Characters (char)

→ 'a', 'b', 'C', 'M', '+', '-', '%', ' '

→ surrounded by single quotes

Strings (String)

→ collection of characters

→ " " double quotes

String str = "Jatin kumar";

In built functions

String str = "Rahul Sharma";
0 1 2 3 4 5 6 7 8 9 10 11

Indexing always start from zero 0.

int len = str.length(); → 12
(returns int)

str.charAt(5); → 'h'
(7); → 'a'
(11); → 'a'
(1); → 'a'
(15); → error
(Index Out of bound)

char ch = charAt(index)
(return char)

Inputs

"Izhar Haider"

Character | `char ch = S.next().charAt(0);`

Strings 1) `String str = S.next();` → Izhar

✓ 2) `String str = S.nextLine();` → Izhar Haider

next → one word

nextLine → Line

```
 Scanner s = new Scanner(System.in);
// String str = s.next();
 String str1 = s.nextLine();

// System.out.println(str);
 System.out.println(str1);
```

```
izhar haider
izhar haider

=== Code Execution Succ
```

ASCII values → unique id assigned to each character.

'A' → 65

'B' → 66

'C' → 67

⋮

'Z' → 90

'a' → 97

'b' → 98

'c' → 99

⋮

'z' → 122

```
if ('A' > 'B') {
        ||
}
```

65 > 66

return false.

Type Casting : converting one data type to another.

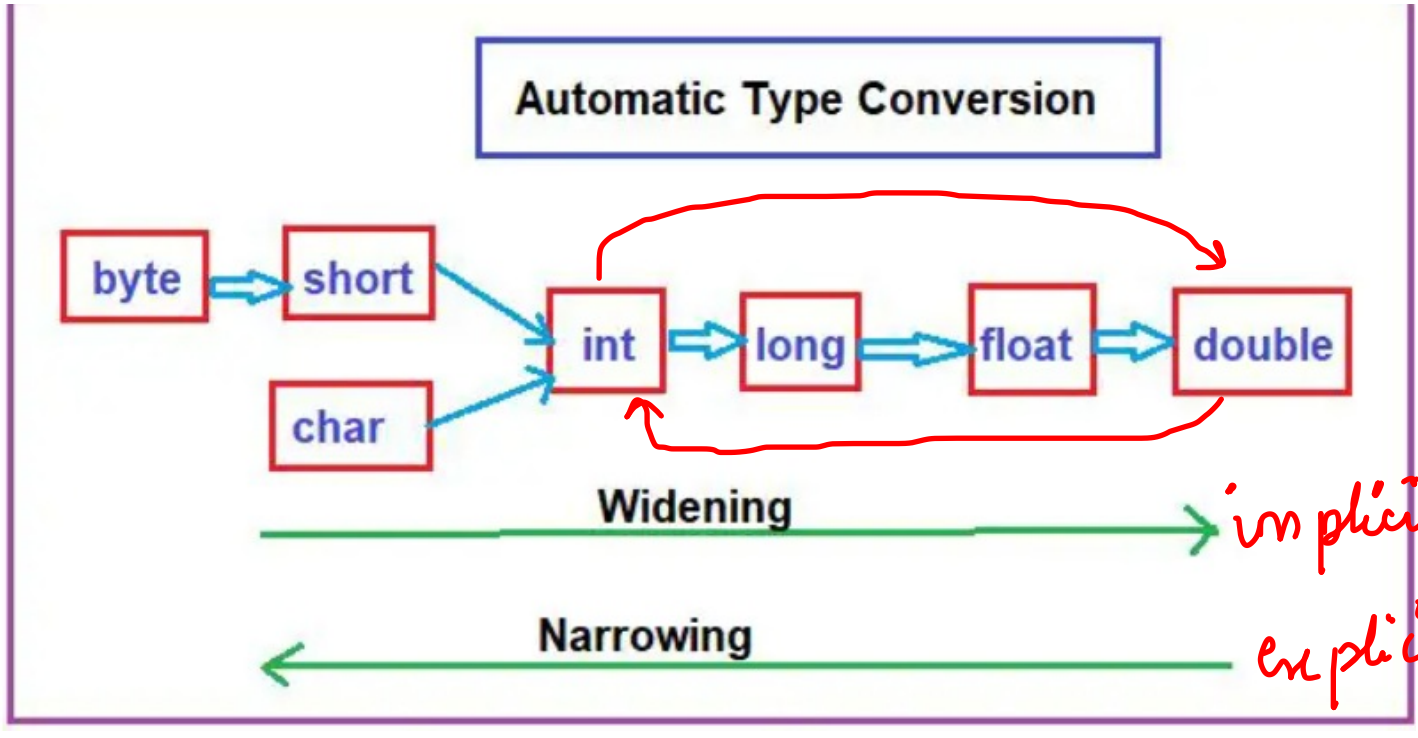
char ch = 'a';

int i = (int x char) →

Implicit → automatic conversion

Explicit → forced conversion (done by us)

```
char ch = 'd';
int i = 2;
int n = (ch + i);
        ('d' + 2);
char c = (char) (ch + i);
```



small to high → implicit (automatic)

high to small → explicit (by suf)

implicit

explicit

```java
int  a = 100;
long b = a;  // 100
float c = b; // 100

System.out.println(a);  // 100
System.out.println(b); // 100
System.out.println(c); // 100.0
```

```
java -cp /tmp/aiIacaNqRi/H
100
100
100.0

=== Code Execution Success
```

Implicit

```
ERROR!
/tmp/IFgZ4iUiDN/HelloWorld.java:18: error: incompatible types: possible
    lossy conversion from double to long
```

→ error printed

```java
double a = 100.0;  // 100.0
long b = (long) a; // 100.0
int c = (int) b; // 100.0

System.out.println(a);  // 100.0
System.out.println(b);  // 100
System.out.println(c);  // 100
```

```
java -cp /tmp/gjsZI
100.0
100
100

=== Code Execution
```

→ Explicit