**Date format in Java**

[**JAVA DATE**](http://beginnersbook.com/category/technology/java-guide/java-date/)

[toc]The objective of this document is to highlight the efficient ways available in JAVA language for formatting and manipulating dates. The document provides different [**date**](http://beginnersbook.com/2013/05/java-date/) formats used in day today work. It also provides ways to calculate [**date differences**](http://beginnersbook.com/2013/04/java-date-difference/), date conversions, [**date validations**](http://beginnersbook.com/2013/05/java-date-validation/). The code snippets are also available which can be reused.

**Date Formats in Java**

We need the dates to be formatted in many different ways. Below is some of the formatting which are done on the date.

**Date in MMddyyyy Format**

This method formats the date in mmddyyyy format. The input to the method is the date to be formatted with the delimiter. The method returns the[**string representation of the Date**](http://beginnersbook.com/2013/05/java-date-string-conversion/) object in ddmmyyyy format.

[code language=”java”]  
/\*\*  
\* Returns the string representation of the Date object in  
\* mmddyyyy format. The month, day and year portions  
\* are kept separated by a delimiter  
\* @param a delimiter  
\* @param a Date object.  
\* @return a string in the desired format.  
\*/  
public static String dateToMmddyyyy(char charDelimiter, Date objDate)  
{  
String strDateString = null;  
if(objDate != null) {  
SimpleDateFormat objFormatter =  
new SimpleDateFormat  
("MM"+charDelimiter+"dd"+charDelimiter+"yyyy");  
strDateString = objFormatter.format(objDate);  
}  
else{  
strDateString = new String(" ");  
}  
return(strDateString);  
}  
[/code]

**Date in ddMMyyyy Format**

Returns the string representation of the Date object in ddmmyyyy format. The day,month and year portions are kept separated by a delimiter. For example : if the Date parameter represents a date like 31st January 2013 and the delimiter is ‘/’ then this function will return a string like “31/01/2013”. This method uses the [**SimpleDateFormat**](http://beginnersbook.com/2013/05/simple-date-format-java/)class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("dd"+charDelimiter+"MM"+charDelimiter+"yyyy");  
[/code]

**Date in yyyyMMdd Format**

Returns the string representation of the Date object in yyyymmdd format. The day,month and year portions are kept separated by a delimiter. For example : if the Date parameter represents a date like 31st January 2013 and the delimiter is ‘/’ then this function will return a string like “2013/01/31”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("yyyy"+charDelimiter+"MM"+charDelimiter+"dd");  
[/code]

**Date in ddMMyy Format**

Returns the string representation of the Date object in ddmmyy format. The day,month and year portions are kept separated by a delimiter.  
For example : if the Date parameter represents a date like 31st January 2013 and the delimiter is ‘/’ then this function will return a string like “31/01/13”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("dd"+charDelimiter+"MM"+charDelimiter+"yy");  
[/code]

**Date in yyMMdd Format**

Returns the string representation of the Date object in yymmdd format. The day,month and year portions are kept separated by a delimiter.  
For example : if the Date parameter represents a date like 31st January 2013 and the delimiter is ‘/’ then this function will return a string like “13/01/31”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("yy"+charDelimiter+"MM"+charDelimiter+"dd");  
[/code]

**Date in yyyyMMdd Format**

Returns the string representation of the Date object in yyyymmdd format. No delimiter is used here to separate the year, month and day portion. For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “20130131”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("yyyy"+"MM"+"dd");  
[/code]

**Date in ddMMyyyy Format**

Returns the string representation of the Date object in ddmmyyyy format. No delimiter is used here to separate the year, month and day portion. For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “31012013”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("dd"+"MM"+"yyyy");  
[/code]

**Date in MMddyyyy Format**

Returns the string representation of the Date object in mmddyyyy format. No delimiter is used here to separate the year, month and day portion. For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “01312013”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("MM"+"dd"+"yyyy");  
[/code]

**Date in yyMMdd Format**

Returns the string representation of the Date object in yymmdd format. No delimiter is used here to separate the year, month and day portion.  
For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “130131”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("yy"+"MM"+"dd");  
[/code]

**Date in ddMMyy Format**

Returns the string representation of the Date object in ddmmyy format. No delimiter is used here to separate the year, month and day portion.  
For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “310113”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat("dd"+"MM"+"yy");  
[/code]

**Date in MMddyy Format**

Returns the string representation of the Date object in mmddyy format. No delimiter is used here to separate the year, month and day portion.  
For example : if the Date parameter represents a date like 31st January 2013 then this function will return a string like “013113”. This method uses the SimpleDateFormat class of java.text package. By replacing the below lines of code in 2.1 will create the new method.

[code language=”java”]  
SimpleDateFormat objFormatter =  
new SimpleDateFormat ("MM"+"dd"+"yy");  
[/code]

**Month in MM format**

The below method returns the two digit month portion of a Date object in String format. For example if the Date object represents 31st January 2013 then this function should return “01”.

[code language=”java”]  
/\*\*  
\* Returns the string representation of the month .  
\* @param a Date object .  
\* @return a string in the desired format.  
\* @exception  
\* @see java.util.Date,java.util.Calendar  
\*/  
public static String getMonth(Date objDate){  
int intMonth = 0;  
String strMonth = null;  
if(objDate != null) {  
SimpleDateFormat objSimpleDateFormat  
= new SimpleDateFormat("MM");  
strMonth = objSimpleDateFormat.format(objDate);  
}  
else{  
strMonth = new String(" ");  
}  
return strMonth;  
}  
[/code]

**Day in dd format**

The below method returns the two digit day portion of a Date object in String format. For example if the Date object represents 31st January 2013 then this function should return “31”.

[code language=”java”]  
/\*\*  
\* Returns the number representation of the day .  
\* @param a Date object .  
\* @return a string in the desired format.  
\* @exception  
\* @see java.util.Date,java.util.Calendar  
\*/  
public static String getDay(Date objDate){  
int intDay = 0;  
String strDay = null;  
if(objDate != null) {  
SimpleDateFormat objSimpleDateFormat =  
new SimpleDateFormat("dd");  
strDay = objSimpleDateFormat.format(objDate);  
}  
else{  
strDay = new String(" ");  
}  
return strDay;  
}[/code]

**Day in ddd format**

The below method returns the day portion of a Date object in String format. It returns Mon for Monday, Sun for Sunday etc. Returns the day portion of a Date object in three character String format. For example if the Date object represents 31st January 2013 and it’s a Monday then this function should return “MON”.

[code language=”java”]  
/\*\*  
\* Returns the string representation of the day.  
\* @param a Date object.  
\* @return a string in the desired format.  
\* @exception  
\* @see java.util.Date,java.util.Calendar  
\*/  
public static String getDayOfWeek(Date objDate){  
int intDay = 0;  
String strDay = null;  
if(objDate != null) {  
SimpleDateFormat objSimpleDateFormat =  
new SimpleDateFormat("EEE");  
strDay = objSimpleDateFormat.format(objDate);  
}  
else{  
strDay = new String(" ");  
}  
return strDay;  
}[/code]

**Year in yyyy format**

Returns the four digit year portion of a Date object in String format. For example if the Date object represents 31st January 2013 then this function should return “2013”.

[code language=”java”]  
/\*\*  
\* Returns the string representation of the year .  
\* @param a Date object .  
\* @return a string in the desired format.  
\* @exception  
\* @see java.util.Date,java.util.Calendar  
\*/  
public static String getYear(Date objDate){  
int intYear = 0;  
String strYear = null;  
if(objDate != null) {  
SimpleDateFormat objSimpleDateFormat =  
new SimpleDateFormat("yyyy");  
strYear = objSimpleDateFormat.format(objDate);  
}  
else{  
strYear = new String(" ");  
}  
return strYear;  
}[/code]

**Year in yy format**

Returns the string representation of the year.

[code language=”java”]  
SimpleDateFormat objSimpleDateFormat =  
new SimpleDateFormat("yy");  
[/code]

**First date of the month**

The below method returns the first date of the month for the given date. For example if the Date object represents 31st January 2013 then the method will return 1st January 2013.

[code language=”java”]  
/\*\*  
\* Returns the first day of the month.  
\* @param a Date .  
\* @return the first Day of the month.  
\* @exception  
\* @see java.util.Date,java.util.Calendar  
\*/  
public static java.sql.Date getFirstDay(Date objDate) {  
java.sql.Date objSQLDate = null;  
long lTime = 0;  
if(objDate != null){  
Calendar objCalendar = Calendar.getInstance();  
objCalendar.setTime(objDate);  
objCalendar.set(objCalendar.get(Calendar.YEAR),  
objCalendar.get(Calendar.MONTH),1);  
lTime = (objCalendar.getTime()).getTime();  
objSQLDate = new java.sql.Date(lTime);  
}  
return objSQLDate;  
}[/code]

**Util date to SQL date**

This method converts the java.util.date to java.sql.date.

[code language=”java”]  
/\*\*  
\* Converts a java.util.Date to java.sql.Date.  
\* @param a java.util.Date .  
\* @return a java.sql.Date  
\* @exception  
\* @see java.util.Date,java.sql.Date  
\*/  
public static java.sql.Date utilDateToSqlDate(Date objDate){  
return (new java.sql.Date(objDate.getTime()));  
}[/code]

**Time in HH:MM:SS format**

This method returns the current time in a string in the format HH:MM:SS.

[code language=”java”]  
/\*\*  
\* Returns the curent time in HH:MM:SS format.  
\* @param void .  
\* @return a string in HH:MM:SS format  
\* @exception  
\* @see java.util.Calendar,java.sql.Time  
\*/  
public static String currentTime(){  
Time objCurrentTime = new Time(System.currentTimeMillis());  
return objCurrentTime.toString();  
}[/code]