

# Kiểu dữ liệu trong java

Một ngôn ngữ lập trình nào cũng có một tập các kiểu dữ liệu, kiểu dữ liệu là cơ bản, và nó khá giống nhau với tất cả các ngôn ngữ. Tất cả các kiểu dữ liệu được cấu tạo từ các **bit**

1. byte
2. boolean
3. char
4. short
5. int
6. long
7. float
8. double
9. Kiểu tham chiếu (Reference Type)

# Kiểu dữ liệu trong java

- **Java** có 2 loại kiểu dữ liệu: Các kiểu dữ liệu nguyên thủy (Primitive Data Types)
- Các kiểu dữ liệu tham chiếu (Reference Types)
- Có 8 kiểu dữ liệu nguyên thủy (Primitive Data type) đó là: **boolean, byte, char, short, int, long, float, double**.

## Kiểu dữ liệu trong java

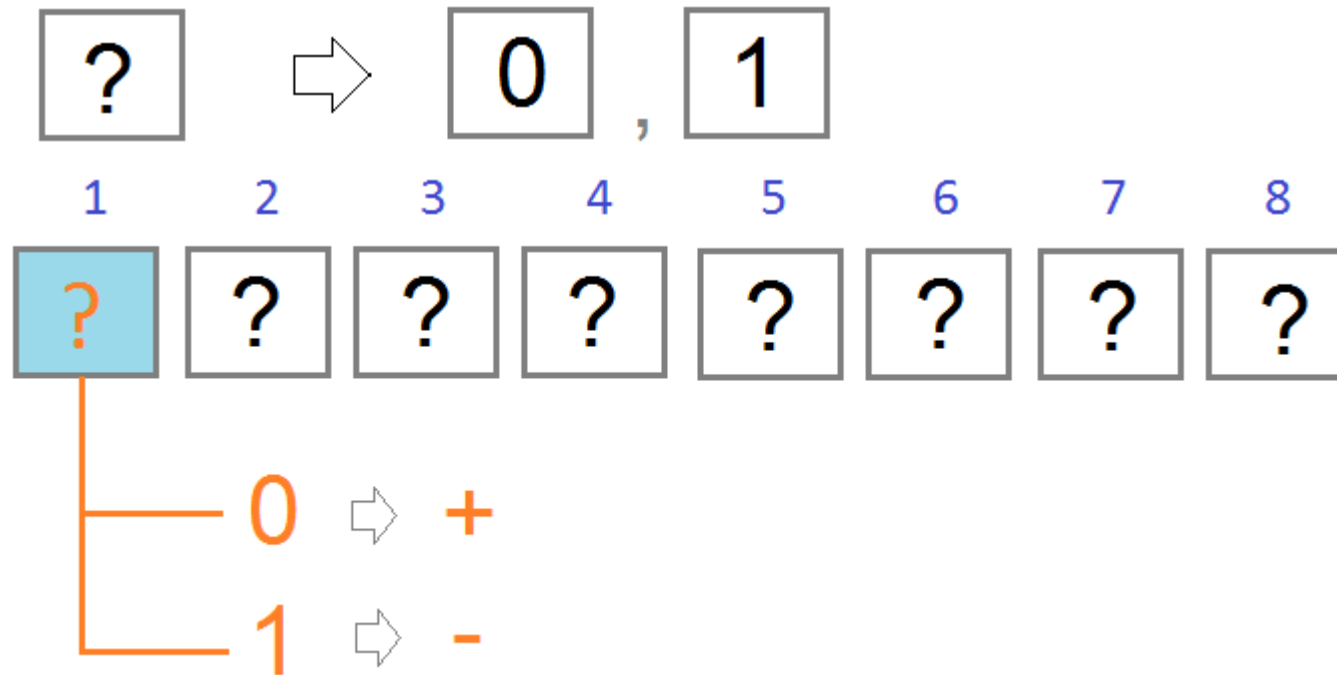
<b>Kiểu dữ liệu</b>	<b>Giá trị mặc định</b>	<b>Kích thước</b>
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

# Kiểu dữ liệu trong java

- Kiểu logic: **boolean**.
- Các kiểu số nguyên (integer): **byte, short, char, int, long**.
- Các kiểu số thực (real number) (Cũng được gọi là kiểu dấu chấm động (floating point)): **float, double**.
- Các kiểu tham chiếu, là các đối tượng được tạo ra bởi Constructor của các lớp.

## Kiểu dữ liệu trong java

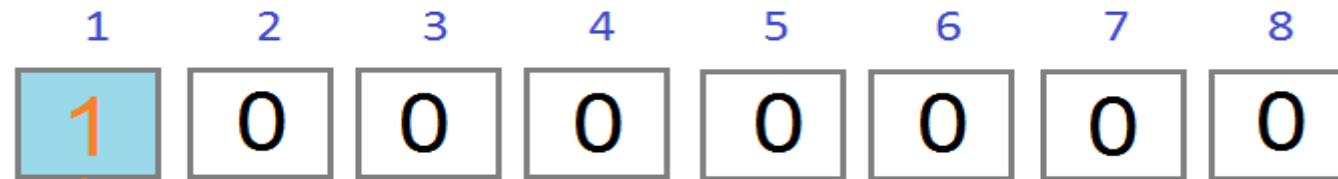
1 **byte** là 8 **bit** nằm kề nhau trên vùng bộ nhớ của máy tính. Mỗi **bit** là một số nhị phân 0 hoặc 1. **Java** lấy tên "**byte**" để đặt cho một kiểu số nguyên (integer) có phạm vi nhỏ (Kích thước 1 **byte**).



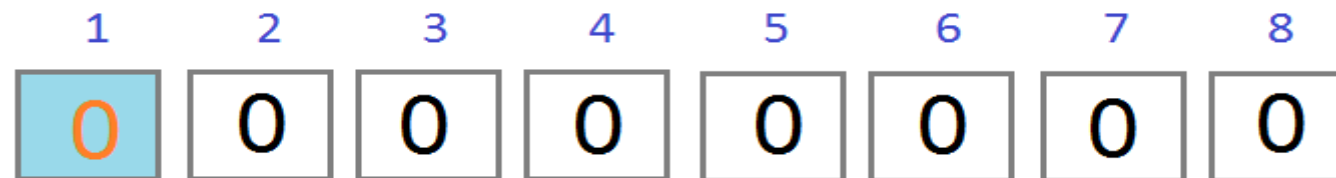
## Kiểu dữ liệu trong java

- **bit** đầu tiên trong dãy 8 **bit** có giá trị là 0 hoặc 1. Nếu là 0 **Java** coi nó là dấu + (Đại diện cho số dương)
- Nếu là 1 **Java** coi nó là dấu - (Đại diện cho số âm)
- Với 7 **bit** phía sau bạn có thể biểu diễn được một số từ 0 tới 127. Suy ra kiểu **byte** trong **Java** có phạm vi  $[-127, 127]$ .  
*Nhưng khoan đã nó phải là  $[-128, 127]$ , tại sao vậy?*
- số nhỏ nhất của kiểu byte trong Java là -128
- Nếu theo quy tắc **bit** đầu tiên có giá trị 0 tương đương dấu +, giá trị 1 tương đương dấu -, thì chúng ta có tới 2 cách để biểu diễn số 0

## Kiểu dữ liệu trong java



⇒ -00000000



⇒ +00000000

Vì vậy dãy nhị phân "1 0 0 0 0 0 0" nên được coi là số nhỏ nhất của kiểu dữ liệu byte trong Java, nó biểu diễn số -128.

## Kiểu dữ liệu trong java

- **3- boolean**
- **boolean** là kiểu dữ liệu đơn giản nhất, nó có kích thước 1 **bit**. Nó có 2 giá trị là **true** và **false**.
- Giá trị mặc định cho kiểu **boolean** là **false**.



# Kiểu dữ liệu trong java

## 4- char

Kiểu "char" là 4 ký tự đầu tiên của từ "Character" (Ký tự), nhưng kiểu char trong Java sử dụng để lưu trữ các số nguyên không âm (Non-negative integers) có kích thước 2 byte. Nó cũng được sử dụng để đại diện cho một ký tự Unicode, vì bản chất mỗi ký đều tương ứng với một con số cụ thể (Con số này được hiểu như là mã của ký tự).

Vì char là kiểu số nguyên không âm, kích thước 2 byte, nên phạm vi của nó là  $[0, 2*16-1]$  (  $[0, 65535]$  ).

Khi char được hiểu như một ký tự Unicode, ký tự nhỏ nhất là '\u0000' (Mã 0), và ký tự lớn nhất là '\uffff' (Mã 65535).

# Kiểu dữ liệu trong java

## 5- short

short là kiểu dữ liệu để biểu diễn một số nguyên 2 byte (16 bit), bao gồm cả số nguyên âm.

- Giá trị nhỏ nhất là -32,768 ( $-2^{15}$ ) và giá trị lớn nhất là 32,767 ( $2^{15} - 1$ ).
- Giá trị mặc định là 0.

# Kiểu dữ liệu trong java

## 6- int

Kiểu dữ liệu int được sử dụng để biểu diễn một số nguyên có kích thước 4 byte (32 bit).

Giá trị nhỏ nhất:  $-2,147,483,648$  ( $-2^{31}$ )

Giá trị lớn nhất:  $2,147,483,647$  ( $2^{31} - 1$ )

Giá trị mặc định 0.

Trong Java, kiểu dữ liệu int được coi là kiểu dữ liệu mặc định cho các số nguyên. Như vậy nếu bạn viết số 100, Java sẽ tạo ra một vùng bộ nhớ 4 byte để lưu trữ. Còn nếu bạn muốn Java tạo ra một vùng bộ nhớ 8 byte để lưu trữ số 100 bạn phải ghi là 100L. (long là kiểu số nguyên 8 byte, được giới thiệu ở mục bên dưới).

# Kiểu dữ liệu trong java

## 7- long

Kiểu dữ liệu long sử dụng để biểu diễn các số nguyên có kích thước lên đến 8 byte (64 bit).

Giá trị nhỏ nhất là  $-9,223,372,036,854,775,808. (-2^{63})$

Giá trị lớn nhất là  $9,223,372,036,854,775,807. (2^{63} - 1)$

Kiểu này được sử dụng khi cần một dải giá trị rộng hơn int.

Giá trị mặc định là 0L.

Chú ý: cần thêm ký tự 'L' ngay sau số 100 để Java tạo ra một vùng bộ nhớ 8 byte để lưu trữ số 100

```
long a = 100L;
```

# Kiểu dữ liệu trong java

## 8- float

Kiểu dữ liệu float được sử dụng để biểu diễn một số thực (real number) có kích thước 4 byte (32 bit).

Giá trị nhỏ nhất:  $-3.4028235 \times 10^{38}$

Giá trị lớn nhất:  $3.4028235 \times 10^{38}$

Giá trị mặc định 0.0f

Khai báo float a=2.5

# Kiểu dữ liệu trong java

## 9- double

Kiểu dữ liệu double được sử dụng để biểu diễn một số thực (real number) có kích thước 8 byte (64 bit). Nó là kiểu mặc định cho các kiểu số thực.

Giá trị nhỏ nhất:  $-1.7976931348623157 \times 10^{308}$

Giá trị lớn nhất:  $1.7976931348623157 \times 10^{308}$

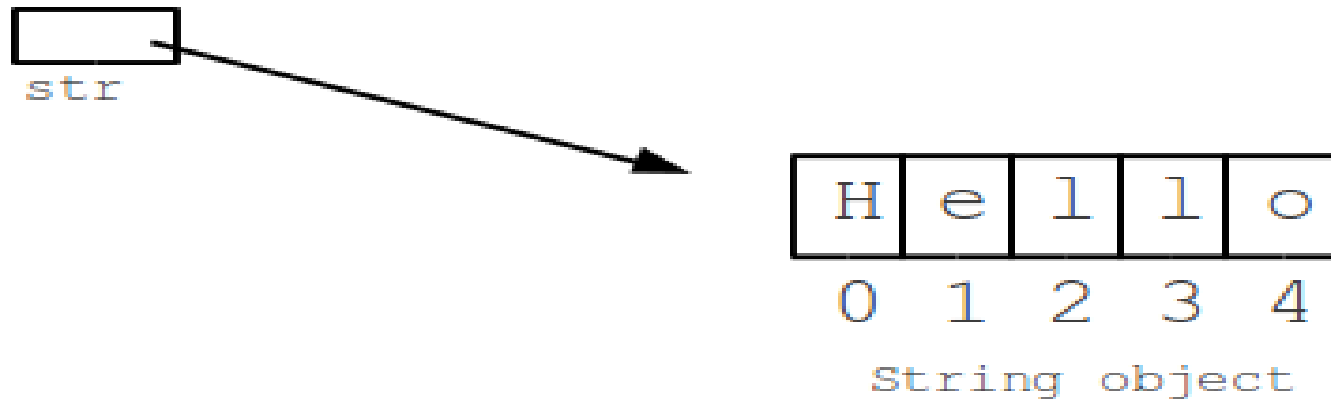
Giá trị mặc định 0.0d

Khai báo cần thêm chữ d

# Kiểu dữ liệu trong java

## 10- Kiểu tham chiếu (Reference Type)

Trong Java một kiểu dữ liệu được tạo ra bởi sự kết hợp các kiểu nguyên thủy với nhau được gọi là kiểu tham chiếu (Reference type). Kiểu tham chiếu thường được sử dụng nhất đó là String, nó là sự kết hợp của các ký tự.



Các kiểu dữ liệu tham chiếu được tạo ra dựa trên một lớp. Lớp (class) giống như một bản thiết kế (blueprint) để định nghĩa một kiểu tham chiếu.

# Kiểu dữ liệu trong java

```
class Address {  
    String address;  
    String cityName;  
}
```

```
class Student {  
  
    String fullName;  
  
    int age;  
  
    Address address;  
  
}
```



# Kiểu dữ liệu trong java

## Dữ liệu kiểu ngày trong java

- Các lớp Date, time, calendar trong Java
- Java cung cấp một số class liên quan tới thời gian và lịch (Calendar), dưới đây là danh sách các class này:

# Kiểu dữ liệu trong java

Class	Mô tả
java.util. <b>Date</b>	Một lớp đại diện cho ngày tháng năm và thời gian. Tiếc là hầu hết các phương thức của nó đã bị lỗi thời, khuyến cáo là không nên sử dụng các phương thức đó, tuy nhiên lớp <b>Date</b> vẫn được sử dụng rộng rãi.
java.util.concurrent. <b>TimeUnit</b>	<b>TimeUnit</b> là một Enum mô tả các đơn vị ngày tháng năm và thời gian.
java.sql. <b>Date</b>	Một lớp mô tả ngày tháng năm. Thông tin về thời gian bị cắt bỏ. Lớp này thường sử dụng trong <b>JDBC</b> .
java.sql. <b>Time</b>	Một lớp mô tả thời gian (Giờ phút giây, milli giây), và không chứa thông tin ngày tháng năm. Lớp này thường sử dụng trong <b>JDBC</b> .
java.sql. <b>Timestamp</b>	Một lớp mô tả ngày tháng năm và thời gian. Lớp này thường sử dụng trong <b>JDBC</b> .
java.util. <b>Calendar</b>	Lớp mô tả bộ Lịch. Nó có các phương thức toán học về thời gian, chẳng hạn thêm ngày, bớt ngày,...
java.util. <b>GregorianCalendar</b>	Là một lớp con trực tiếp của <b>java.util.Calendar</b> , mô tả ngày Dương Lịch, bộ lịch được sử dụng rộng rãi trên thế giới ngày nay. Nó có tất cả các method từ <b>java.util.Calendar</b> để thao tác toán học trên ngày tháng năm và thời gian.
java.util. <b>TimeZone</b>	<b>TimeZone</b> là lớp mô tả múi giờ, nó có ích khi bạn làm việc với Lịch trên múi giờ.
java.text. <b>SimpleDateFormat</b>	Lớp này giúp bạn chuyển một <b>String</b> có định dạng ngày tháng sang kiểu <b>Date</b> và ngược lại

# Kiểu dữ liệu trong java

# Kiểu dữ liệu trong java

Class	Mô tả
java.util. <b>Date</b>	Một lớp đại diện cho ngày tháng năm và thời gian. Tiếc là hầu hết các phương thức của nó đã bị lỗi thời, khuyến cáo là không nên sử dụng các phương thức đó, tuy nhiên lớp <b>Date</b> vẫn được sử dụng rộng rãi.
java.util.concurrent. <b>TimeUnit</b>	<b>TimeUnit</b> là một Enum mô tả các đơn vị ngày tháng năm và thời gian.
java.sql. <b>Date</b>	Một lớp mô tả ngày tháng năm. Thông tin về thời gian bị cắt bỏ. Lớp này thường sử dụng trong <b>JDBC</b> .
java.sql. <b>Time</b>	Một lớp mô tả thời gian (Giờ phút giây, milli giây), và không chứa thông tin ngày tháng năm. Lớp này thường sử dụng trong <b>JDBC</b> .
java.sql. <b>Timestamp</b>	Một lớp mô tả ngày tháng năm và thời gian. Lớp này thường sử dụng trong <b>JDBC</b> .
java.util. <b>Calendar</b>	Lớp mô tả bộ Lịch. Nó có các phương thức toán học về thời gian, chẳng hạn thêm ngày, bớt ngày,...
java.util. <b>GregorianCalendar</b>	Là một lớp con trực tiếp của <b>java.util.Calendar</b> , mô tả ngày Dương Lịch, bộ lịch được sử dụng rộng rãi trên thế giới ngày nay. Nó có tất cả các method từ <b>java.util.Calendar</b> để thao tác toán học trên ngày tháng năm và thời gian.
java.util. <b>TimeZone</b>	<b>TimeZone</b> là lớp mô tả múi giờ, nó có ích khi bạn làm việc với Lịch trên múi giờ.
java.text. <b>SimpleDateFormat</b>	Lớp này giúp bạn chuyển một <b>String</b> có định dạng ngày tháng sang kiểu <b>Date</b> và ngược lại