

→ About Java

→ Printing in Java

```
    system.out.println();           /→ quites  
    system.out.print() + "\n";      %> remains
```

→ Operators in Java

- ① Arithmetic Operator → +, -, /, \*, %, --
- ② Assignment Operator → =, +=, -=, \*=, /=, %=
- ③ Logical Operator → &&, ||, !
- ④ Comparison Operator → >, <, >=, <=, ==, !=
- ⑤ Bitwise Operator

$a = \text{false}$        $b = \text{true}$        $a \& b \rightarrow \text{true}$        $a \& b \rightarrow \text{true}$        $a \& b \rightarrow \text{false}$

$\downarrow$        $\downarrow$        $\downarrow$        $\downarrow$        $\downarrow$

$a = \text{false}$        $b = \text{true}$        $a \& b \rightarrow \text{true}$        $a \& b \rightarrow \text{false}$

$! = \text{false}$

$\downarrow$   
True  
 $\downarrow$   
False

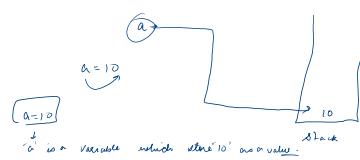
$= =$   
 $\downarrow$   
Equal

(used when we want  
to compare two  
things)

= → assignment  
↳ to assign values

## Variables :-

It is a name given to a memory location, which may have some value.

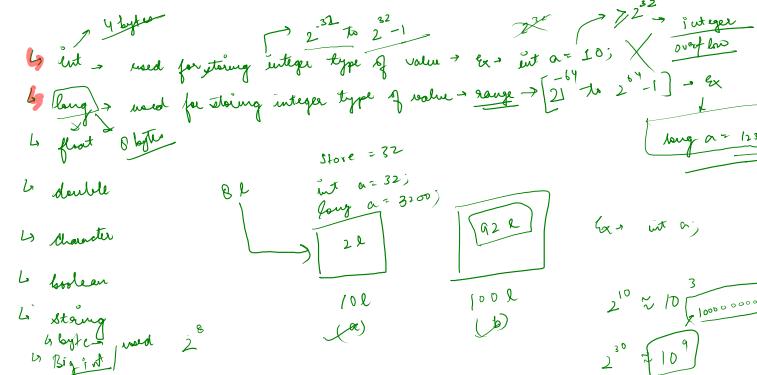


Syntax

```
data-type name-of-variable = value;
```

Ex:- Make a variable which has value `20.05` and name of variable is `k`

```
float k = 20.05;
```



float → used to store decimal-type of value;

↳ ex → float a = 10.002f  
↳ range

double → used to store decimal-type of value;

↳ ex → double a = 10.002507;

## Character :-

↳ It is used to store a single character.

↳ "char"

↳ put in ' ' { single Quotes }

↳ Ex - char ch = 'a'; → ~~char ch = "a";~~  
~~+ char ch = a;~~

## Boolean

↳ stores true | false value.

↳ boolean

↳ ex = boolean d = true; → assigning true value to a variable 'd'

boolean e = false; → assigning false value to a variable 'e'

$$\begin{array}{r} 0 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 0 \ 1 \\ \hline 0 \ 0 \ 0 \ 1 \end{array}$$

a = 5  
b = 9  
c = 1

a & b  
+  
bitwise AND

String :-

- ↳ collection of character
- ↳ String
- ↳ " " ( double quotes )
- ↳ ex → String k = " Abc " ;

int, long, float, double, boolean, char → Primitive data type.

String → Non-Primitive data type

↳

↳ methods.

↳ .length()

```
int a = 10000000000000000000000000000000;  
System.out.println(a);
```

### Compile time error

#### Compile Message

```
Solution.java:38: error: integer number too large: 10000000000000000000000000000000  
    int a = 10000000000000000000000000000000;  
          ^  
1 error
```

```
int a ;  
System.out.println(a);
```

### Compile time error

#### Compile Message

```
Solution.java:39: error: variable a might not have been initialized  
    System.out.println(a);  
                      ^  
1 error
```

with 'L'

```
long a = 1000000000000L;  
System.out.println(a);
```

Your Output

```
1000000000000
```

without 'L'

```
long a = 1000000000000;  
System.out.println(a);
```

Compile time error

Compile Message

```
Solution.java:38: error: integer number too large: 100000000000  
      long a = 100000000000;  
                           ^  
1 error
```

```
    System.out.println(ch);
char ch = a;
System.out.println(ch);
```

### Compile time error

#### Compile Message

```
Solution.java:37: error: cannot find symbol
    char ch = a;
               ^
symbol:  variable a
location: class Solution
1 error
```

```
char ch = "a";
System.out.println(ch);
```

### Compile time error

#### Compile Message

```
Solution.java:37: error: incompatible types: String cannot be converted to char
    char ch = "a";
               ^
1 error
```

```
char ch = 'awqetyfg';
System.out.println(ch);
```

### Compile time error

#### Compile Message

```
Solution.java:37: error: unclosed character literal
    char ch = 'awqetyfg';
                           ^
Solution.java:37: error: unclosed character literal
    char ch = 'awqetyfg';
                           ^
Solution.java:37: error: not a statement
    char ch = 'awqetyfg';
                           ^
3 errors
```

```
boolean a = true, b = false;  
System.out.println(a);  
System.out.println(b);
```

Your Output

```
true  
false
```

```
System.out.println(a + " "+ b);
```

Compilation Successful

Input (stdin)

Your Output

```
true false
```

+ → act as a concatenation

L

```
String str = "SKDNGKDNGSKGN";
System.out.println(str);
```

```
String str1 = null;
System.out.println(str1);
```

### Your Output

```
SKDNGKDNGSKGN
null
```

## Rules for writing variable names

- ↳ Not start with nos.
- ↳ 1 variable , 1a, 2a, ~~X~~
- ↳ Start with any alphabet, — , \$
- ↳ meaningful →

Name of student = " Abc ";

a = ~~X~~  
b = ~~X~~

~~xx~~ Note:-  
↳ There is no concept of pointers in Java

break till 11:25 am

## Scanner Class

↳ For taking user defined input

→ `Scanner scn = new Scanner( System.in );`

- \* `integer` → `int a = scn.nextInt();`
- \* `long` → `int a = scn.nextLong();`



```
String str1 = scn.nextLine();
System.out.println(str1);

String str2 = scn.next();
System.out.println(str2);
```

Compilation Successful  
Input (stdin)  
→ (I am a student)  
I am a \_student

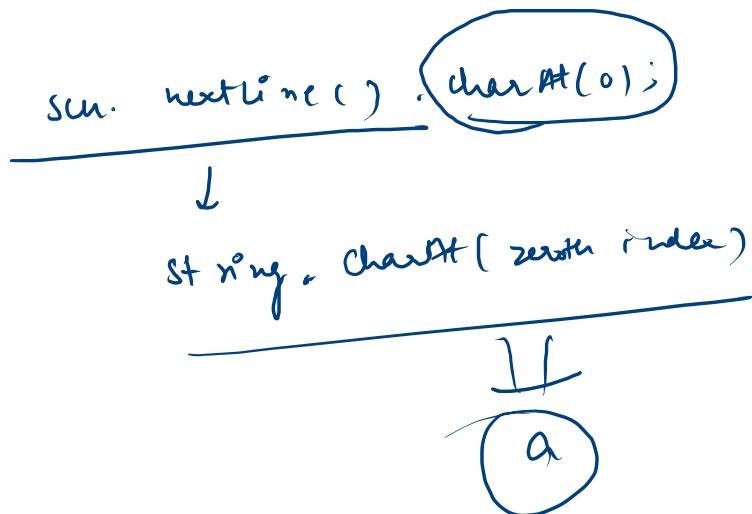
Your Output  
I am a student →

scn.nextLine() → reads complete line

scn.next() → it stops reading the input when encounters white space

```
// char ch = scn.nextCharacter();
char ch = scn.nextLine().charAt(0);

System.out.println("The character is: " + ch);
```



Q "a b c d e f"

## User defined input

```
Scanner scn = new Scanner(System.in);

int a = scn.nextInt();
System.out.println("The value of a is: " + a);

long b = scn.nextLong();
String message = "The value of b is: ";
System.out.println(message + b);

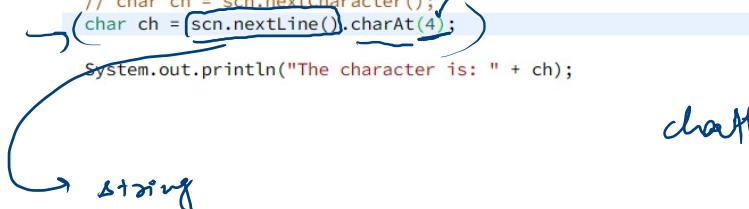
String str1 = scn.nextLine();
System.out.println(str1);

String str2 = scn.next();
System.out.println(str2);

boolean c = scn.nextBoolean();
System.out.println(c);

// char ch = scn.nextCharacter();
char ch = scn.nextLine().charAt(4);
System.out.println("The character is: " + ch);
```

char ch = 'a';



charAt() fn helps us to give a specific character.