**Data Types in Java**

**Java Data Types**

There are majorly two types of languages. The first one is **Statically typed language**where each variable and expression type is already known at compile time. Once a variable is declared to be of a certain data type, it cannot hold values of other data types. Example: C,C++, Java. The Other one is **Dynamically typed languages:**These languages can receive different data types over time. Example: Ruby, Python.  
  
Java is a **statically typed and also a strongly typed language** because in Java each type of data (such as integer, character, hexadecimal, packed decimal, and so forth) is predefined as part of the programming language and all constants or variables defined for a given program must be described with one of the data types.  
  
Java has two categories of data:

* Primitive data (e.g., number, character)
* Object data (programmer created types)

**Primitive data**

Primitive data are only single values; they have no special capabilities. There are 8 primitive data types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Description** | **Default** | **Size** | **Examples** |
| **boolean** | true or false | false | 1 bit | true, false |
| **byte** | two's complement integer | 0 | 8 bits | (none) |
| **char** | unicode character | \u0000 | 16 bits | 'a','\n','\u0041' |
| **short** | two's complement integer | 0 | 16 bits | (none) |
| **int** | two's complement integer | 0 | 32 bits | -1,0,1 |
| **long** | two's complement integer | 0 | 64 bits | -1L,0L,1L |
| **float** | IEEE 754 floating point | 0.0 | 32 bits | -1.23e100f,2.45e100f |
| **double** | IEEE 754 floating point | 0.0 | 64 bits | -3.45e300d,4.56e245d |

Let us look at each of these types in detail:

* **boolean:**boolean data type represents only one bit of information, **either true or false** . Values of type boolean are not converted implicitly or explicitly (with casts) to any other type; however, the programmer can easily write the conversion code.

Java

// A Java program to demonstrate boolean data type

**class** **GeeksforGeeks**

{

public static void main(String args[])

{

boolean b = true;

**if** (b == true)

System.out.println("Hi Geek");

}

}

**Output**:

Hi Geek

* **byte:**The byte data type is an 8-bit signed two's complement integer. The byte data type is useful for saving memory in large arrays.
  + Size: 8-bit
  + Value: -128 to 127

Java

// Java program to demonstrate byte data type **in** Java

**class** **GeeksforGeeks**

{

public static void main(String args[])

{

byte a = 126;

// byte **is** 8 bit value

System.out.println(a);

a++;

System.out.println(a);

// It overflows here because

// byte can hold values **from** -128 to 127

a++;

System.out.println(a);

// Looping back within the range

a++;

System.out.println(a);

}

}

**Output**:

126

127

-128

-127

* **short:**The short data type is a 16-bit signed two's complement integer. Similar to byte, use a short to save memory in large arrays, in situations where the memory savings actually matters.
  + **Size:** 16 bit
  + **Value:** -32,768 to 32,767 (inclusive)
* **int**: It is a 32-bit signed two's complement integer.
  + **Size:**32 bit
  + **Value:** -231 to 231-1

**Note**: In Java SE 8 and later, we can use the int data type to represent an unsigned 32-bit integer, which has value in the range [0, 232-1]. Use the Integer class to use int data type as an unsigned integer.

* **long:** The long data type is a 64-bit two's complement integer.
  + Size: 64 bit
  + Value: -263 to 263-1.

**Note:** In Java SE 8 and later, you can use the long data type to represent an unsigned 64-bit long, which has a minimum value of 0 and a maximum value of 264-1. The Long class also contains methods like compareUnsigned, divideUnsigned etc., to support arithmetic operations for unsigned long.

* **Floating point Numbers : float and double float:**The float data type is a single-precision 32-bit [IEEE 754](https://en.wikipedia.org/wiki/IEEE_floating_point) floating point. Use a float (instead of double) if you need to save memory in large arrays of floating point numbers.
  + **Size:**32 bits
  + **Suffix :**F/f Example: 9.8f

**double:**The double data type is a double-precision 64-bit IEEE 754 floating point. For decimal values, this data type is generally the default choice. **Note:** Both float and double data types were designed especially for scientific calculations, where approximation errors are acceptable. If accuracy is the most prior concern then, it is recommended not to use these data types and use [BigDecimal](http://docs.oracle.com/javase/1.5.0/docs/api/java/math/BigDecimal.html" \t "_blank) class instead.

* **char**: The char data type is a single 16-bit Unicode character. A char is a single character.
  + Value: '\u0000' (or 0) to '\uffff' 65535

Java

// Java program to demonstrate primitive data types **in** Java

**class** **GeeksforGeeks**

{

public static void main(String args[])

{

// declaring character

char a = 'G';

// Integer data type **is** generally

// used **for** numeric values

int i=89;

// use byte **and** short **if** memory **is** a constraint

byte b = 4;

// this will give error **as** number **is**

// larger than byte range

// byte b1 = 7888888955;

short s = 56;

// this will give error **as** number **is**

// larger than short range

// short s1 = 87878787878;

// by default fraction value **is** double **in** java

double d = 4.355453532;

// **for** float use 'f' **as** suffix

float f = 4.7333434f;

System.out.println("char: " + a);

System.out.println("integer: " + i);

System.out.println("byte: " + b);

System.out.println("short: " + s);

System.out.println("float: " + f);

System.out.println("double: " + d);

}

}

**Output**:

char: G

integer: 89

byte: 4

short: 56

float: 4.7333436

double: 4.355453532