TOKENS:

✓ A Token is the smallest Element of a Program that is meaningful to the Compiler.

✓ The Compiler Identifies these Token easily because their meanings are predefined in the Compiler S/W.

Eg: Here Compiler automatically Identifies that the int is the keyword and number is the value and a is the variable here.

```
int a = 10;
```

There are basically 5 types of Tokens are available in Java programming-

1.Keywords
2.Identifiers
3.Literals
4.Operators
5.Separators

Keywords: There are 50 keywords currently defined in the Java Language.
And we have 3 Reserved Literals.

Primitive Types (8)	Control Statements (11)	Error Handling (6)	Access Modifiers (3)	Non Access Modifiers (6)	OOPS (7)
int	if	try	private	static	class
byte	else	catch	protected	final	Implements
short	while	throw	public	volatile	extends
long	for	throws		transient	interface
float	do	finally		abstract	new
double	break	assert (1.4)		synchronized	super
char	continue				this
boolean	return				
	switch				
	default				
	case				

Other Keywords (6)	Return Type (1)	Un used (2)	Reserved Literals
strictfp (1.2)	void	goto	true
import		const	False
package			null
native			
instanceof			
enum			

Note:

✓ These Keywords cannot be used as Identifiers.
✓ They cannot be used as Names for a Varibale, Class, Method.
✓ const and goto Keywords are reserved but not used.

We have 8 types of Primitive Data Types:

Data Type	Default Value	Data type in bytes //public static final int BYTES	Data type in size //public static final int SIZE
byte	0	1 byte	8 bit
short	0	2 bytes	16 bit
int	0	4 bytes	32 bit
long	0L	8 bytes	64 bit
float	0.0f	4 bytes	32 bit
double	0.0d	8 bytes	64 bit
boolean	False	1 bit	can't be defined precisely.
char	‘\u0000’	2 bytes	16

Note: Java uses Unicode System.
The \is the Lowest range of Unicode system.
The \uFu0000 FFF is the Highest range of Unicode system.

Data Type	Range Values
int	-2147483648 to 2147483647
byte	-128 to 127
short	-32768 to 32767
long	-9223372036854775808 to 9223372036854775807
float	1.4E-45 to 3.4028235E38
double	4.9E-324 to 1.7976931348623157E308
boolean	true and false
char	0 to 65535

```
float f1 = 5E8f;      //after 5, allowing 8 zeros
float f2 = 500000000f;
System.out.println(f1);
System.out.println(f2);
```

```
5.0E8
5.0E8
```

```java
byte b1 = 101;
System.out.println(b1); //101

short s1 = 200;
System.out.println(s1); // 200

int i1 = 1000;
System.out.println(i1); // 1000

long l1 = 987654321;
System.out.println(l1); // 987654321

float f1 = 10.0f;
System.out.println(f1); // 10.0

double d1 = 20.0d;
System.out.println(d1); // 20.0

char ch = 'A';
System.out.println(ch); // A

boolean b2 = true;
System.out.println(b2); // true
```

```
//modifier and type
static byte b;  // static variables
static short s;
static int i;
static long l;
static float f;
static double d;
static char ch;
static boolean b1;

public static void main(String[] args) {

System.out.println(b); // 0
System.out.println(s); //0
System.out.println(i); // 0
System.out.println(l); // 0
System.out.println(f); // 0.0
System.out.println(d); // 0.0
System.out.println(ch); //
System.out.println(b1); // false
}
```

Wrapper class

Primitive Type	Wrapper Type
boolean	Boolean public final class **Boolean** extends Number implements Comparable<Boolean>
char	Character public final class **Character** extends Object implements Serializable, Comparable<Character>
byte	Byte public final class **Byte** extends Number implements Comparable<Byte>
Short	Short public final class **Short** extends Number implements Comparable<Short>
Int	Integer public final class **Integer** extends Number implements Comparable<Integer>
Long	Long public final class **Long** extends Number implements Comparable<Long>
Float	Float public final class **Float** extends Number implements Comparable<Float>
double	Double public final class **Double** extends Number implements Comparable<Double>

https://docs.oracle.com/javase/9/docs/api/java/lang/package-summary.html#:~:text=The%20wrapper%20classes%20Boolean%20%2C%20Character,a%20variable%20of%20reference%20type.

```java
//Wrapper Types
public class Eg3 {

Byte b;
Short s;
Integer i;
Float f;
Double d;
Long l;
Boolean b1;
Character ch;

public static void main(String[] args) {

}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Byte> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Byte.BYTES); // 1
System.out.println(Byte.SIZE); // 8
System.out.println(Byte.MAX_VALUE + " " + Byte.MIN_VALUE); // 127 -128
System.out.println(Byte.TYPE); // byte
}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Short> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Short.BYTES); // 2
System.out.println(Short.SIZE); // 16
System.out.println(Short.MAX_VALUE + " " + Short.MIN_VALUE); // 32767 -32768
System.out.println(Short.TYPE); // short
}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Short> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Integer.BYTES); // 4
System.out.println(Integer.SIZE); // 32
System.out.println(Integer.MAX_VALUE + " " + Integer.MIN_VALUE); // 32767 -32768
System.out.println(Integer.TYPE); // int

}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Short> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Long.BYTES); // 8
System.out.println(Long.SIZE); // 64
System.out.println(Long.MAX_VALUE + " " + Integer.MIN_VALUE); // 9223372036854775807 -2147483648
System.out.println(Long.TYPE); // long

}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Float> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Float.BYTES); // 4
System.out.println(Float.SIZE); // 32
System.out.println(Float.MAX_VALUE + " " + Float.MIN_VALUE); //3.4028235E38  1.4E-45
System.out.println(Float.TYPE); // float
}
```

```
public static void main(String[] args) {

// Modifier and Type //Field
// static int Bytes
// static byte MAX_VALUE
// static byte MINVALUE
// static int SIZE
// static Class<Double> Type

// ClassName.Field -- BYTES, SIZE, MAX_VALUE and MIN_VALUE, TYPE
System.out.println(Double.BYTES); // 8
System.out.println(Double.SIZE); // 64
System.out.println(Double.MAX_VALUE + " " + Double.MIN_VALUE); //1.7976931348623157E308   4.9E-324
System.out.println(Double.TYPE); // double
}
```

```
public static void main(String[] args) {

// ClassName.Field -- TRUE, FALSE, TYPE
System.out.println(Boolean.TRUE); // true
System.out.println(Boolean.FALSE); // false
System.out.println(Boolean.TYPE); // boolean
}
```

```java
public static void main(String[] args) {

System.out.println(Character.BYTES); // 2
System.out.println(Character.MAX_VALUE +0); // 65535
System.out.println(Character.MIN_VALUE+0); // 0
System.out.println(Character.TYPE); // char
}
```