

Chapter 4 :-

Math Class & Methods

- [1] pow(\square);
- [2] sqrt(\square); , cbrt(\square);
- [3] max(\square, \square); , min(\square, \square);
- [4] round(\square); , rint(\square); , floor(\square); , ceil(\square); , floorDiv(\square);
- [5] random(\square);
- [6] tan(\square);
- [7] Sin (\square); } radian angles
- [8] Cos (\square);
- [9] atan(\square); , atan2(\square) ; Polar Coordinates
- [10] asin(\square);
- [11] acos(\square);
- [12] tanh(\square); } radian angles
- [13] sinh(\square);
- [14] cosh(\square);
- [15] abs(\square);
- [16] PI(); , E();
- [17] exp(\square);
- [18] Log(\square); , log10(\square); , log1p(\square);
- [19] toRadians(\square); , toDegrees(\square);

Ceil (\square); round up to the nearest integer as a double

ex:

Math.Ceil (2.1); returns 2.0

Math.Ceil (2.0); returns 2.0

Math.Ceil (-2.0); returns -2.0

Math.Ceil (-2.1); returns -2.0

Floor (\square); round down to the nearest even integer as a double

ex:

Math.floor (2.1); returns 2.0

Math.floor (2.0); returns 2.0

Math.floor (-2.0); returns -2.0

Math.floor (-2.1); returns -4.0

rint (\square); round up to the nearest integer. if the number
close to 2 integers, the even one is returned

ex:

Math.rint (2.1); returns 2.0

Math.rint (-2.0); returns -2.0

Math.rint (-2.1); returns -2.0

Math.rint (2.5); returns 2.0

Math.rint (4.5); returns 4.0

Math.rint (-2.5); returns -2.0

`round()`; is the standard rounding with the range of number out of Decimal Point 5 - 9 ++

`2.6f`

ex: `Math.round(1)`; returns 3 int

`Math.round(2.0)`; returns 2 long

`Math.round(-2.0f)`; returns -2 int

`Math.round(-2.6)`; returns 3 long

`Math.round(-2.4)`; returns -2 long

`random()`; Generate random number between a and b

`0.9 [0, 1]`

forms `a + Math.random() * b;` $[a, a+b] \Leftrightarrow [a, a+b]$

Character Datatype and Operations

1

'0' to '9' 48 to 57 $\backslash u0030$ to $\backslash u0039$

2

'A' to 'Z' 65 to 90 $\backslash u0041$ to $\backslash u005a$

3

'a' to 'z' 97 to 122 $\backslash u0061$ to $\backslash u007a$ Carnegie

4

'\b' Back Space, '\t' Tab, '\n' newline, '\f' formfeed, '\r' return

'\' Backslash, '\"' double quote

note

You can make casting to char only with int datatype

and you can compare characters as their ASCII

Code Comparing

Character Class & Methods

- 1 isdigit();
- 2 toLowerCase(); , toUpperCase();
- 3 Compare();
- 4 isLetter();
- 5 isLowerCase(); , isUpperCase();
- 6 isLetterOrDigit();
- 7 toString();
- 8 valueOf();

String Class & Methods

- 1 length();
- 2 charAt();
- 3 toLowerCase(); , toUpperCase();
- 4 equals(); , equalsIgnoreCase();
- 5 indexOf(); , lastIndexOf();
- 6 trim();
- 7 Substring(); , startsWith(); , endsWith();
- 8 Contains();
- 9 compareTo(); , compareToIgnoreCase();
- 10 concat();
- 11 isEmpty(); , isBlank();
- 12 toString();
- 13 valueOf();

indexOf (); is a method which take a string or a character and it's first occurrence

indexOf (,); is a method which take a string or a character and the index begin and display its first occurrence from index which specified

lastIndexOf (); is a method which take a string or a character and returns its last occurrence of it from backward

lastIndexOf (,); is a method which take a string or a character and returns its last occurrence starting from the specified index

CompareTo (); is a method which returns 0 or positive or negative integer whether the 2 strings are equal or less or greater than each other in ASCII Code

ex: "abc".compareTo("bg"); returns -4
a=a, b=b, c < g \Rightarrow 96 - 98 = -2

Formatting Console Output :

Def:

is away to save format of any string and we use it instead of concatenation operators and we write it as following Syntax :

`System.out.printf("String", vars);`

and we denoted it by something called Format Specifier and we have some of them as following :

- [1] `%b` a boolean value
- [2] `%c` a character
- [3] `%d` a decimal integer
- [4] `%f` a floating Point number
- [5] `%e` a Number is standard Scientific Notation
- [6] `%s` a String
- [7] `%5c` add 4 spaces before character
- [8] `%6b` add space before false and 2 before true
- [9] `%5d` Print 5 digits of integer
- [10] `%10.2f` Print floating number and 2 digits after decimal
- [11] `%10.2e` Scientific Notation with at least 10 digits
- [12] `%12s` Print a string at least 12 characters