

# Data type

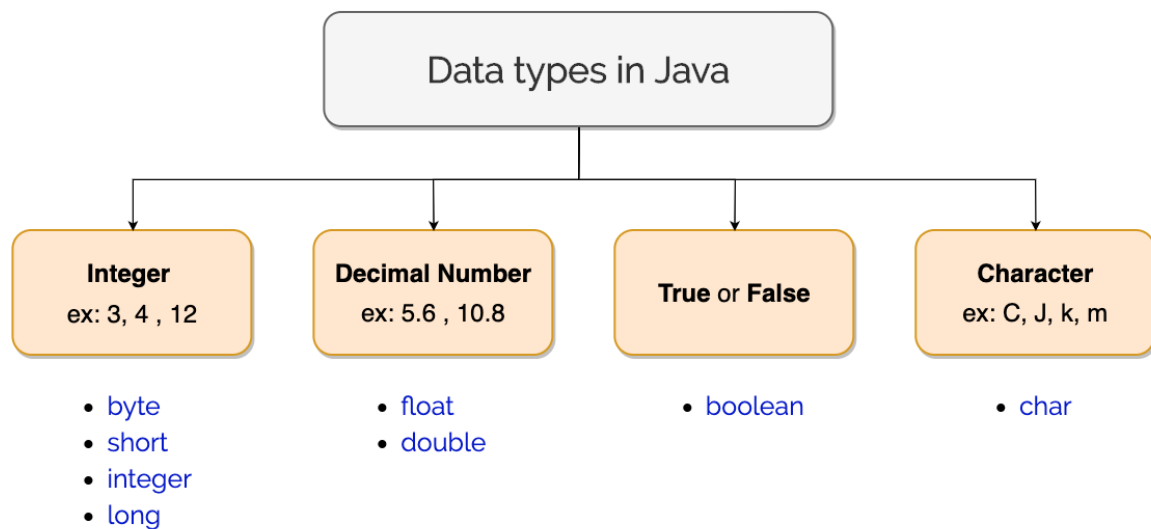
Tags

Done

## Data type - Kiểu dữ liệu trong java

Trong 3 java có 3 loại data type:

- Primitive Data Types
- Wrapper Data Types
- Reference Data Types



## Primitive type

- **Integer types - Kiểu dữ liệu số nguyên**
  - **byte**: Dùng để lưu trữ số nguyên nhỏ. Kích thước là 8 bit.
  - **short**: Dùng để lưu trữ số nguyên nhỏ hơn. Kích thước là 16 bit.

- **int** : Dùng để lưu trữ số nguyên. Kích thước là 32 bit.
- **long** : Dùng để lưu trữ số nguyên lớn. Kích thước là 64 bit.

## DATA TYPES: PRIMITIVE DATA TYPES

Java's Primitive Data Types				
Type	Name	Byte	Bit	Range
Integer	byte	1	8	-128..127
	short	2	16	-32768..32767
	int	4	32	-2147483648..2147483647
	long	8	64	-9223372036854775808..9223372036854775807

**Memory Storage::**

Bit <--> Binary Digit (0 or 1)      Byte == 8 Bits

1 == Negative  
0 == Positive

1

1

1

1

1

1

1

1

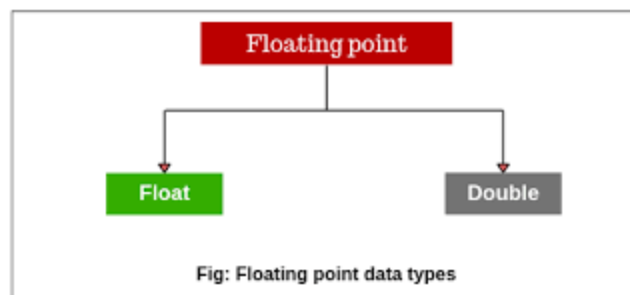
== -1

Sign Bit

Signed Integer

- **Decimal Number** - Kiểu dữ liệu số thực

- **float** : Dùng để lưu trữ số thực với độ chính xác thấp. Kích thước là 32 bit.
- **double** : Dùng để lưu trữ số thực với độ chính xác cao. Kích thước là 64 bit.



- **Character type** - Kiểu ký tự

- **char** : Dùng để lưu trữ một ký tự Unicode. Kích thước là 16 bit.

- **Boolean type** - Kiểu boolean
  - `boolean` : Dùng để lưu trữ giá trị true hoặc false.

## Wrapper type

Tương tự như primitive type thì Wrapper type cũng có các kiểu dữ liệu tương tự

Primitive Data Type	Wrapper Class
<code>char</code>	<code>Character</code>
<code>byte</code>	<code>Byte</code>
<code>short</code>	<code>Short</code>
<code>int</code>	<code>Integer</code>
<code>long</code>	<code>Long</code>
<code>float</code>	<code>Float</code>
<code>double</code>	<code>Double</code>
<code>boolean</code>	<code>Boolean</code>

## Sự khác nhau giữa primitive và wrapper

Tính Chất	Kiểu Dữ Liệu Nguyên Thủy	Kiểu Dữ Liệu Bọc
Loại Dữ Liệu	Các kiểu dữ liệu cơ bản như <code>int</code> , <code>double</code> , <code>boolean</code> , v.v.	Là các lớp đóng gói cho các kiểu nguyên thủy như <code>Integer</code> , <code>Double</code> , <code>Boolean</code> , v.v.
Tham Số Hóa (Generic)	Không thể sử dụng trong generic programming.	Có thể sử dụng trong generic programming để làm việc với các cấu trúc dữ liệu tổng quát.

Giá Trị Rỗng (Null Values)	Không thể biểu diễn giá trị rỗng (null).	Có thể biểu diễn giá trị rỗng (null), cho phép xác định các trường hợp không có giá trị.
Trường Hợp Sử Dụng	Sử dụng khi cần hiệu suất cao và ít tài nguyên.	Sử dụng khi cần làm việc với đối tượng, generic programming, hoặc khi cần biểu diễn giá trị rỗng.
Sự Kích Thước Trong Bộ Nhớ	Dung lượng bộ nhớ cố định, không thay đổi.	Tốn thêm một lượng nhỏ bộ nhớ để lưu trữ tham chiếu đến đối tượng.
Sự Không An Toàn (Unsafe)	Có thể dẫn đến lỗi NPE (NullPointerException) nếu không kiểm tra giá trị null.	Có thể tránh được lỗi NPE bằng cách kiểm tra giá trị null trước khi sử dụng.
Chuyển Đổi Tự Động (Autoboxing/Unboxing)	Không tự động chuyển đổi giữa kiểu nguyên thủy và kiểu bọc.	Có tự động chuyển đổi giữa kiểu nguyên thủy và kiểu bọc.
Tên Các Hằng Số (Constants)	Dùng tên các hằng số như <code>int MAX_VALUE = 10;</code>	Dùng tên các hằng số của kiểu dữ liệu bọc như <code>Integer.MAX_VALUE</code> .

## Reference Data Type

Một số kiểu dữ liệu tham chiếu trong Java:

- **Lớp (Class):** Đây là loại kiểu dữ liệu tham chiếu cơ bản nhất.

```
class Person {
    String name;
    int age;
}

// Tạo một đối tượng Person
Person person = new Person();
```

- **Mảng (Array):** Mảng là một kiểu dữ liệu tham chiếu, cho phép bạn lưu trữ một tập hợp các giá trị cùng kiểu dữ liệu trong một cấu trúc dữ liệu.

```
int[] numbers = new int[5]; // Tạo một mảng kiểu int
```

- **Giao Diện (Interface):** Giao diện là một kiểu dữ liệu tham chiếu đặc biệt trong Java.

```
interface Shape {  
    double area();  
    double perimeter();  
}
```

- **Chuỗi:**

```
String greeting = "Hello, World!";
```

- **Lớp Của Thư Viện (Library Classes):** Java đi kèm với nhiều thư viện chuẩn, và bạn có thể sử dụng các lớp từ các thư viện này để làm việc với các đối tượng như `ArrayList`, `HashMap`, `Socket`, v.v.