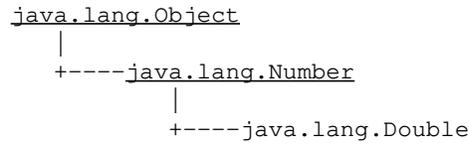


Class `java.lang.Double`



public final class **Double**
extends [Number](#)

The Double class provides an object wrapper for Double data values and serves as a place for double-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since doubles are not objects in Java, they need to be "wrapped" in a Double instance.

Version:

1.29, 10/02/95

Author:

Lee Boynton, Arthur van Hoff

Variable Index

- o **MAX VALUE**
The maximum value a double can have.
- o **MIN VALUE**
The minimum value a double can have.
- o **NEGATIVE INFINITY**
Negative infinity.
- o **NaN**
Not-a-Number.
- o **POSITIVE INFINITY**
Positive infinity.

Constructor Index

- o **Double**(double)
Constructs a Double wrapper for the specified double value.

Method Index

- o **doubleToLongBits**(double)
Returns the bit representation of a double–float value
- o **doubleValue**()
Returns the double value of this Double.
- o **equals**(Object)
Compares this object against the specified object.
- o **floatValue**()
Returns the float value of this Double.
- o **hashCode**()
Returns a hashcode for this Double.
- o **intValue**()
Returns the integer value of this Double (by casting to an int).
- o **isInfinite**(double)
Returns true if the specified number is infinitely large in magnitude.
- o **isInfinite**()
Returns true if this Double value is infinitely large in magnitude.
- o **isNaN**(double)
Returns true if the specified number is the special Not–a–Number (NaN) value.
- o **isNaN**()
Returns true if this Double value is the special Not–a–Number (NaN) value.
- o **longBitsToDouble**(long)
Returns the double–float corresponding to a given bit representation.
- o **longValue**()
Returns the long value of this Double (by casting to a long).
- o **toString**()
Returns a String representation of this Double object.
- o **valueOf**(String)
Returns a new Double value initialized to the value represented by the specified String.

Variables

o **POSITIVE_INFINITY**

```
public final static double POSITIVE_INFINITY
```

Positive infinity.

o **NEGATIVE_INFINITY**

```
public final static double NEGATIVE_INFINITY
```

Negative infinity.

o **NaN**

```
public final static double NaN
```

Not-a-Number. *Note: is not equal to anything, including itself*

o **MAX_VALUE**

```
public final static double MAX_VALUE
```

The maximum value a double can have. The greatest maximum value that a double can have is 1.79769313486231570e+308d.

o **MIN_VALUE**

```
public final static double MIN_VALUE
```

The minimum value a double can have. The lowest minimum value that a double can have is 4.94065645841246544e-324d.

Constructors

o **Double**

```
public Double(double value)
```

Constructs a Double wrapper for the specified double value.

Parameters:

value – the initial value of the double

Methods

o **isNaN**

```
public static boolean isNaN(double v)
```

Returns true if the specified number is the special Not-a-Number (NaN) value.

Parameters:

v – the value to be tested

o **isInfinite**

```
public static boolean isInfinite(double v)
```

Returns true if the specified number is infinitely large in magnitude.

Parameters:

v – the value to be tested

o **isNaN**

```
public boolean isNaN()
```

Returns true if this Double value is the special Not-a-Number (NaN) value.

o **isInfinite**

```
public boolean isInfinite()
```

Returns true if this Double value is infinitely large in magnitude.

o **toString**

```
public String toString()
```

Returns a String representation of this Double object.

Overrides:

toString in class Object

o **intValue**

```
public int intValue()
```

Returns the integer value of this Double (by casting to an int).

Overrides:

intValue in class Number

o **longValue**

```
public long longValue()
```

Returns the long value of this Double (by casting to a long).

Overrides:

longValue in class Number

o **floatValue**

```
public float floatValue()
```

Returns the float value of this Double.

Overrides:

floatValue in class Number

o **doubleValue**

```
public double doubleValue()
```

Returns the double value of this Double.

Overrides:

doubleValue in class Number

o hashCode

```
public int hashCode()
```

Returns a hashcode for this Double.

Overrides:

hashCode in class Object

o equals

```
public boolean equals(Object obj)
```

Compares this object against the specified object.

Note: To be useful in hashtables this method considers two NaN double values to be equal. This is not according to IEEE specification

Parameters:

obj – the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

o valueOf

```
public static Double valueOf(String s) throws NumberFormatException
```

Returns a new Double value initialized to the value represented by the specified String.

Parameters:

s – the String to be parsed

Throws: NumberFormatException

If the String cannot be parsed.

o doubleToLongBits

```
public static long doubleToLongBits(double value)
```

Returns the bit representation of a double–float value

o longBitsToDouble

```
public static double longBitsToDouble(long bits)
```

Returns the double–float corresponding to a given bit representation.

[All Packages](#)

[This Package](#)

[Previous](#)

[Next](#)