# java.util.Date

A Date object represents a precise moment in time, down to the millisecond. Dates are represented as a long that counts the number of milliseconds since midnight, January 1, 1970, Greenwich Mean Time.

Does this have a year 2000 problem? If so in what year?

To create a Date object for the current date and time use the noargs Date() constructor like this:

Date now = new Date();

To create a Date object for a specific time, pass the number of milliseconds since midnight, January 1, 1970, Greenwich Meantime to the constructor, like this:

Date midnight\_jan2\_1970 = new Date(24L\*60L\*60L\*1000L);

You can return the number of milliseconds in the Date as a long, using the getTime() method. For example, to time a block of code, you might do this

Date d1 = new Date();  
// timed code goes here  
Date d2 = new Date();  
long elapsed\_time = d2.getTime() - d1.getTime();  
System.out.println("That took " + elapsed\_time + " milliseconds");

You can change a Date by passing the new date as a number of milliseconds since midnight, January 1, 1970, GMT, to the setTime() method, like this:

Date midnight\_jan2\_1970 = new Date();  
midnight\_jan2\_1970.setTime(24L\*60L\*60L\*1000L);

The before() method returns true if this Date is before the Date argument, false if it's not. For example

if (midnight\_jan2\_1970.before(new Date())) {

The after() method returns true if this Date is after the Date argument, false if it's not. For example

if (midnight\_jan2\_1970.after(new Date())) {

The Date class also has the usual hashCode(), equals(), and toString() methods.

[Previous](http://docs.google.com/67.html) | [Next](http://docs.google.com/69.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified July 24, 1997