

Java က object-oriented ဆိုပေမယ့် type တွေအားလုံးကတော့ object တွေမဟုတ်ပါဘူး Primitives လို့ခေါ်တဲ့အခြေခံ variable types တွေရှိပါတယ်။အဲ့ဒါတွေကိုအောက်မှာ list နဲ့ဖော်ပြထားပါတယ်။

- `byte` (number, 1 byte)
- `short` (number, 2 bytes)
- `int` (number, 4 bytes)
- `long` (number, 8 bytes)
- `float` (float number, 4 bytes)
- `double` (float number, 8 bytes)
- `char` (a character, 2 bytes)
- `boolean` (true or false, 1 byte)

Java မှာ variable တွေကိုအသုံးမပြုခင်အရင်ဆုံးကြေငြာရမှာဖြစ်ပါတယ်။

### Numbers

နံပါတ်တစ်ခုကိုကြေငြာ(declare)ပြီးတန်ဖိုးထည့်(assign)ဖို့အောက်ပါပုံစံ(syntax)အတိုင်းရေးရပါတယ်။

```
int myNumber;  
myNumber = 5;
```

ဒါမှမဟုတ်နှစ်ကြောင်းကိုတစ်ကြောင်းပေါင်းရေးလည်းရပါတယ်။

```
int myNumber = 5;
```

ဒဿမကိန်းအတွက်ဆိုရင် double နဲ့ float ဆိုပြီးနှစ်မျိုးရှိပါတယ်။

double အတွက်အောက်ပါအတိုင်းရေးရပြီး

```
double d = 4.5;  
d = 3.0;
```

float အတွက်ဆိုရင်တော့အောက်ပါအတိုင်းရေးရပါတယ်။

```
float f = (float) 4.5;
```

ဒါမှမဟုတ် float အတွက် ဒီလိုသုံးလည်းရပါတယ်။

```
float f = 4.5f;
```

2.1 လိုပုံမှန်ဒဿမကိန်းတစ်ခုဆိုရင် java မှာ double အနေနဲ့ယူပါတယ်။

float အနေနဲ့သုံးချင်တာဆိုရင်တော့ double ကနေ float ဖြစ်အောင် type casting လုပ်ရပါတယ်။

type casting လုပ်တဲ့နည်းနှစ်မျိုးရှိပါတယ်။ (နှစ်နည်းလုံးအပေါ်မှာဖော်ပြပြီးပါပြီ)

ပထမနည်းက

```
float f = (float) 4.5;
```

ဒုတိယနည်းက

```
float f = 4.5f;
```

ပဲဖြစ်ပါတယ်။

Characters and Strings

Character အတွက် ရေးပုံရေးနည်း (syntax) ကအောက်ပါအတိုင်းဖြစ်ပါတယ်။

```
char c = 'g';
```

String ကတော့ Java မှာ primitive data type မဟုတ်ပါဘူး။ String

ကိုအောက်ပါနည်းလမ်းတွေအတိုင်းအသုံးပြုလို့ရပါတယ်။

```
// constructor သုံးပြီး String တစ်ခုဖန်တီးပုံပါ
```

```
String s1 = new String("Who let the dogs out?"); // String object တွေကို heap memory ထဲမှာသိမ်းပါတယ်
```

```
// constructor သုံးစရာမလို (အပေါ်ကနည်းအတိုင်းမဟုတ်ဘဲ) "" ကိုသာသုံးပြီး String တစ်ခုဖန်တီးလို့လည်းရပါတယ်
```

```
String s2 = "Who who who who!"; // ဒီလိုပုံစံနဲ့ဖန်တီးတဲ့ String တွေကိုတော့ String pool ထဲမှာသိမ်းပါတယ်
```

```
// Java မှာ String နှစ်ခုပေါင်းဖို့အတွက် + operator ကိုသုံးပါတယ်
```

```
String s3 = s1 + s2;
```

String နဲ့ primitive တွေကိုလည်းအောက်ကအတိုင်းပေါင်းလိုရပါတယ်

```
int num = 5;  
String s = "I have " + num + " cookies"; // primitive နဲ့ပေါင်းဖို့ဆိုရင်တော့ constructor  
ကိုအသုံးမပြုဘဲ "" ကိုအသုံးပြုပြီး String ဖန်တီးဖို့လိုပါတယ်
```

boolean

Java မှာနှိုင်းယှဉ်တဲ့ comparison operator တိုင်းဟာ boolean ကို return ပြန်ပေးပါတယ်။ boolean မှာ true နဲ့ false ဆိုပြီးတန်ဖိုးနှစ်ခုပဲရှိပါတယ်။

```
boolean b = false;  
b = true;  
  
boolean toBe = false;  
b = toBe || !toBe;  
if (b) {  
    System.out.println(toBe);  
}  
  
int children = 0;  
b = children; // boolean type ထဲကို int type ထည့်မှာမို့အလုပ်လုပ်မှာမဟုတ်ပါ  
  
if (children) { // if သုံးပြီးစစ်လို့ရတာက boolean တွေကိုသာစစ်လို့ရတာပါ children က int type  
မို့အလုပ်လုပ်မှာမဟုတ်ပါဘူး  
    // Will not work  
}  
  
int a;  
boolean b = true;  
boolean c = false;  
a = b + c; //boolean တန်ဖိုးနှစ်ခုကို + သုံးပြီးပေါင်းလို့မရတဲ့အပြင် boolean တန်ဖိုးကို int  
ထဲထည့်လို့မရပါ။ ဒါကြောင့်အလုပ်လုပ်မှာမဟုတ်ပါ။  
System.out.println(a);
```