

Processing: Data Types

There are eight primitive data types supported by Java. Primitive data types are predefined by the language and named by a keyword. Let us now look into detail about the eight primitive data types.

byte:

- Minimum value is -128 (-2^7), Maximum value is 127 (inclusive) ($2^7 - 1$). Default value is 0
- Example: `byte a = 100` , `byte b = -50`

short:

- Minimum value is $-32,768$ (-2^{15}), Maximum value is $32,767$ (inclusive) ($2^{15} - 1$). Default value is 0 .
- Example: `short s = 10000`, `short r = -20000`

int:

- Minimum value is $-2,147,483,648$ (-2^{31}), Maximum value is $2,147,483,647$ (inclusive). ($2^{31} - 1$)
- Int is generally used as the default data type for integral values unless there is a concern about memory.
- The default value is 0 .
- Example: `int a = 100000`, `int b = -200000`

long:

- Minimum value is $-9,223,372,036,854,775,808$ (-2^{63}), Maximum value is $9,223,372,036,854,775,807$ (inclusive). ($2^{63} - 1$)
- This type is used when a wider range than int is needed. The default value is `0L`.
- Example: `long a = 100000L`, `int b = -200000L`

float:

- Float is mainly used to save memory in large arrays of floating point numbers.
- Default value is `0.0f`.
- Example: `float f1 = 234.5f`

double:

- This data type is generally used as the default data type for decimal values, generally the default choice.
- Default value is `0.0d`.
- Example: `double d1 = 123.4`

boolean:

- There are only two possible values: true and false.
- This data type is used for simple flags that track true/false conditions. Default value is false.
- Example: boolean one = true

char:

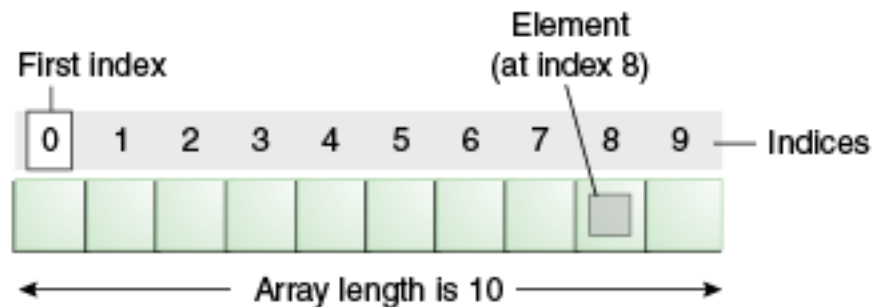
- Char data type is used to store any character, such as letters and numbers.
- Example: char letterA ='A';

String:

- Strings are a special data type. They are an *array* of **char**
- Example: String someWords = "This is a String, enclosed in quotes.";

Arrays

An *array* is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.



An array of 10 elements.

Each item in an array is called an *element*, and each element is accessed by its numerical *index*. As shown in the preceding illustration, numbering begins with 0. The 9th element, for example, would therefore be accessed at index 8.

Example: `int[] arrayOfIntegers = new int[256];` // creates an Array of integers with 256 elements