



28TECH

Become A Better Developer



HÀM (FUNCTION)



NỘI DUNG BÀI HỌC:

1

Định nghĩa hàm

2

Cú pháp

3

Gọi hàm

4

Tham số và GT trả về

5

Những chú ý

6

Nạp chồng hàm

7

Các hàm toán học



1. Định nghĩa hàm (Function):



Phần lớn các chương trình máy tính được xây dựng để **giải quyết các bài toán lớn** trong thực tế.



Cách tốt nhất để xây dựng và bảo trì một chương trình đó là xây dựng nó từ những **thành phần nhỏ** được xây dựng đơn lẻ.



Chức năng Hàm (Function) được sử dụng để **chia nhỏ chương trình** thành các thủ tục nhỏ giải quyết từng chức năng nhỏ.



1. Định nghĩa hàm (Function):



2. Cú pháp (Syntax):

CÚ PHÁP

```
public static returnType functionName (parameter1, parameter2,...) {  
    // function body  
}
```

- **returnType**: Kiểu trả về của hàm ví dụ : void, bool, int, long long...
- **functionName**: Tên hàm
- **parameter**: Tham số của hàm
- **function body**: Các câu lệnh bên trong của hàm



2. Cú pháp (Syntax):

VÍ DỤ

Kiểu trả về **void**, tức hàm này **không** trả về giá trị nào

xinChao: tên hàm

Hàm này **không** có tham số

EXAMPLE

```
public static void xinChao(){  
    System.out.println("Hello !");  
}
```

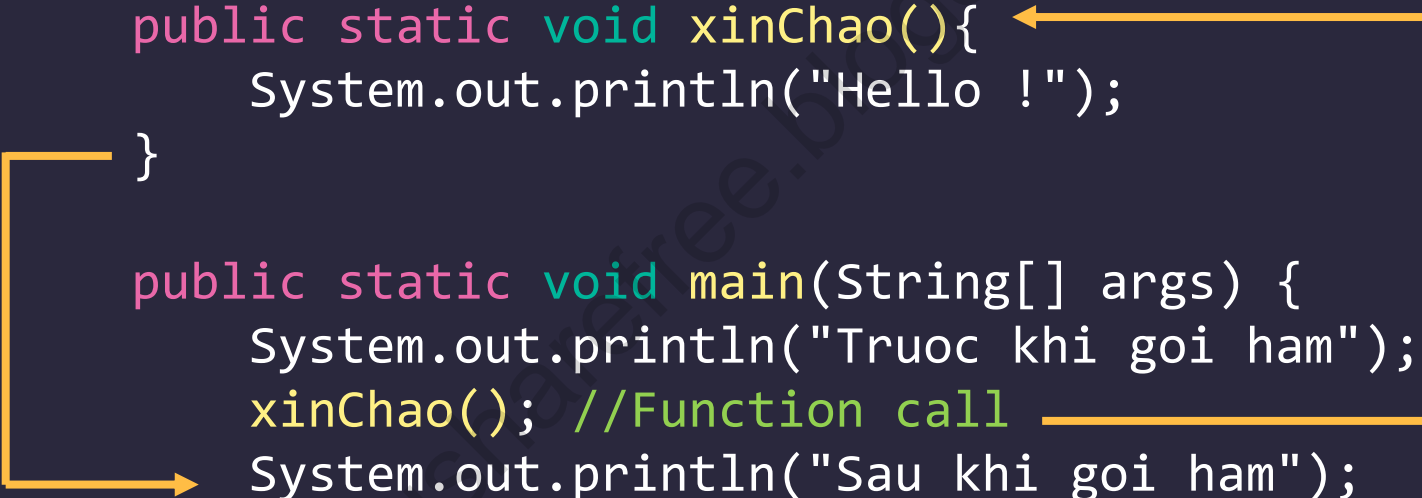




3. Gọi hàm (Function call):

EXAMPLE

```
public class Main {  
    public static void xinChao(){  
        System.out.println("Hello !");  
    }  
  
    public static void main(String[] args) {  
        System.out.println("Truoc khi goi ham");  
        xinChao(); //Function call  
        System.out.println("Sau khi goi ham");  
    }  
}
```





4. Tham số và giá trị trả về (Parameter & return type):

a. Tham số và đối số:





4. Tham số và giá trị trả về (Parameter & return type):

a. Tham số và đối số:

EXAMPLE

```
public class Main {  
    public static void printNumber(int a){  
        System.out.println(a);  
    }  
    public static void main(String[] args){  
        int n = 10;  
        printNumber(n);  
    }  
}
```

OUTPUT

10

Giải thích

- Khi bạn gọi hàm printNumber(n) thì n được gọi là **Đối số** hay **tham số thực sự**, còn a trong **tham số hình thức**.
- Giá trị của n sẽ được gán cho a, vì thế a cũng sẽ có giá trị là 10, và khi câu lệnh bên trong hàm printNumber thực hiện màn hình sẽ hiển thị giá trị 10.





4. Tham số và giá trị trả về (Parameter & return type):

b. Kiểu trả về:



Khi chuẩn bị xây dựng một hàm, các bạn cần xác định kiểu trả về :

Nếu hàm không cần trả về giá trị gì, kiểu trả về sẽ là void.



Nếu hàm cần trả về giá trị đúng hoặc sai, kiểu trả về là boolean.



Nếu hàm cần trả về số nguyên thì kiểu trả về là int hoặc long tùy vào độ lớn.



Nếu hàm cần trả về số thực thì kiểu trả về là float hoặc double tùy vào độ chính xác mong muốn.





5. Những chú ý khi xây dựng hàm:



Hàm này có cần trả về giá trị không, nếu có thì trả về kiểu dữ liệu là gì ?



Hàm này có bao nhiêu tham số, các tham số có kiểu dữ liệu là gì ?



Hàm của bạn xây dựng đã đủ tổng quát chưa hay quá chi tiết và chỉ phù hợp cho 1 bài toán cụ thể ?



Bạn gọi hàm có đúng thứ tự tham số mà mình mong muốn hay không, kiểu dữ liệu của tham số hình thức và tham số chính thức có hợp lí hay không?





6. Nạp chồng hàm (Function Overloading):



Nạp chồng hàm là một tính năng của Java khi mà một hàm có cùng tên nhưng khác nhau về tham số, kiểu trả về. Khi hàm được gọi thì tùy vào kiểu dữ liệu của tham số, hàm phù hợp sẽ được gọi.

EXAMPLE

```
public class Main {  
    public static int findMax(int a, int b) {  
        if (a > b) {  
            return a;  
        } else {  
            return b;  
        }  
    }  
    public static float findMax(float a, float b) {  
        if (a > b) {  
            return a;  
        } else {  
            return b;  
        }  
    }  
    public static void main(String[] args) {  
        System.out.println(findMax(10, 20));  
        System.out.println(findMax(10.2f, 30.4f));  
    }  
}
```

OUTPUT

20
30.4



7. Các hàm toán học phổ biến trong Java:

Hàm	Chức năng	Ví dụ
max	Tìm giá trị lớn nhất	<code>System.out.println(Math.max(10, 20));</code>
min	Tìm giá trị nhỏ nhất	<code>System.out.println(Math.min(10, 20));</code>
sqrt(n)	Hàm tính căn bậc 2 của n, trả về số double	<code>System.out.println(Math.sqrt(100));</code>
pow(x,y)	Hàm tính x^y , trả về số double	<code>System.out.println(Math.pow(2, 10));</code>
ceil(n)	Hàm trả về số nguyên lớn hơn gần nhất với n	<code>System.out.println(Math.ceil(20.3));</code>
floor(n)	Hàm trả về số nguyên nhỏ hơn gần nhất với n	<code>System.out.println(Math.floor(20.3));</code>
round(n)	Làm tròn n tùy thuộc vào phần thập phân	<code>System.out.println(Math.round(20.8));</code>
abs(n)	Trả về trị tuyệt đối của n	<code>System.out.println(Math.abs(-100));</code>
cbrt(n)	Trả về căn bậc 3 của n	<code>System.out.println(Math.cbrt(27));</code>
PI	Số Pi	<code>System.out.println(Math.PI);</code>
E	Số E	<code>System.out.println(Math.E);</code>