

Number Methods

1	xxxValue() Converts the value of <i>this</i> Number object to the xxx data type and returns it.
2	compareTo() Compares <i>this</i> Number object to the argument.
3	equals() Determines whether <i>this</i> number object is equal to the argument.
4	valueOf() Returns an Integer object holding the value of the specified primitive.
5	toString() Returns a String object representing the value of a specified int or Integer.
6	parseInt() This method is used to get the primitive data type of a certain String.
7	abs() Returns the absolute value of the argument.
8	ceil() Returns the smallest integer that is greater than or equal to the argument. Returned as a double.
9	floor() Returns the largest integer that is less than or equal to the argument.

	Returned as a double.
10	rint() Returns the integer that is closest in value to the argument. Returned as a double.
11	round() Returns the closest long or int, as indicated by the method's return type to the argument.
12	min() Returns the smaller of the two arguments.
13	max() Returns the larger of the two arguments.
14	exp() Returns the base of the natural logarithms, e, to the power of the argument.
15	log() Returns the natural logarithm of the argument.
16	pow() Returns the value of the first argument raised to the power of the second argument.
17	sqrt() Returns the square root of the argument.
18	sin() Returns the sine of the specified double value.

19	cos() Returns the cosine of the specified double value.
20	tan() Returns the tangent of the specified double value.
21	asin() Returns the arcsine of the specified double value.
22	acos() Returns the arccosine of the specified double value.
23	atan() Returns the arctangent of the specified double value.
24	atan2() Converts rectangular coordinates (x, y) to polar coordinate (r, theta) and returns theta.
25	toDegrees() Converts the argument to degrees.
26	toRadians() Converts the argument to radians.
27	random() Returns a random number.

Java - xxxValue() Method

```
byte byteValue()  
short shortValue()  
int intValue()  
long longValue()  
float floatValue()  
double doubleValue()
```

```
public static void main(String args[]) {  
    Integer x = 5;  
  
    // Returns byte primitive data type  
    System.out.println( x.byteValue() );  
  
    // Returns double primitive data type  
    System.out.println(x.doubleValue());  
  
    // Returns long primitive data type  
    System.out.println( x.longValue() );  
}
```

```
5  
5.0  
5
```

Java - compareTo() Method

The method compares the Number object that invoked the method to the argument. It is possible to compare Byte, Long, Integer

```
public int compareTo( NumberSubClass referenceName )
```

```
public static void main(String args[]) {  
    Integer x = 5;  
  
    System.out.println(x.compareTo(3));  
    System.out.println(x.compareTo(5));  
    System.out.println(x.compareTo(8));  
}
```

```
1  
0  
-1
```

Java - equals() Method

```
public boolean equals(Object o)
```

The method returns True if the argument is not null and is an object of the same type and with the same numeric value

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

```
Integer x = 5;

Integer y = 10;

Integer z = 5;

Short a = 5;


System.out.println(x.equals(y));

System.out.println(x.equals(z));

System.out.println(x.equals(a));

}
```

Output

```
false
true
false
```

Java - valueOf() Method

The valueOf method returns the relevant Number Object holding the value of the argument passed. The argument can be a primitive data type, String, etc.

This method is a static method. The method can take two arguments, where one is a String and the other is a radix

```
static Integer valueOf(int i)
static Integer valueOf(String s)
static Integer valueOf(String s, int radix)
```

- **i** – An int for which Integer representation would be returned.
- **s** – A String for which Integer representation would be returned.
- **radix** – This would be used to decide the value of returned Integer based on the passed String

Return Value

- **valueOf(int i)** – This returns an Integer object holding the value of the specified primitive.
- **valueOf(String s)** – This returns an Integer object holding the value of the specified string representation.
- **valueOf(String s, int radix)** – This returns an Integer object holding the integer value of the specified string representation, parsed with the value of radix.

```
public static void main(String args[]) {
    Integer x = Integer.valueOf(9);
    Double c = Double.valueOf(5);
    Float a = Float.valueOf("80");
    Integer b = Integer.valueOf("444",16);
}
```

```
System.out.println(x);  
System.out.println(c);  
System.out.println(a);  
System.out.println(b);  
}
```

Output

```
9  
5.0  
80.0  
1092
```

Java - toString() Method

The method is used to get a String object representing the value of the Number Object.

If the method takes a primitive data type as an argument, then the String object representing the primitive data type value is returned

```
String toString()  
static String toString(int i)
```

- **toString()** – This returns a String object representing the value of **this** Integer.

- **toString(int i)** – This returns a String object representing the specified integer.

```
public static void main(String args[]) {  
    Integer x = 5;  
  
    System.out.println(x.toString());  
    System.out.println(Integer.toString(12));  
}
```

Output

```
5  
12
```

Java - parseInt() Method

This method is used to get the primitive data type of a certain String. `parseXxx()` is a static method and can have one argument or two

```
static int parseInt(String s)  
static int parseInt(String s, int radix)
```

- **s** – This is a string representation of decimal.
- **radix** – This would be used to convert String s into integer

- **parseInt(String s)** – This returns an integer (decimal only).
- **parseInt(int i)** – This returns an integer, given a string representation of decimal, binary, octal, or hexadecimal (radix equals 10, 2, 8, or 16 respectively)

```
public static void main(String args[]) {  
    int x =Integer.parseInt("9");  
    double c = Double.parseDouble("5");  
    int b = Integer.parseInt("444",16);  
  
    System.out.println(x);  
    System.out.println(c);  
    System.out.println(b);  
}
```

Output

```
9  
5.0  
1092
```

Java - abs() Method

The method gives the absolute value of the argument. The argument can be int, float, long, double, short, byte.

```
double abs(double d)
float abs(float f)
int abs(int i)
long abs(long lng)
```

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

    Integer a = -8;

    double d = -100;

    float f = -90;


    System.out.println(Math.abs(a));

    System.out.println(Math.abs(d));

    System.out.println(Math.abs(f));

}
```

```
8
100.0
90.0
```

Java - ceil() Method

The method ceil gives the smallest integer that is greater than or equal to the argument.

```
double ceil(double d)
double ceil(float f)
```

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

    double d = -100.675;
```

```
float f = -90;

System.out.println(Math.ceil(d));

System.out.println(Math.ceil(f));

System.out.println(Math.floor(d));

System.out.println(Math.floor(f));

}

-100.0
-90.0
-101.0
-90.0
```

Java - floor() Method

The method floor gives the largest integer that is less than or equal to the argument

```
double floor(double d)
double floor(float f)
```

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

    double d = -100.675;

    float f = -90;

    System.out.println(Math.floor(d));

    System.out.println(Math.floor(f));

    System.out.println(Math.ceil(d));
```

```
System.out.println(Math.ceil(f));  
  
}  
-101.0  
-90.0  
-100.0  
-90.0
```

Java - rint() Method

The method rint returns the integer that is closest in value to the argument

```
double rint(double d)  
  
public static void main(String args[]) {  
    double d = 100.675;  
    double e = 100.500;  
    double f = 100.200;  
  
    System.out.println(Math.rint(d));  
    System.out.println(Math.rint(e));  
    System.out.println(Math.rint(f));  
  
}  
101.0  
100.0  
100.0
```

Java - round() Method

The method round returns the closest long or int, as given by the methods return type

```
long round(double d)
int round(float f)
```

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

    double d = 100.675;

    double e = 100.500;

    float f = 100;

    float g = 90f;


    System.out.println(Math.round(d));

    System.out.println(Math.round(e));

    System.out.println(Math.round(f));

    System.out.println(Math.round(g));

}
```

```
101
101
100
90
```

Java - min() Method

The method gives the smaller of the two arguments. The argument can be int, float, long, double.

```
double min(double arg1, double arg2)
float min(float arg1, float arg2)
int min(int arg1, int arg2)
long min(long arg1, long arg2)
```

```
public static void main(String args[]) {  
    System.out.println(Math.min(12.123, 12.456));  
    System.out.println(Math.min(23.12, 23.0));  
}  
12.123  
23.0
```

Java - max() Method

This method gives the maximum of the two arguments. The argument can be int, float, long, double

```
double max(double arg1, double arg2)  
float max(float arg1, float arg2)  
int max(int arg1, int arg2)  
long max(long arg1, long arg2)
```

```
public static void main(String args[]) {  
    System.out.println(Math.max(12.123, 12.456));  
    System.out.println(Math.max(23.12, 23.0));  
}  
12.456  
23.12
```

Java - exp() Method

The method returns the base of the natural logarithms, e, to the power of the argument

```
double exp(double d)
```

```
public static void main(String args[]) {  
    double x = 11.635;  
    double y = 2.76;  
  
    System.out.printf("The value of e is %.4f%n", Math.E);  
    System.out.printf("exp(%.3f) is %.3f%n", x, Math.exp(x));  
}
```

```
The value of e is 2.7183  
exp(11.635) is 112983.831
```

Java - log() Method

The method returns the natural logarithm of the argument.

```
double log(double d)
```

```
public static void main(String args[]) {  
    double x = 11.635;  
    double y = 2.76;  
  
    System.out.printf("The value of e is %.4f%n", Math.E);  
    System.out.printf("log(%.3f) is %.3f%n", x, Math.log(x));  
}
```

```
The value of e is 2.7183  
log(11.635) is 2.454
```


Java - pow() Method

```
double pow(double base, double exponent)
```

```
public static void main(String args[]) {  
    double x = 11.635;  
    double y = 2.76;  
  
    System.out.printf("The value of e is %.4f\n", Math.E);  
    System.out.printf("pow(%.3f, %.3f) is %.3f\n", x, y, Math.pow(x,  
y));  
}
```

Java - sqrt() Method

```
public static void main(String args[]) {  
    double x = 11.635;  
    double y = 2.76;  
  
    System.out.printf("The value of e is %.4f\n", Math.E);  
    System.out.printf("sqrt(%.3f) is %.3f\n", x, Math.sqrt(x));  
}
```

```
The value of e is 2.7183  
sqrt(11.635) is 3.411
```

Java - sin() Method

```
double sin(double d)
```

```

public static void main(String args[]) {

    double degrees = 45.0;

    double radians = Math.toRadians(degrees);

    System.out.format("The value of pi is %.4f%n", Math.PI);

    System.out.format("The sine of %.1f degrees is %.4f%n", degrees,
Math.sin(radians));

}

```

Java - cos() Method

```

double cos(double d)

public static void main(String args[]) {

    double degrees = 45.0;

    double radians = Math.toRadians(degrees);

    System.out.format("The value of pi is %.4f%n", Math.PI);

    System.out.format("The cosine of %.1f degrees is %.4f%n", degrees,
Math.cos(radians));

}

```

The value of pi is 3.1416
The cosine of 45.0 degrees is 0.7071

Java - tan() Method

The method returns the tangent of the specified double value

```

public static void main(String args[]) {

    double degrees = 45.0;

```

```
double radians = Math.toRadians(degrees);

System.out.format("The value of pi is %.4f%n", Math.PI);

System.out.format("The tangent of %.1f degrees is %.4f%n", degrees,
Math.tan(radians));

}
```

Output

```
The value of pi is 3.1416
The tangent of 45.0 degrees is 1.0000
```

Java - asin() Method

The method returns the arcsine of the specified double value.

```
double asin(double d)
```

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

    double degrees = 45.0;

    double radians = Math.toRadians(degrees);

    System.out.format("The value of pi is %.4f%n", Math.PI);

    System.out.format("The arcsine of %.4f is %.4f degrees %n",
Math.sin(radians),

        Math.toDegrees(Math.asin(Math.sin(radians))));

}
```

The value of pi is 3.1416
The arcsine of 0.7071 is 45.0000 degrees

Java - acos() Method

double acos(double d)

```
public static void main(String args[]) {  
    double degrees = 45.0;  
    double radians = Math.toRadians(degrees);  
  
    System.out.format("The value of pi is %.4f\n", Math.PI);  
    System.out.format("The arccosine of %.4f is %.4f degrees %n",  
Math.cos(radians),  
    Math.toDegrees(Math.acos(Math.cos(radians))));  
}  
The value of pi is 3.1416  
The arccosine of 0.7071 is 45.0000 degrees
```

Java - atan() Method

double atan(double d)

```

public static void main(String args[]) {

    double degrees = 45.0;

    double radians = Math.toRadians(degrees);


    System.out.format("The value of pi is very very very approximately
%.4f%n", Math.PI);

    System.out.format("The arctangent of %.4f is %.4f degrees %n",
Math.cos(radians),

    Math.toDegrees(Math.atan(Math.sin(radians))));

}

```

The value of pi is very very very approximately 3.1416
The arctangent of 0.7071 is 35.2644 degrees

Java - atan2() Method

The method converts rectangular coordinates (x, y) to polar coordinate (r, theta) and returns theta.

- **X** – X co-ordinate in double data type.
- **Y** – Y co-ordinate in double data type.

```
public static void main(String args[]) {  
  
    double x = 45.0;  
  
    double y = 30.0;  
  
  
    System.out.println( Math.atan2(x, y) );  
  
}
```

0.982793723247329

Java - toDegrees() Method

```
double toDegrees(double d)
```

```
public static void main(String args[]) {  
  
    double x = 45.0;  
  
    double y = 30.0;  
  
  
    System.out.println( Math.toDegrees(x) );  
  
    System.out.println( Math.toDegrees(y) );  
  
}
```

2578.3100780887044
1718.8733853924698

Java - toRadians() Method

```
double toRadians(double d)

public static void main(String args[]) {

    double x = 45.0;

    double y = 30.0;


    System.out.println( Math.toRadians(x) );

    System.out.println( Math.toRadians(y) );

}
```

```
0.7853981633974483
0.5235987755982988
```

Java - random() Method

The method is used to generate a random number between 0.0 and 1.0. The range is: $0.0 \leq \text{Math.random()} < 1.0$. Different ranges can be achieved by using arithmetic operations

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

    System.out.println( Math.random() );

    System.out.println( Math.random() );

}
```

0.16763945061451657
0.400551253762343