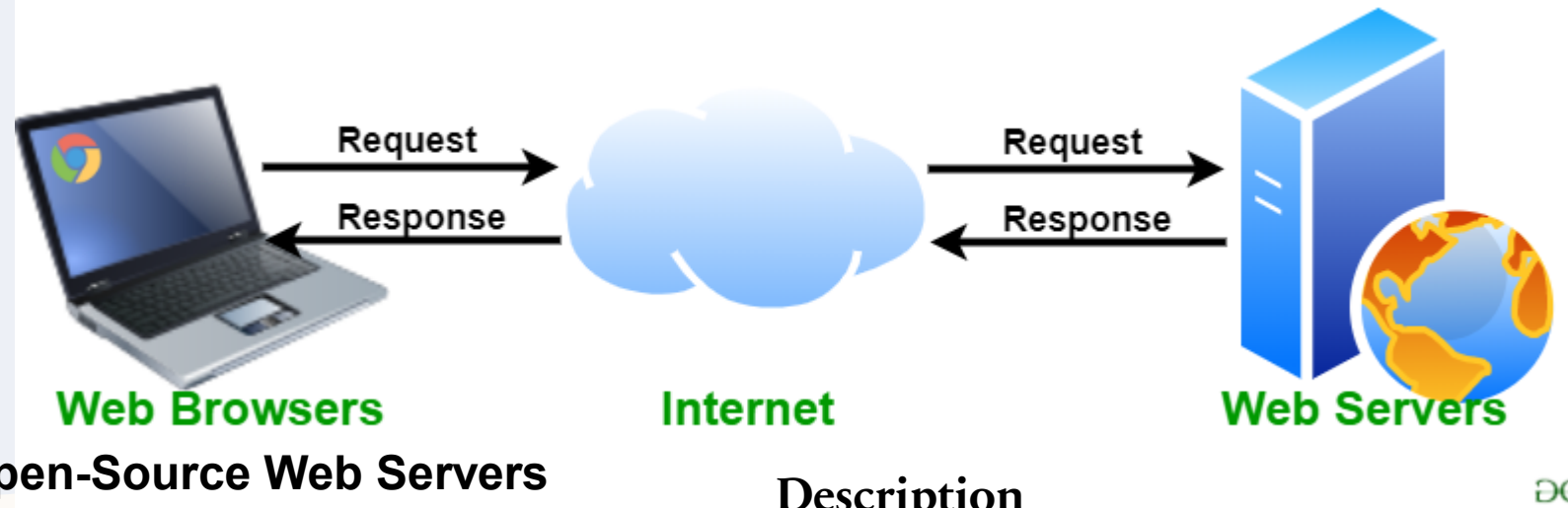
Enterprise-grade Java EE application server.

Commercial server for large-scale applications.

Robust server supporting Java EE and cloud applications.

Supports ASP.NET and integrates with Windows environments.





Web Server

Open-Source Web Servers

Description

Apache HTTP Server (Apache)

One of the most widely used web servers, developed by the Apache Software Foundation. Supports modules and virtual hosting. General-purpose web server for serving static and dynamic web pages written in C

NGINX

High-performance web server and reverse proxy. Known for handling high traffic and load balancing.

LiteSpeed

Lightweight, high-performance server that is often used as an alternative to Apache.

Caddy

A modern, secure web server with automatic HTTPS and easy configuration.

H2O

Optimized for HTTP/2 performance, offering fast response times.

Microsoft Web Servers

IIS (Internet Information Services)

ASP.NET Core Kestrel

Web Server

Apache Tomcat

Jetty

JBoss (WildFly)

GlassFish

Java-Based Web Servers

Description

Developed by Microsoft, mainly used for hosting ASP.NET applications on Windows.

A cross-platform web server for ASP.NET Core applications, often used with IIS or NGINX as a reverse proxy.

Description

Open-source Java-based web server for running Java Servlets and JSP applications written in Java

Lightweight and fast Java-based web server.

Enterprise-level web and application server for Java applications.

Java EE server, originally developed by Sun Microsystems (now Oracle).

Web Server

Cloud & Enterprise Web Servers

AWS Elastic Load Balancer (ELB)

Google Cloud Run

Azure App Service

Description

Amazon's cloud-based load balancing solution that distributes traffic to backend servers.

Serverless computing service that automatically manages the infrastructure.

Microsoft's PaaS offering for hosting web applications.

Web Server

Lightweight & Embedded Web Servers

Node.js (Express.js, Fastify, Koa)

Python (Flask, Django, Gunicorn, Tornado)

CivetWeb

Monkey HTTP Server

Description

JavaScript-based servers, often used for APIs and real-time applications.

Python-based web servers for handling web applications.

Embedded web server for IoT and small devices.

A minimalistic web server designed for embedded systems.

Web Server

Squid

Varnish

HAProxy

Proxy & Caching Web Servers

Description

Caching proxy for HTTP, HTTPS, and FTP.

High-performance HTTP accelerator and reverse proxy.

Used for load balancing and proxying web requests.

The choice of a web server depends on **performance, scalability, compatibility, and use case** (e.g., static content, dynamic applications, APIs, or cloud-based services).

Web servers are crucial for delivering web content to users efficiently and securely. Depending on the use case, different web servers are chosen for static or dynamic content, load balancing, and scalability.



4. UNDERSTANDING THE DIFFERENCE BETWEEN INTERNET AND INTRANET

22

Internet

- Public network accessible globally.
- Uses public IP addresses.
- Open to all users with an internet connection.
- Example: Google, Facebook, YouTube.

Intranet

- Private network within an organization.
- Uses private IP addresses and firewall protection.
- Restricted access for employees or authorized users.
- Example: Company's internal portal, university learning management systems.

Internet	Intranet	
Internet is used to connect different networks of computers simultaneously.	Intranet is owned by private firms.	24
On the internet, there are multiple users.	On an intranet, there are limited users.	
Internet is unsafe.	Intranet is safe.	
On the internet, there is more number of visitors.	In the intranet, there is less number of visitors.	
Internet is a public network.	Intranet is a private network.	
It is open to everyone and allows access to public information, such as websites and online services.	An intranet is only accessible to authorized users within the organization.	
Users can access the Internet from any location with an Internet connection and a compatible device.	Access to an intranet is restricted to authorized users within the organization and is typically limited to specific devices and locations.	

Security measures, such as firewalls, encryption, and secure sockets layer (SSL) protocols, are used to protect against threats like hacking, viruses, and malware.

Examples of Internet-based services include email, social media, search engines, and online shopping sites.

Intranets employ similar security measures to protect against unauthorized access and ensure the privacy and integrity of shared data.

Examples of intranet-based services include internal communications, knowledge management systems, and collaboration tools

5. HTML 5 OVERVIEW

Features:

<!DOCTYPE html> and has conditional statements for multiple browser versions.

- An inline frame is used to embed another document(html page) within the current HTML document. -<iframe>
- The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images. Canvas is supported by all major browsers.

- Improved security (CSP - Content Security Policy).
- Native support for rich media (audio & video elements).
- Enhanced form controls and validation.
- Semantic elements (e.g., <article>, <section>, <header>).
- Web components for reusable UI elements.

New Features in HTML5

- Semantic Elements: `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>` (Improve SEO and readability).
- Multimedia Support: `<audio>` and `<video>` (No need for Flash).
- Forms & Input Types: `<input type="email">`, `<input type="date">`, `<input type="range">`.
- Canvas & SVG: For graphics and animations.
- Local Storage & Session Storage: Replaces cookies.
- Geolocation API: Access user location.
- WebSockets & Web Workers: Improves real-time communication and performance.

List of new semantic tags in HTML5

Tag

`<header>`

`<nav>`

`<main>`

`<section>`

`<article>`

`<aside>`

`<footer>`

`<figure>`

`<figcaption>`

Meaning

Top section / heading area

Navigation links

Main content of the page

Section of content

Independent article/post

Side content (ads/related links)

Bottom section

Holds images/diagrams

Caption for figure

6. CSS 3.0 OVERVIEW

CSS3 is the latest standard for styling web pages, providing advanced design capabilities.

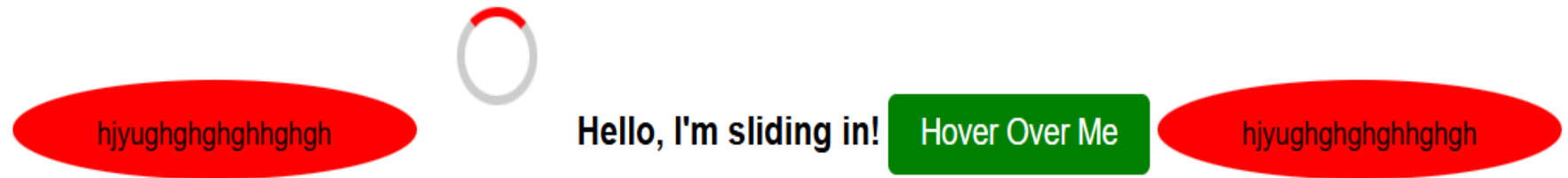
Features:

- **Selectors & Combinators:** Improved ways to target elements.
- **Media Queries:** Responsive design for different screen sizes.
- **Flexbox & Grid:** Layout models for better alignment and spacing.
- **Animations & Transitions:** Smooth effects without JavaScript.
 - @keyframes is a CSS rule that defines animations by specifying different styles at various points during an animation sequence.

It is used to create smooth transitions between states without using JavaScript.

- **Shadows & Gradients:** Enhanced visual appeal.

CSS Animations Demo



Welcome to My Website!

Hover to Fade In

Flashing Text!

Syntax of @keyframes

1. The basic syntax of @keyframes is:

<style>

```
@keyframes animationName {
```

```
  0% { /* Initial state */ }
```

```
  50% { /* Midway state */ }
```

```
 100% { /* Final state */ }
```

```
}
```

2. Applying @keyframes with animation Property

After defining the animation, it must be applied to an element using the animation property.

```
.element {                                --→ name of animation
```

```
  animation-name: animationName;
```

```
  animation-duration: 2s;
```

```
  animation-iteration-count: infinite;
```

```
}
```

```
</style>
```

4. Applying the @keyframe animation in html5 page

```
<div class="element"><br>This is my Animated text</div>
```

Animation List for @keyframes

@keyframes fadeIn

@keyframes rotate

@keyframes slideIn

@keyframes pulse

@keyframes bounce

@keyframes flash

Conclusion

- HTML is the backbone of web development.
- HTML5 enhances functionality with **new elements, better media handling, and improved performance.**
- Combine HTML5 with **CSS3 & JavaScript** for modern web applications.