**Task3: Tìm hiểu về Function(Hàm) trong Java**

### 1. Function(Hàm) là gì?

**Function** (hàm, chức năng), gọi chung là **subprogram (chương trình con)**có thể được gọi ở bên ngoài hoặc bên trong chính nó.

Nó bao gồm tập hợp các câu lệnh gọi là **function** **body.**Các giá trị có thể truyền đến một hàm, và một hàm có thể trả về giá trị.

Bây giờ, với các ứng dụng hiện đại, các function có thể là một chương trình hoàn chỉnh, chứ không phải là khái niệm tổng quát như ''subprogram" nữa. Có sự khác nhau giữa function và procedure (thủ tục) rằng sự lý tưởng của function nên trả về một giá trị còn procedure thì không ( bây giờ điều này có thể thay đổi theo ngôn ngữ lập trình).

**1.1 Khai báo hàm**

Để khai báo hàm, chúng ta dùng từ khóa function, theo sau nó là:

+ Tên hàm

+ Danh sách các tham số truyền vào hàm, được đặt trong ngoặc đơn và cách nhau bởi dấu phẩy.

+ Các câu lệnh của JavaScript để tạo ra một hàm, được đặt trong ngoặc nhọn {...}.

**1.2 Biểu thức hàm(Hàm trong biến)**

Trong khi việc khai báo hàm ở trên là một câu lệnh về mặt cú pháp, các hàm cũng có thể tạo ra bằng một biểu thức hàm (**function expression**). Một hàm như vậy có thể **nặc danh**; nó không cần phải có tên. Ví dụ, hàm sayHello ở trên có thể được khai báo như sau:

function sayHello () {

  console.log("Hello World");

}

Tuy nhiên, một cái tên *có thể* được cung cấp trong một biểu thức hàm. Việc cung cấp tên cho phép hàm có thể chạy chính nó, hoặc có thể sử dụng hệ thống debug để nhận dạng hàm trong stack traces.

**2. Thao tác với Hàm(method)**

Khi chúng ta đã viết xong một hàm rồi làm sao để chúng ta có thể sử dụng chúng đây ta! Trước tiên chúng ta sẽ có 2 loại hàm, mỗi loại sẽ có cách gọi khác nhau:

* Hàm có trả về kết quả
* Hàm không trả về kết quả

**2.1 Hàm có trả về kết quả**

Đối với hàm có kết quả trả về, chúng ta cần dùng từ khoá return để trả về kết qủa mà nó đã tính toán được.

Khai báo biến có kiểu dữ kiệu tương ứng với kết quả trả về của hàm để nhận giá trị trả về.

Như ví dụ tìm sum() của chúng ta, nhiệm vụ của nó là tính tổng của 2 số nguyên a, b nhập vào.

public class Main {

    public static int sum(int a, int b) {

        return a + b;

    }

    public static void main(String[] args) {

        int sum = sum(2,5);

        System.out.println(sum);

    }

}

Chúng ta sẽ dùng biến có kiểu dữ liệu tương ứng để nhận kết quả trả về từ hàm. Hàm **sum()** trả về**int**chúng ta dùng biến**int sum để**nhận kết quả.

**2.2 Hàm không trả về kết quả**

Đối với loại hàm này chúng ta chỉ cần gọi để sử dụng.   
public static void printHello() {

        System.out.println("Hello");

}

Hàm **printHello()** là một hàm không có kiểu trả về nên được thay thế thành **void.**Từ khoá void cho chúng ta biết là method đó sẽ không có kết quả trả về.

**2.3 Public, Satic(modifier)**

* Public: bất kỳ một method hay một phương thức nào có từ khoá public đứng trước thì nó sẽ được truy cập bất cứ ở đâu. Nghĩa là nếu như bạn đang ở một class khác, một package khác,… thì bạn đều có thể gọi đến nó.
* Static: cho phép rằng biến hay phương thức chỉ được khởi tạo lúc tải class. Chúng ta chỉ cần gọi method chứ không khởi tạo class chứa nó. Trong loạt bài java core chúng ta chỉ cần code trong class Main. Nếu bỏ static trong khai báo hàm đi thì cần phải khởi tạo class Main rồi mới gọi hàm.

**2.4 Truyền tham số khi gọi hàm**

Khi bạn gọi đến một hàm đã được định và nó có danh sách các thông số nhất định. Thì khi gọi các bạn sẽ phải truyền đúng thứ tự và kiểu dữ liệu cho từng tham số. Nếu sai xót trong quá trình này có thể dẫn đến biên dịch lỗi hoặc là kết quả không đúng như mong muốn.