**Design and Rendering patterns**

* **Design Patterns:**

Design patterns are ready-made solutions that software developers use to solve common problems when creating any computer program. These patterns make software stronger, easier to maintain, and capable of handling more tasks and hence reduce its complexity. They are organized into types focusing on specific aspects, like creating objects or managing how different parts of the program work together. In simple terms, design patterns help developers build software in a more organized and efficient way.

The difference types of Design patterns are :

1. Creational design pattern
2. Structural design pattern
3. Behavioral design pattern

**1.Creational Patterns:**

Creational design patterns focus on the process of object creation, providing mechanisms for class instantiation. They address how objects are created and help in composing systems in a way that is flexible, modular, and extensible.

**2.Structural Patterns**

Structural design patterns concentrate on the composition of classes and objects to form larger structures. They define ways to compose classes and objects to create more complex and adaptable systems.

**3.Behavioral Patterns**:

Behavioral design patterns revolve around the interaction and communication between objects. They define patterns for the assignment of responsibilities between objects, simplifying the communication and collaboration within a system.

**Rendering patterns:**

Rendering refers to the various ways of converting data into images or visuals through which we can demonstrate or present it. The idea is on how to make information look attractive and used to understand.

The important rendering patterns are as follows:

**1.Client-Side Rendering (CSR):**

Client-Side Rendering is a web rendering technique where the browser takes on the responsibility of rendering the user interface. The initial HTML is minimal, and the client fetches and processes JavaScript to dynamically update and render content on the client side. CSR is well-suited for dynamic and interactive web applications.

**For example: React framework uses CSR rendering pattern .**

**ample:** React framework uses CSR rendering pattern for delivering dynamic and interactive web pages. It uses components which make web pages reload partially without affecting the entire web page.

**2.Server-Side Rendering (SSR)**

Server-Side Rendering is a web rendering approach where the server generates the complete HTML for a page and sends it to the client. This results in a fully rendered page delivered to the browser, providing faster initial page loads. SSR is particularly beneficial for content-heavy websites and those prioritizing search engine optimization (SEO).

For example:

**For example:** Java servlets uses SSR rendering pattern.Java servlets are like containers which create html pages on servlets based on user request.

**3.Static Site Generation (SSG):**

Definition: Static Site Generation is a method of pre-rendering web pages during the build process. The server generates HTML files for each page at build time, allowing them to be served statically. SSG is suitable for websites with relatively stable content, such as blogs or documentation sites, offering fast loading times and efficient content delivery.

**For example**: Blogger is one of the most popular blogging websites which uses a SSG rendering pattern to provide users to create static web pages. This static web pages do not offer much interaction

The choice of selecting these patterns completely depends on the content dynamics,performance need and overall goals.