**Java Primitive Data Types:**

* These data types are predefined.
* All primitive types are keywords.
* Primitive data types are mainly divided into three types they are.
* Numeric.
* Single Character.
* Boolean.
* Again, Numeric data is classified into two types
* Integer.
* Decimal.
* Integer data is represented by using.
* byte.
* Byte data type is an 8-bit signed integer.
* Minimum value is -128.
* Maximum value is 127.
* Default value is 0
* short.
* Short data type is a 16-bit signed integer.
* Minimum value is -32,768.
* Maximum value is 32,767.
* Int.
* Int data type is a 32-bit signed integer.
* Minimum value is - 2,147,483,648.
* Maximum value is 2,147,483,647.
* Long
* Long data type ia s 64-bit signed integer.
* Minimum value is -9,223,372,036,854,775,808.
* Maximum value is 9,223,372,036,854,775,807.
* Decimal data is represented by using
* float.
* Float data type is a single-precision 32-bit floating point.
* Float is mainly used to save memory in large arrays of floating point numbers.
* Default value is 0.0f.
* double.
* double data type is a double-precision 64-bit floating point.
* This data type is generally used as the default data type for decimal values, generally the default choice.
* Default value is 0.0d.
* Single Character data is represented by using *char* keyword.
* Char is a single 16-bit Unicode character.
* Minimum value is '\u0000' or 0.
* Maximum value is '\uffff' or 65,535.
* Boolean data is represented by using *boolean* keyword.
* These types serve as the building blocks of data manipulation in Java.

**Primitive Type Casting:**

* Converting one primitive data type to another primitive data type is called primitive type casting.
* Primitive data types are divided in two types
* Narrowing.
* Widening.

**Narrowing:**

* Process of converting bigger primitive data type to smaller primitive data type in case of narrowing there is always a loss of data, hence it is unsafe and explicit.
* Example: int a=(int)5.9

System.out.println(a);

//Output will be 5.

**Widening:**

* Process of converting smaller primitive datatype to bigger primitive data type.
* In case of widening there is no loss of data, hence it is safe and implicit(We can also achieved through explicit).
* Example: double i=6;

System.out.println(i);

//Output will be 6.0