Solved Short questions Past Paper 2022

Q1: MVC Stand for?

Ans: MVC stands for Model-View-Controller. It's a software design pattern frequently used to develop user interfaces (UIs). The core idea is to separate the application into three parts:

Model: This represents the data and business logic of the application. It handles things like data storage, retrieval, and any calculations needed.

View: This is the user interface itself. It's what the user sees and interacts with. It displays information from the model and doesn't contain any application logic.

Controller: This acts as an intermediary between the model and the view. It receives user input from the view, updates the model as needed, and instructs the view how to update itself based on the changes in the model.

MVC is popular because it promotes separation of concerns, which makes code easier to understand, maintain, and test. Each component has a specific responsibility, making it easier for developers to work on individual parts without worrying about the whole system breaking.

MVC is commonly used in web development, but it can also be applied to desktop applications. There are also various web frameworks that enforce the MVC pattern, providing a structured way to build web applications.

Q2: How to comment in HTML?

Ans: Adding comments in HTML is useful for explaining your code, making it easier to understand later on, especially for complex websites or when working with other developers. Here's how to do it:

HTML Comments Syntax

HTML uses special comment tags to hide sections of code from the browser. Browsers ignore anything placed between these tags when rendering the webpage.

The **syntax** for creating comments in HTML is:

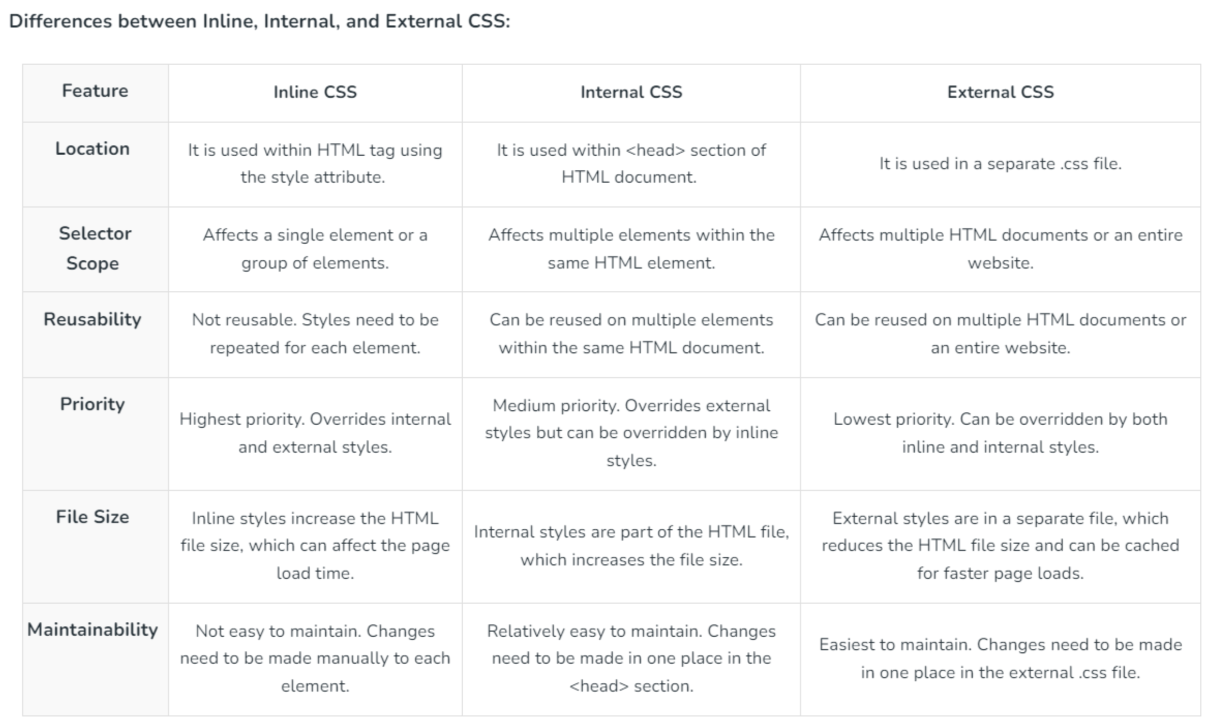
HTML

```
<! -- your comment here -->
```

Q3: Differentiate between Internal and Inline CSS?

Q4: Differentiate between Internal and External CSS?

Ans:



Differences between Inline, Internal, and External CSS:

Feature	Inline CSS	Internal CSS	External CSS
Location	It is used within HTML tag using the style attribute.	It is used within <head> section of HTML document.	It is used in a separate .css file.
Selector Scope	Affects a single element or a group of elements.	Affects multiple elements within the same HTML element.	Affects multiple HTML documents or an entire website.
Reusability	Not reusable. Styles need to be repeated for each element.	Can be reused on multiple elements within the same HTML document.	Can be reused on multiple HTML documents or an entire website.
Priority	Highest priority. Overrides internal and external styles.	Medium priority. Overrides external styles but can be overridden by inline styles.	Lowest priority. Can be overridden by both inline and internal styles.
File Size	Inline styles increase the HTML file size, which can affect the page load time.	Internal styles are part of the HTML file, which increases the file size.	External styles are in a separate file, which reduces the HTML file size and can be cached for faster page loads.
Maintainability	Not easy to maintain. Changes need to be made manually to each element.	Relatively easy to maintain. Changes need to be made in one place in the <head> section.	Easiest to maintain. Changes need to be made in one place in the external .css file.

Q5: Differentiate between stateless and stateful protocol?

Ans: Stateless protocols treat each request from a client as an independent event, with no memory of previous interactions. Every request contains all the information the server needs to respond. This makes them:

- **Simpler and faster:** Stateless servers are easier to design and scale as they don't need to manage session data.
- **Less reliable:** Since there's no memory of past requests, error handling and ensuring data arrives in order can be more complex.

Stateful protocols, on the other hand, maintain a session between the client and server. The server remembers past interactions and uses that context to process new requests. This allows for:

- **More complex interactions:** Stateful protocols are suited for scenarios like file transfers or video calls where keeping track of progress is essential.
- **Improved reliability:** Stateful protocols can check for errors and retransmit lost data packets.

Some common examples:

Stateless protocols: HTTP (used for web browsing), DNS (used for domain name resolution), UDP (used for real-time data transfer)

Stateful protocols: TCP (used for reliable data transfer), FTP (used for file transfer), Telnet (used for remote terminal access).

Q6: Difference between Http and https protocol?

Ans: HTTP and HTTPS are both protocols that enable communication between web browsers and servers. The main difference lies in security:

HTTP (Hypertext Transfer Protocol): This is the foundation of web communication. It establishes the rules for how data is transmitted between browsers and servers. However, HTTP doesn't encrypt the data being transferred. This means information you send or receive, like passwords or credit card details, can be intercepted by someone lurking on the network. It's like sending a postcard - anyone can read it.

HTTPS (Hypertext Transfer Protocol Secure): This is the secure version of HTTP. HTTPS uses encryption technology (TLS/SSL) to scramble the data being sent between the browser and server. This encryption makes it very difficult for anyone to intercept and decode the information, protecting your privacy. Think of it as sending a sealed letter - only the intended recipient can open it.

Q7: What is Dynamic we and how it differs from static web?

Ans: **Static Website**

- Think of a static website like a brochure. The content is pre-written and doesn't change unless someone edits the code.
- Every visitor sees the same information, regardless of location or previous interactions with the site.
- Static websites are typically built with HTML, CSS, and Javascript.
- They are faster to load since the content is already prepared.
- Updating a static website requires modifying the code for each page you want to change. This can be cumbersome for large websites.
- Examples of static websites include personal portfolios, simple business websites, or informational pages.

Dynamic Website

- Dynamic websites are more like interactive applications. They can tailor content to the user or update based on information from a database.
- This allows for features like personalized logins, shopping carts, or news feeds that update with new content.
- Dynamic websites often use server-side scripting languages like PHP, Python, or ASP.NET to process information and generate custom content.
- They can be slower to load because the content is assembled on the fly.
- Dynamic websites are easier to update since changes can be made in one central location (like a database) and reflected across the entire website.
- E-commerce stores, social media platforms, and online banking applications are all examples of dynamic websites.

Q8: What is Difference between 3 layers and 3 tier architecture?

Ans:: The terms "3 layers" and "3-tier architecture" are related, but there's a subtle distinction between them:

Layers:

- Represent a functional division of software.
- Focuses on how the application is logically organized.
- A single program can have multiple layers.
- Layers typically communicate with each other directly.

3-Tier Architecture:

- A specific type of software design based on layers.
- Separates the application into three physical tiers that can run on separate machines.
- Offers advantages like scalability, security, and easier maintenance.
- Communication between tiers often happens through well-defined interfaces.

Q9: Which protocol(s) used to access the web?

Ans: The primary protocol used to access the web is **Hypertext Transfer Protocol (HTTP).** HTTP defines the rules for how data is formatted and transmitted between web browsers and web servers. When you enter a website address into your browser, your browser sends an HTTP request to the web server for that address. The server then sends an HTTP response back to your browser, which includes the content of the web page.

In addition to HTTP, other protocols are also involved in web browsing, such as:

- **Transmission Control Protocol (TCP):** TCP ensures reliable data transmission by breaking down data into packets, guaranteeing their order of arrival, and checking for errors.
- **Internet Protocol (IP)**: IP assigns unique addresses (IP addresses) to devices connected to the internet, allowing them to be identified and located for communication.

Q10: Differentiate between Get and Post Method?

Ans: GET and POST are both fundamental methods used in HTTP requests, the language web browsers and servers use to communicate. They seem similar at first glance, but they have distinct purposes:

GET vs. POST: Key Differences

Function:

- **GET:** Used to **retrieve data** from a server. Think of it as asking the server a question.
- **POST:** Used to **send data** to the server, typically to create or update something. Imagine giving instructions or new information to the server.

Data Placement:

- **GET**: Data is appended to the URL in the form of **query parameters**. You've likely seen this when searching the web; the search terms are added to the URL after a question mark.

- **POST:** Data is sent in the **request body**, hidden from the URL. This is often used for form submissions where you fill out data in a web form.

Security:

- **GET**: Less secure because the data is visible in the URL, which can be stored in browser history or bookmarks. Avoid using GET for sensitive information.
- **POST**: More secure because the data is not exposed in the URL.

Caching and Bookmarks:

- **GET:** Requests can be cached by the browser and saved as bookmarks since they are retrieving public information.
- **POST:** Requests are typically not cached and cannot be bookmarked as they are modifying data.

Data Size:

- **GET**: Limited data size due to URL length restrictions.
- **POST:** Can handle larger amounts of data as it's not restricted by URL length.

Here's an analogy: Think of GET like a librarian looking up a book in the catalog (retrieving data), while POST is like giving the librarian a new book to add to the collection (sending data).

Q11: What is default action parameter of form?

Ans: HTML forms themselves don't actually have a default action parameter. The attribute that controls where the form data gets submitted is called action.

This action attribute specifies the URL of the page that will process the form data when the user submits the form. By default, if you don't set this attribute, the form will submit the data to the same page it's located on.

Q12: Why we use XSLT?

Ans: There are a couple of key reasons why **XSLT (eXtensible Stylesheet Language Transformations)** is still used today for working with XML data:

Transforming XML into different formats: XSLT excels at taking structured data in XML format and converting it into a human-readable format like HTML for web pages, but it can also output PDF, plain text, or even create presentations. This is useful because XML itself isn't designed to be easily displayed.

Separation of concerns: XSLT allows you to separate the data (stored in XML) from the presentation (created by XSLT). This makes it easier to maintain and update your data and presentation layers independently. For instance, you can change the way your data is displayed on a webpage without affecting the underlying XML data.

Here are some additional benefits of using XSLT:

Reusable stylesheets: XSLT stylesheets can be reused for multiple XML documents that share a similar structure. This saves time and effort compared to writing custom code for each document.

Data manipulation: XSLT provides functions for sorting, filtering, and performing other manipulations on XML data before it's transformed into the desired output format. This can be useful for presenting the data in a specific way.

Q13: Write down some of the advantages and disadvantages of cookies?

Ans: Cookies, the little bits of data that websites store on your browser, can be both helpful and creepy. Here's a quick rundown of the pros and cons:

Advantages:

Convenience: Cookies can remember things about you, like login information or what you added to your shopping cart. This saves you time from having to re-enter that info all the time.

Personalization: Cookies can track your browsing habits and interests, which allows websites to show you content and ads that are more relevant to you.

Improved browsing experience: Cookies can help websites remember your preferences, such as your preferred language or location. This can make browsing the web a more enjoyable experience.

Disadvantages:

Privacy concerns: Cookies can be used to track your online activity across different websites. This can make some people feel uncomfortable, like they're being constantly watched.

Security risks: In some cases, cookies can be stolen or used by malicious actors to gain access to your personal information.

Targeted advertising: While relevant ads can be nice, seeing the same ad follow you around the web can get annoying.

Q14: What is the use of AJAX?

Ans: AJAX stands for Asynchronous JavaScript and XML. It's a set of techniques used to create more responsive web applications. The key idea is that AJAX allows web pages to update their content without having to reload the entire page. This makes websites feel faster and more interactive for users.

Here are some of the common uses of AJAX:

Updating content without reloading: Imagine you're filling out a form on a website. With AJAX, you can submit the form and have the results displayed on the same page, without having to wait for a new page to load. This is a much smoother experience for the user.

Live chats and messaging: AJAX is often used to power live chat features on websites. This allows users to chat with each other in real time, without having to refresh the page to see new messages.

Social media feeds: Many social media websites use AJAX to update your feed in real time. As new posts are added, they can be added to your feed without you having to refresh the page.

Infinite scrolling: Some websites use AJAX to implement infinite scrolling. This means that as you scroll down the page, new content is automatically loaded without you having to click on a "Next" button.

Q15: Explain the web- architecture?

Ans: Web architecture is essentially the blueprint for how websites and web applications function. It defines how various components work together to deliver what you see on your screen when you visit a website. Here's a breakdown of the key aspects:

Components:

Client: This is the user's side, typically a web browser like Chrome or Firefox. It sends requests to the server and displays the information it receives.

Server: This is the computer or network of computers that stores the website's data and runs its applications. It processes the client's requests and sends back responses.

Network: This is the infrastructure that connects the client and server, like the internet. It allows data to flow between them.

Database: This is where the website's information is stored, like product details for an e-commerce site or articles for a news website.

Communication:

- When you interact with a website, your browser (client) sends a request to the server using a protocol called HTTP (Hypertext Transfer Protocol).
- The server processes the request, retrieves relevant data from the database (if needed), and generates a response.
- This response is sent back to your browser using HTTP, often in the form of HTML (Hypertext Markup Language) that defines the structure of the webpage.
- Your browser interprets the HTML and other elements like CSS (Cascading Style Sheets, for styling) and JavaScript (for interactivity) to render the webpage you see.

Types of Web Architecture:

There are different architectures depending on the website's complexity. Here are two common ones:

Client-Server Model: This is the most basic architecture where the client requests data and the server deliver it.

Multi-Tier Model: This is a more complex architecture where the application is divided into tiers (presentation, business logic, data) for better organization and scalability.

Other Topics

Q: Differentiate between Static and Dynamic Website?

Static Websites

- **Content:** Fixed and unchanging for every visitor. Think of it like a printed brochure.
- **Updates:** Updating content requires modifying the underlying code for each webpage, which can be time-consuming for large websites.
- **Technology:** Relies on HTML, CSS, and JavaScript for basic interactivity.

Advantages:

- Faster loading times due to simpler code.
- Easier and cheaper to create and maintain for small websites.
- More secure as there are no complex server-side processes.

Disadvantages:

- Not ideal for frequently updated content.
- Limited interactivity compared to dynamic websites.
- **Examples**: Simple business websites, online portfolios, brochure websites.

Dynamic Websites

- **Content:** Can change based on user input, time of day, or other factors.
- **Updates:** Easier to update content as changes can be made in one central location (like a database) and reflected across the entire website.
- **Technology:** Relies on server-side scripting languages like PHP, Python, or ASP.NET to generate content on the fly. Often uses databases to store content.

Advantages:

- Ideal for websites with frequently changing content or user interaction.
- Enables features like user logins, shopping carts, and personalized content.

Disadvantages:

- Slower loading times compared to static websites due to server-side processing.
- More complex and expensive to create and maintain.
- Requires additional security measures to protect databases and user information.
- **Examples**: E-commerce websites, social media platforms, news websites, online banking applications.

Q: Differentiate between servlet and java server pages?

Ans: **Purpose:**

- **Servlets:** Pure Java programs that extend the capabilities of web servers. They handle requests, perform business logic, and generate dynamic content. Think of them as the workhorses behind the scenes.
- **JSP (Java Server Pages):** A combination of HTML and Java code. They are easier to read and write compared to pure servlets as they allow you to embed Java code within HTML. JSP pages are primarily for presentation logic, meaning they focus on how data is displayed.

Development Complexity:

- **Servlets:** Require more coding expertise as they involve writing pure Java code.
- **JSP:** Generally considered easier to develop due to the mix of HTML and Java snippets. This makes them a good choice for layouts and user interfaces.

Performance:

- **Servlets:** Execute faster because there's no compilation step involved.
- **JSP:** Slower than servlets because the JSP engine needs to translate the JSP page with embedded Java code into a servlet class before it can be executed.

Q: What is purpose of XSLT?

Ans: XSLT stands for Extensible Stylesheet Language Transformations. It's a language designed to transform XML documents into other formats, including:

Different XML formats

- Human-readable formats like HTML webpages or plain text
- Other document formats like PDF or XSL Formatting Objects (which can be further converted to formats like PDF)

Q: What is meant by a mark-up language?

Ans: A markup language is a system that uses instructions hidden within a text document to define how that text should be formatted and displayed. These instructions are called "tags" and they are not shown in the final version of the document.

Think of it like adding annotations to a recipe. The recipe itself is the content, but you might add tags (like italics or bold) to highlight important instructions or ingredients. In the same way, markup languages use tags to tell a computer program how to display the content, such as headings, paragraphs, or images.

Here are some key points about markup languages:

Structure and Formatting: They are used to define the structure and formatting of a document, such as headings, paragraphs, lists, and images.

Tags: They use special symbols or tags inserted in the document to achieve this. These tags are instructions for the computer program, not part of the content you see yourself.

Human and Computer Readable: They are designed to be readable by both humans and computer programs. The content itself is normal text, and the tags are easy to understand for humans.

Examples: Some of the most common markup languages include HTML (used for web pages), XML (used for data exchange), and LaTeX (used for technical documents).

Q: What are the differences between sessions and cookies?

Ans: Sessions and cookies are both parts of the machinery that keeps websites working smoothly behind the scenes, but they differ in where they store information and how long they last.

Location:

- **Cookies:** These are tiny text files stored on the user's device, like their computer or phone. Whenever you visit a website that uses cookies, information gets added to your cookie stash.
- **Sessions:** On the other hand, sessions live on the server side. The server is basically the computer running the website. Think of it as the website's brain. Sessions store information there temporarily.

Lifespan:

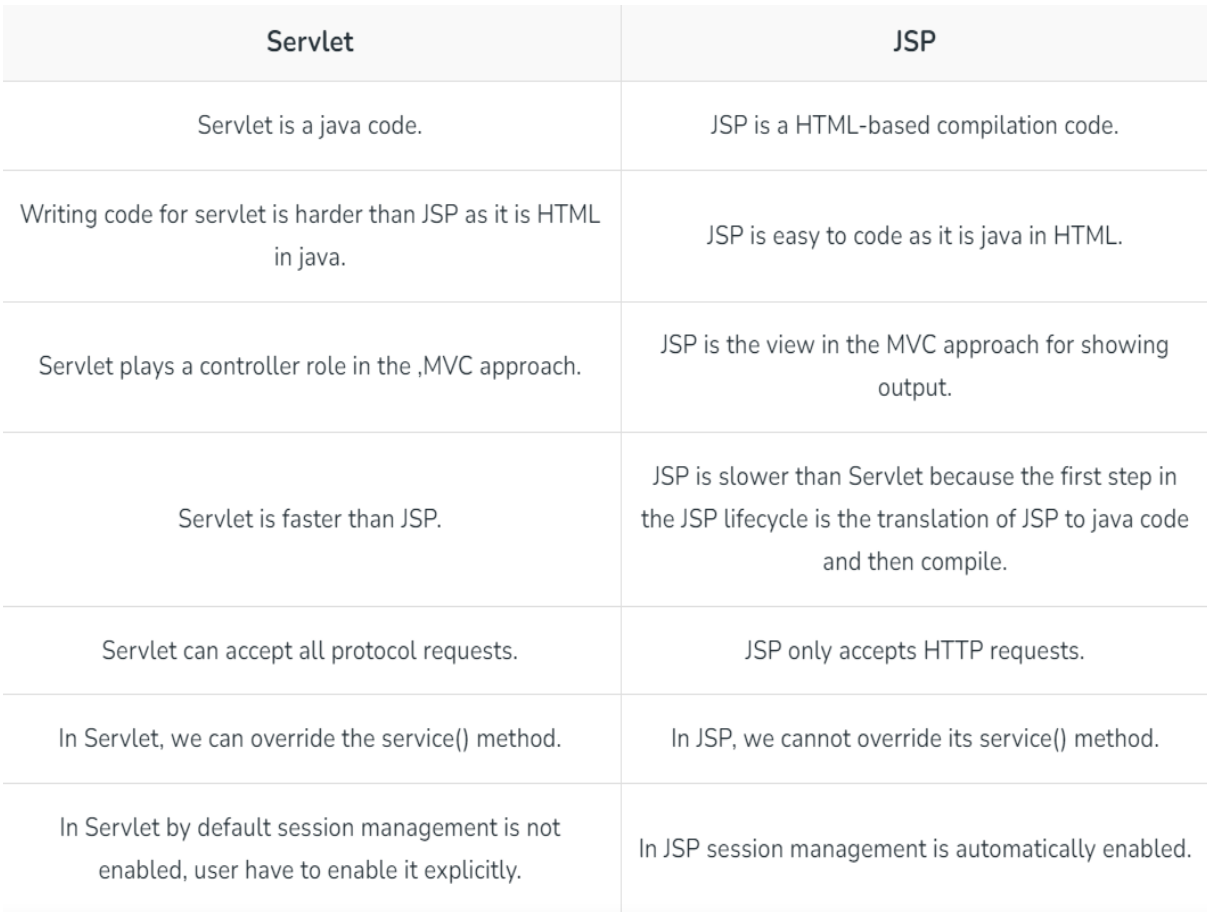
- **Cookies:** These can last for a short time, like just during your current visit to a website, or for a much longer period. It depends on how the website is set up.
- **Sessions:** Sessions generally disappear once you close your browser window. That's because the server forgets about the session that was associated with you.

Data Storage:

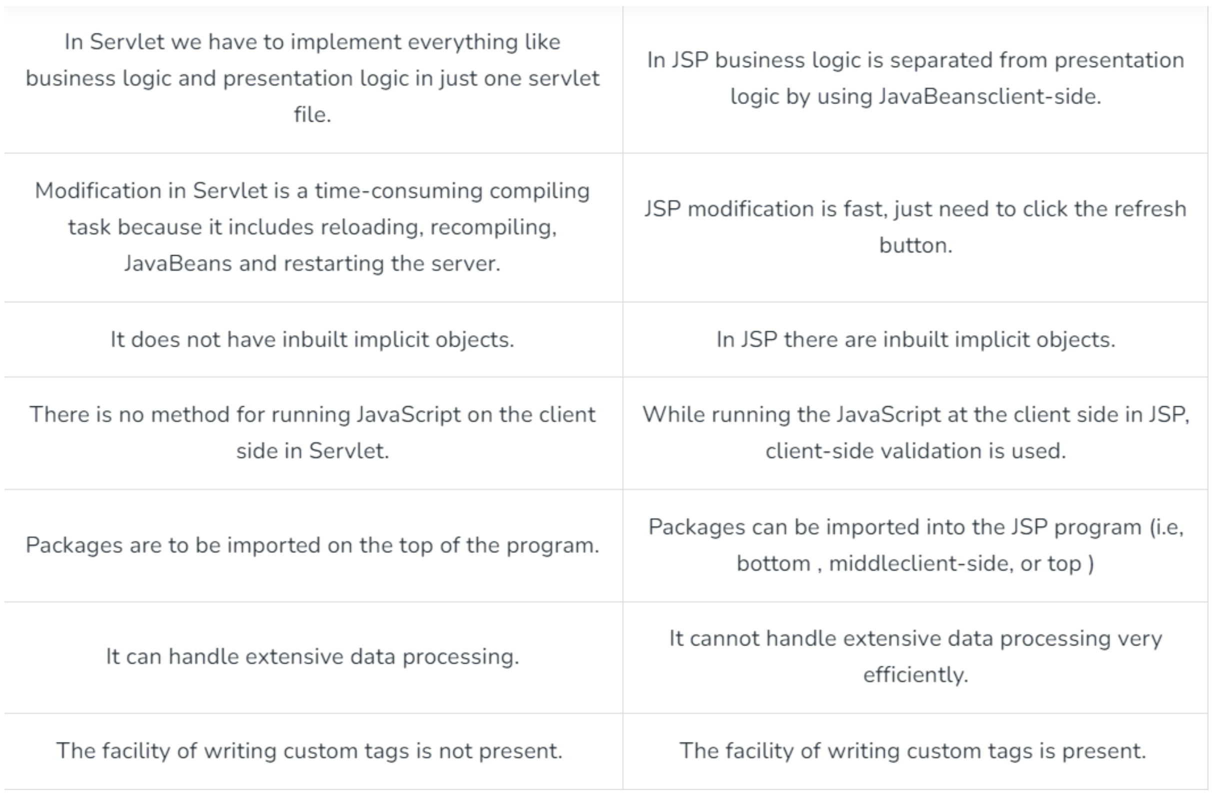
- **Cookies:** Cookies can only hold a small amount of data, usually up to 4 kilobytes (KB). They're good for things like remembering your login information or what you added to your shopping cart.
- **Sessions:** Sessions can hold a lot more data compared to cookies. This lets them store more complex information about your activity on a website.

Q: What are the differences between a servlet and JSP?

Ans:



Servlet	JSP
Servlet is a java code.	JSP is a HTML-based compilation code.
Writing code for servlet is harder than JSP as it is HTML in java.	JSP is easy to code as it is java in HTML.
Servlet plays a controller role in the ,MVC approach.	JSP is the view in the MVC approach for showing output.
Servlet is faster than JSP.	JSP is slower than Servlet because the first step in the JSP lifecycle is the translation of JSP to java code and then compile.
Servlet can accept all protocol requests.	JSP only accepts HTTP requests.
In Servlet, we can override the service() method.	In JSP, we cannot override its service() method.
In Servlet by default session management is not enabled, user have to enable it explicitly.	In JSP session management is automatically enabled.



In Servlet we have to implement everything like business logic and presentation logic in just one servlet file.	In JSP business logic is separated from presentation logic by using JavaBeansclient-side.
Modification in Servlet is a time-consuming compiling task because it includes reloading, recompiling, JavaBeans and restarting the server.	JSP modification is fast, just need to click the refresh button.
It does not have inbuilt implicit objects.	In JSP there are inbuilt implicit objects.
There is no method for running JavaScript on the client side in Servlet.	While running the JavaScript at the client side in JSP, client-side validation is used.
Packages are to be imported on the top of the program.	Packages can be imported into the JSP program (i.e, bottom , middleclient-side, or top)
It can handle extensive data processing.	It cannot handle extensive data processing very efficiently.
The facility of writing custom tags is not present.	The facility of writing custom tags is present.

Q: What do you mean by request dispatcher and send redirection?

Ans: **Request Dispatcher (forward):**

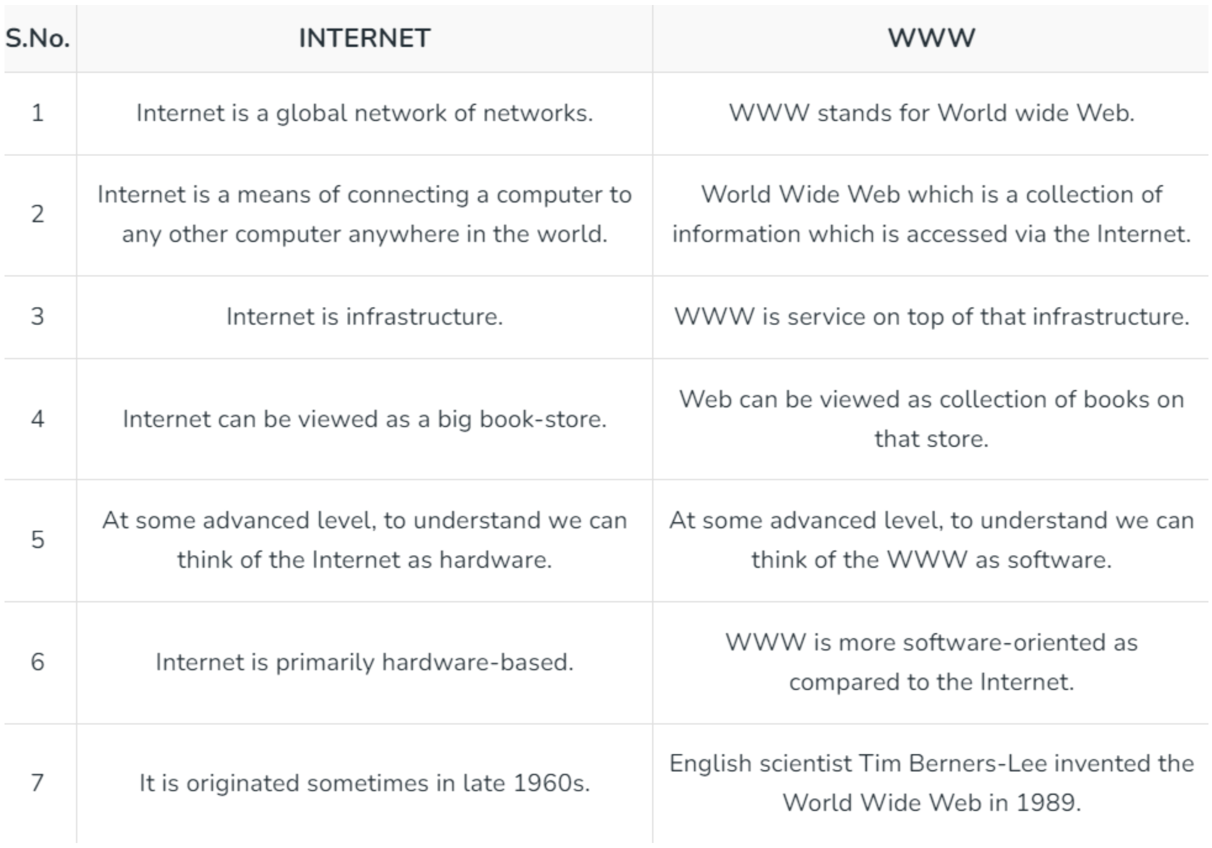
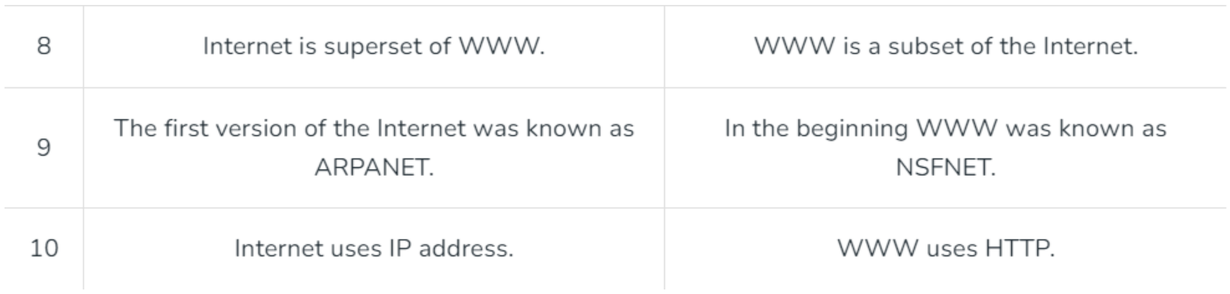
- **Server-side:** Works entirely on the server. The web container manages the flow without involving the user's browser.
- **Single request:** The original user request is forwarded to another resource (like a servlet or JSP) within the same server for further processing.
- **Transparent:** The user's browser address bar remains unchanged, making the redirection invisible.
- **Faster:** Since it's handled internally, it's generally faster than redirection.
- **Request object preserved:** The original request object is available to the forwarded resource.

Send Redirection:

- **Client-side:** Involves the user's browser. The server sends a response with a status code (like 302 Found) instructing the browser to make a new request to a different URL.
- **Two requests:** A new request is initiated by the browser to the redirected URL.
- **Visible:** The user's browser address bar updates to show the new URL.
- **Slower:** Requires an extra round trip between the browser and server, making it slower.
- **Request object lost:** The original request object isn't accessible on the redirected resource.

Q: Difference between WWW and Internet?

Ans:

S.No.	INTERNET	WWW
1	Internet is a global network of networks.	WWW stands for World wide Web.
2	Internet is a means of connecting a computer to any other computer anywhere in the world.	World Wide Web which is a collection of information which is accessed via the Internet.
3	Internet is infrastructure.	WWW is service on top of that infrastructure.
4	Internet can be viewed as a big book-store.	Web can be viewed as collection of books on that store.
5	At some advanced level, to understand we can think of the Internet as hardware.	At some advanced level, to understand we can think of the WWW as software.
6	Internet is primarily hardware-based.	WWW is more software-oriented as compared to the Internet.
7	It is originated sometimes in late 1960s.	English scientist Tim Berners-Lee invented the World Wide Web in 1989.
8	Internet is superset of WWW.	WWW is a subset of the Internet.
9	The first version of the Internet was known as ARPANET.	In the beginning WWW was known as NSFNET.
10	Internet uses IP address.	WWW uses HTTP.

Q: Write down a life cycle of a JSP Page?

ANS: The life cycle of a JSP page governs its entire process, from its creation to when it's no longer needed. It closely resembles the life cycle of a servlet, with an extra step for translation. Here's a breakdown of the key stages:

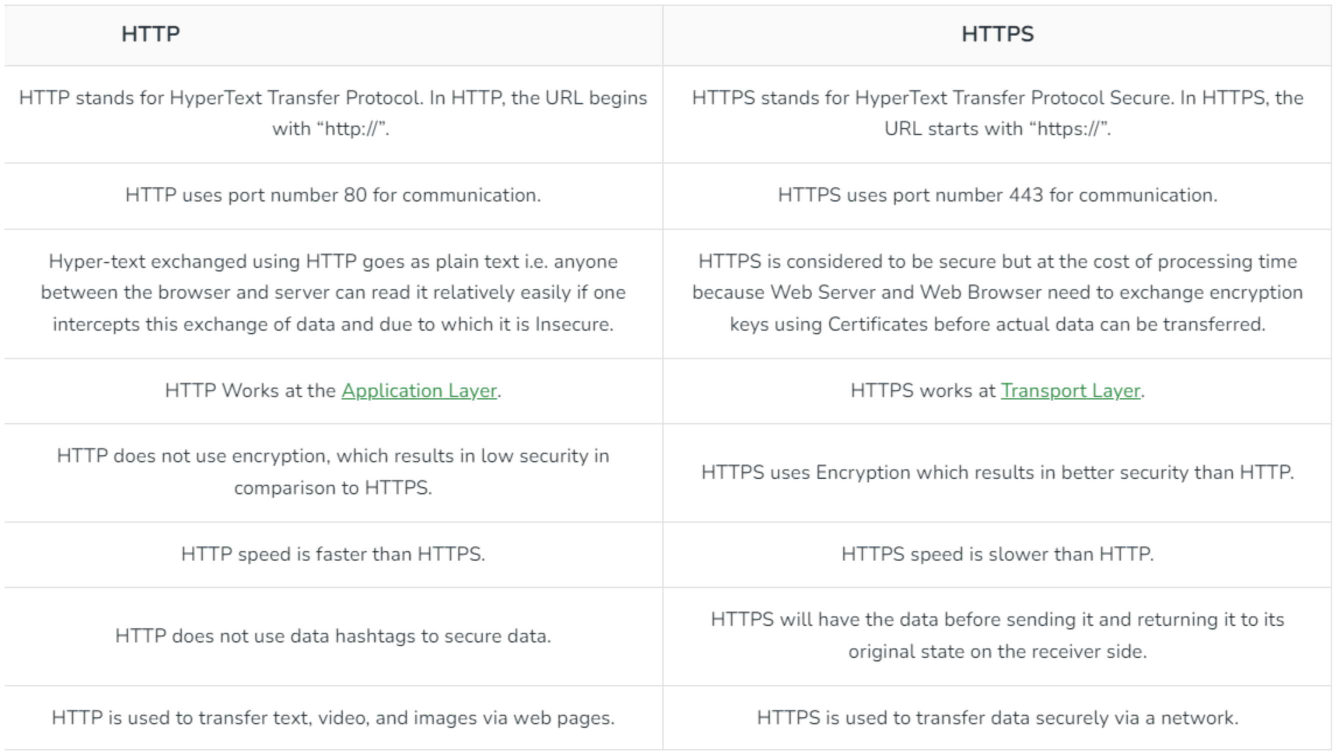
- **Translation:** When a JSP page is first requested, the web container's JSP translator kicks in. This translator transforms the JSP page's code, including HTML tags and JSP elements, into a servlet class.
- **Compilation:** Following translation, the web container compiles the generated servlet class into a bytecode file (class file). This bytecode can be directly executed by the Java Virtual Machine (JVM).
- **Class loading**: The class file is then loaded by the web container's class loader. This makes the compiled servlet class accessible for further processing.
- **Instantiation:** An instance of the servlet class is created. This creates an object representation of the JSP page in memory.

- **Initialization (jspInit()):** The container invokes the jspInit() method of the servlet class. This method is used for initialization tasks specific to the JSP page, like establishing database connections or loading resources.
- **Request Processing (_jspService()):** This is the heart of the JSP page's functionality. The container calls the _jspService() method to handle incoming client requests. Here, the JSP page interacts with various resources like databases, session objects, and request parameters to generate dynamic content.

Q: What are the components of Java Beans?

Ans: JavaBeans are reusable components in Java that follow specific design conventions. These conventions allow for easy integration with development tools and frameworks. Here are the key components of a JavaBean:

- **Properties:** These represent the data or state of the bean. Properties have getter and setter methods, also known as accessor and mutator methods, to control access and modification of the data.
- **Methods**: Similar to regular Java methods, these define the functionalities of the bean. Unlike properties, methods don't have a specific naming convention.
- **Events:** This mechanism allows beans to communicate with other objects by notifying them of specific occurrences. JavaBeans components can utilize event handling similar to Swing in Java.
- **Persistence:** This optional feature enables JavaBeans to store their state for later retrieval. This is useful for maintaining data across program sessions.



HTTP	HTTPS
HTTP stands for HyperText Transfer Protocol. In HTTP, the URL begins with “http://”.	HTTPS stands for HyperText Transfer Protocol Secure. In HTTPS, the URL starts with “https://”.
HTTP uses port number 80 for communication.	HTTPS uses port number 443 for communication.
Hyper-text exchanged using HTTP goes as plain text i.e. anyone between the browser and server can read it relatively easily if one intercepts this exchange of data and due to which it is Insecure.	HTTPS is considered to be secure but at the cost of processing time because Web Server and Web Browser need to exchange encryption keys using Certificates before actual data can be transferred.
HTTP Works at the Application Layer.	HTTPS works at Transport Layer.
HTTP does not use encryption, which results in low security in comparison to HTTPS.	HTTPS uses Encryption which results in better security than HTTP.
HTTP speed is faster than HTTPS.	HTTPS speed is slower than HTTP.
HTTP does not use data hashtags to secure data.	HTTPS will have the data before sending it and returning it to its original state on the receiver side.
HTTP is used to transfer text, video, and images via web pages.	HTTPS is used to transfer data securely via a network.