**LỘ TRÌNH HỌC**

# **Quy trình học**

1. **JSP (JavaServer Pages)**:
   * **Đơn giản là gì**: JSP là một cách để tạo ra trang web mà có thể thay đổi nội dung bên trong trang đó một cách linh hoạt. Nó giống như trang HTML, nhưng bạn có thể viết một ít mã Java vào để làm trang trở nên "thông minh" hơn.
   * **Ví dụ**: Giả sử bạn có một trang chào mừng. Nếu bạn dùng HTML thông thường, trang chỉ hiển thị "Chào bạn!". Nhưng với JSP, bạn có thể viết một đoạn mã để hiển thị "Chào [Tên người dùng]!" (tên thay đổi dựa trên người dùng nào đang truy cập).
   * **Học gì**: Bạn sẽ học cách viết mã Java trong các trang web, cách lấy thông tin từ người dùng và hiển thị thông tin trên trang.
2. **Java Servlet**:
   * **Đơn giản là gì**: Servlet là một chương trình Java nằm trên máy chủ (server), chịu trách nhiệm xử lý các yêu cầu từ người dùng và trả kết quả về. Nó giống như một "người phục vụ" cho các yêu cầu từ trang web, ví dụ như khi người dùng bấm nút "Gửi" trên biểu mẫu (form), Servlet sẽ nhận và xử lý yêu cầu đó.
   * **Ví dụ**: Khi bạn đăng nhập vào một trang web, dữ liệu (như tên đăng nhập và mật khẩu) sẽ được gửi đến máy chủ. Một Servlet sẽ nhận thông tin này, kiểm tra trong cơ sở dữ liệu xem thông tin có đúng không, rồi gửi lại kết quả như "Đăng nhập thành công" hoặc "Sai mật khẩu".
   * **Học gì**: Bạn sẽ học cách làm việc với các yêu cầu từ trang web, xử lý dữ liệu và trả lại kết quả cho trang.
3. **Spring MVC**:
   * **Đơn giản là gì**: Spring MVC là một công cụ mạnh mẽ giúp bạn tổ chức một ứng dụng web phức tạp một cách dễ quản lý và dễ nâng cấp. Nó chia ứng dụng thành ba phần: **Model** (nơi lưu dữ liệu), **View** (giao diện hiển thị cho người dùng), và **Controller** (điều khiển, xử lý yêu cầu từ người dùng).
   * **Ví dụ**: Khi bạn truy cập vào một trang web mua hàng, **Controller** sẽ lấy danh sách sản phẩm từ **Model** (cơ sở dữ liệu), sau đó gửi danh sách này cho **View** để hiển thị ra màn hình.
   * **Học gì**: Bạn sẽ học cách tổ chức mã nguồn rõ ràng hơn, làm việc với dữ liệu dễ dàng hơn, và làm cho ứng dụng trở nên chuyên nghiệp, dễ bảo trì.

**Tóm gọn lại:**

* **JSP**: Dùng để tạo trang web có thể thay đổi nội dung dựa trên dữ liệu từ máy chủ.
* **Java Servlet**: Dùng để xử lý các yêu cầu từ người dùng, như gửi thông tin đăng nhập hoặc yêu cầu dữ liệu.
* **Spring MVC**: Một bộ công cụ giúp bạn tạo ứng dụng web một cách có tổ chức và dễ dàng quản lý khi dự án lớn lên.

Nếu bạn là người mới, nên bắt đầu học **JSP** trước để quen với việc làm trang web, sau đó học **Servlet** để hiểu cách xử lý yêu cầu từ người dùng, và cuối cùng là **Spring MVC** để làm ứng dụng phức tạp hơn.

# **Spring MVC**

Spring MVC là một phần của Spring Framework, chính vì thế bạn nên hiểu rõ về Spring Framework trước khi học chi tiết về Spring MVC. Spring Framework là một nền tảng phát triển [ứng dụng Java](https://hanoi.codegym.vn/blog/tim-hieu-java-la-gi/" \t "_blank) được thiết kế, để giúp cho việc phát triển ứng dụng Java dễ dàng hơn và nhanh hơn. Bao gồm: một bộ sưu tập các thành phần và module, được thiết kế để hỗ trợ cho việc xây dựng các ứng dụng Java từ các ứng dụng desktop đơn giản đến các ứng dụng web phức tạp.

Spring Framework cung cấp nhiều tính năng hữu ích cho phát triển ứng dụng như: Inversion of Control (IoC), Dependency Injection (DI), Aspect-Oriented Programming (AOP), Spring Data và Spring MVC (Model-View-Controller). Các tính năng này giúp cho việc phát triển ứng dụng trở nên dễ dàng và giảm thiểu sự phụ thuộc giữa các thành phần trong ứng dụng.