

# Functions Sample

This sample illustrates some of the built-in functions

## Date & Time - Category for date and time manipulation functions

Name	Description	Parameters
DATE	Creates a date object using the specified information on day, month and year. Return type: java.util.Date	Year* - The year of the new date. Type: java.lang.Integer Month* - The month of the new date. Type: java.lang.Integer Day of month* - The day of the new date. Type: java.lang.Integer
DATEFORMAT	Format the specified date object using the chosen format pattern. Return type: java.lang.String	Selected date* - The date to format. Type: java.util.Date Format pattern* - Format pattern to apply when printing the date. Type: java.lang.String
DATERANGE	Allows to create a JasperReports DateRange instance starting from either a String expression or a Date instance. Return type: net.sf.jasperreports.types.date.DateRange	Date range details* - The date range information. Type: java.lang.Object
DATEVALUE	Gives the corresponding numeric value (long milliseconds) for a specified date object. Return type: java.lang.Long	Date object* - The object representing the date. Type: java.lang.Object
DAY	Returns the day of a given date. Date object can be a String, long value (milliseconds) or Date instance itself. Return type: java.lang.Integer	Selected date* - The object representing the date. Type: java.lang.Object
DAYS	Returns the number of days between two dates. Return type: java.lang.Integer	Start date* - The initial date. Type: java.lang.Object End date* - The end date. Type: java.lang.Object
DAYSINMONTH	Returns the number of days in a month. Return type: java.lang.Integer	Selected date* - The date to check. Type: java.lang.Object
DAYSINYEAR	Returns the number of days in a year. Return type: java.lang.Integer	Selected date* - The date to check. Type: java.lang.Object
EDATE	Returns a date a number of months away. Return type: java.util.Date	Selected date* - The object representing the date. Type: java.lang.Object Months* - The number of months after the given date. Type: java.lang.Integer
HOUR	Returns the hour (0-23) of the day for a given date. Date object can be a String, long value (milliseconds) or Date instance itself. Return type: java.lang.Integer	Selected date* - The object representing the date. Type: java.lang.Object
ISLEAPYEAR	Checks if the given date occurs in a leap year. Return type: java.lang.Boolean	Selected date* - The date to check. Type: java.lang.Object
MINUTE	Returns the minute (0-59) of the hour for a given date. Date object can be a String, long value (milliseconds) or Date instance itself. Return type: java.lang.Integer	Selected date* - The object representing the date. Type: java.lang.Object
MONTH	Returns the month of a given date. Date object can be a String, long value (milliseconds) or Date instance itself. Return type: java.lang.Integer	Selected date* - The object representing the date. Type: java.lang.Object
MONTHS	Returns the number of months between two dates. Return type: java.lang.Integer	Start date* - The initial date. Type: java.lang.Object End date* - The end date. Type: java.lang.Object
NETWORKDAYS	Returns the number of working days between two	

## Date & Time - Category for date and time manipulation functions

Name	Description	Parameters
	<p>dates (inclusive). Saturday and Sunday are not considered working days.</p> <p>Return type: java.lang.Integer</p>	<p>Start date* - The initial date. Type: java.lang.Object</p> <p>End date* - The end date. Type: java.lang.Object</p>
NOW	<p>Returns the current instant as date object</p> <p>Return type: java.util.Date</p>	<ul style="list-style-type: none"> <li>This function has no parameters.</li> </ul>
SECOND	<p>Returns the second (0-59) of the minute for a given date. Date object can be a String, long value (milliseconds) or Date instance itself.</p> <p>Return type: java.lang.Integer</p>	<p>Selected date* - The object representing the date. Type: java.lang.Object</p>
TIME	<p>Returns a text string representing a time value (hours, seconds and minutes). If no specific pattern is provided a default formatter is used.</p> <p>Return type: java.lang.String</p>	<p>Hours* - The hours for the new time value. Type: java.lang.Integer</p> <p>Minutes* - The minutes for the new time value. Type: java.lang.Integer</p> <p>Seconds* - The seconds for the new time value. Type: java.lang.Integer</p> <p>Format pattern - The pattern to format the time value. Type: java.lang.String</p>
TODAY	<p>Returns the current date as date object</p> <p>Return type: java.util.Date</p>	<ul style="list-style-type: none"> <li>This function has no parameters.</li> </ul>
WEEKDAY	<p>Returns the day of the week for a given date. Date object can be a String, long value (milliseconds) or Date instance itself.</p> <p>Return type: java.lang.Integer</p>	<p>Selected date* - The object representing the date. Type: java.lang.Object</p> <p>Sunday is first day - Boolean flag to decide if Sunday should be considered as first day. Default is not. Type: java.lang.Boolean</p>
WEEKNUM	<p>Returns the week number of a given date.</p> <p>Return type: java.lang.Integer</p>	<p>Selected date* - The date to check. Type: java.lang.Object</p>
WEEKS	<p>Returns the number of weeks between two dates.</p> <p>Return type: java.lang.Integer</p>	<p>Start date* - The initial date. Type: java.lang.Object</p> <p>End date* - The end date. Type: java.lang.Object</p>
WEEKSINYEAR	<p>Returns the number of weeks in a year.</p> <p>Return type: java.lang.Integer</p>	<p>Selected date* - The date to check. Type: java.lang.Object</p>
WORKDAY	<p>Returns a date a number of workdays away. Saturday and Sundays are not considered working days.</p> <p>Return type: java.util.Date</p>	<p>Selected date* - The object representing the date. Type: java.lang.Object</p> <p>Working days* - The number of days after the given date. Type: java.lang.Integer</p>
YEAR	<p>Returns the year of a given date. Date object can be a String, long value (milliseconds) or Date instance itself.</p> <p>Return type: java.lang.Integer</p>	<p>Selected date* - The object representing the date. Type: java.lang.Object</p>
YEARS	<p>Returns the number of years between two dates.</p> <p>Return type: java.lang.Integer</p>	<p>Start date* - The initial date. Type: java.lang.Object</p> <p>End date* - The end date. Type: java.lang.Object</p>

Logical - Category for logical operations functions

Name	Description	Parameters
AND	Returns true if all arguments are considered true, false otherwise. Return type: java.lang.Boolean	Argument* - A boolean expression or value. Type: [Ljava.lang.Boolean;
EQUALS	Checks if the two specified objects are equal. Return type: java.lang.Boolean	Object 1* - The first element to be compared. Type: java.lang.Object Object 2* - The second element to be compared. Type: java.lang.Object
FALSE	Returns the logical value FALSE. Return type: java.lang.Boolean	• This function has no parameters.
IF	Returns one of two values, depending on a test condition. Return type: java.lang.Object	Test condition* - An expression returning a boolean value. Type: java.lang.Boolean Value 1 (true)* - The value returned when the test is true. Type: java.lang.Object Value 2 (false)* - The value returned when the test is false. Type: java.lang.Object
NOT	Returns the negation of the specified boolean expression. Return type: java.lang.Boolean	Argument* - A boolean expression or value. Type: java.lang.Boolean
OR	Returns true if any of the arguments is considered true, false otherwise. Return type: java.lang.Boolean	Argument* - A boolean expression or value. Type: [Ljava.lang.Boolean;
TRUE	Returns the logical value TRUE. Return type: java.lang.Boolean	• This function has no parameters.

## Numeric / Mathematical - Category for mathematical operations functions

Name	Description	Parameters
ABS	Returns the absolute value of a number. Return type: java.lang.Number	Number* - The number to check. Type: java.lang.Number
CEIL	Returns the smallest value that is greater than or equal to the argument and is equal to a mathematical integer. Return type: java.lang.Double	Number* - Value. Type: java.lang.Number
FACT	Returns the factorial of a number. Return type: java.lang.Long	Integer number* - The argument. Type: java.lang.Integer
FLOOR	Returns the largest value that is less than or equal to the argument and is equal to a mathematical integer. Return type: java.lang.Double	Number* - Value. Type: java.lang.Number
ISEVEN	Checks if a number is even. If a non-integer number is specified, any digits after the decimal point are ignored. Return type: java.lang.Boolean	Number* - The number to check. Type: java.lang.Number
ISODD	Checks if a number is odd. If a non-integer number is specified, any digits after the decimal point are ignored. Return type: java.lang.Boolean	Number* - The number to check. Type: java.lang.Number
MAX	Returns the maximum of a list of numeric values. Return type: java.lang.Number	Number* - Number to compare. Type: [Ljava.lang.Number;
MIN	Returns the minimum of a list of numeric values. Return type: java.lang.Number	Number* - Number to compare. Type: [Ljava.lang.Number;
PRODUCT	Returns the product of a list of numbers. Return type: java.lang.Number	Number* - Argument. Type: [Ljava.lang.Number;
RAND	Returns a random number between 0.0 and 1.0. Return type: java.lang.Double	• This function has no parameters.
RANDBETWEEN	Returns an Integer random number between bottom and top range (both inclusive). Return type: java.lang.Integer	Bottom range* - Integer number for the bottom range. Type: java.lang.Integer Top range* - Integer number for the top range. Type: java.lang.Integer
SIGN	Returns the sign of a number. Return type: java.lang.Integer	Number* - The number to check. Type: java.lang.Number
SQRT	Returns the positive square root of a number. The number must be positive. Return type: java.lang.Number	Positive number* - Argument. Type: java.lang.Number
SUM	Returns the sum of a list of numbers. Return type: java.lang.Number	Number* - Addendum. Type: [Ljava.lang.Number;

## Text - Category for text/string manipulation functions

Name	Description	Parameters
BASE	Returns a text representation of a number, in a specified base radix. Return type: java.lang.String	Number* - The positive integer number to convert. Type: java.lang.Integer Radix* - The base radix, an integer between 2 and 36. Type: java.lang.Integer Minimum length - Specifies the minimum number of characters returned; zeroes are added on the left if necessary. Type: java.lang.Integer
CHAR	Returns a single text character, given a character code. Return type: java.lang.String	Char code* - The character code, in the range 1-255. Type: java.lang.Integer
CLEAN	Returns a new text string without non-printable characters. Return type: java.lang.String	Text* - The text to be cleaned. Type: java.lang.String
CODE	Returns the numeric code (0-255) for the first character in a string. Return type: java.lang.Integer	Text* - The string containing the character to convert. Type: java.lang.String
CONCATENATE	Combines a list of strings into a single one. Return type: java.lang.String	Text* - Argument. Type: [Ljava.lang.String;
DOUBLE_VALUE	Returns a Double number representing the given text string. Return type: java.lang.Double	Number (as text)* - The input text string representing a number. Type: java.lang.String
EXACT	Returns TRUE if the two text specified are exactly the same (case sensitive compare). Return type: java.lang.Boolean	Text 1* - The first text to compare. Type: java.lang.String Text 2* - The second text to compare. Type: java.lang.String
FIND	Returns the character position of a string inside another text. If the text is not found then -1 is returned. Return type: java.lang.Integer	Find text* - The text to look into. Type: java.lang.String Text to search* - The text string to search. Type: java.lang.String Start position - The position from which the search should start. Type: java.lang.Integer
FIXED	Returns the text representing number with the specified decimal places. Return type: java.lang.String	Number* - The number to print out. Type: java.lang.Number Decimals* - The number of decimal places. Type: java.lang.Integer Omit separators - The flag to specify if the thousands separators should be included or not. Type: java.lang.Boolean
FLOAT_VALUE	Returns a Float number representing the given text string. Return type: java.lang.Float	Number (as text)* - The input text string representing a number. Type: java.lang.String
INTEGER_VALUE	Returns an Integer number representing the given text string. Return type: java.lang.Integer	Number (as text)* - The input text string representing a number. Type: java.lang.String
LEFT	Returns the specified number of characters (1 by default) from the left side of the input text. Return type: java.lang.String	Text* - The input text. Type: java.lang.String Characters num - The number of characters. Default (not specified) is 1. Type: java.lang.Integer
LEN	Returns the length of the specified text string. Return type: java.lang.Integer	Text* - The input text string. Type: java.lang.String
LONG_VALUE	Returns a Long number representing the given text string. Return type: java.lang.Long	Number (as text)* - The input text string representing a number. Type: java.lang.String
LOWER	Performs the lower case conversion of the specified text string.	Text* - The input text string. Type: java.lang.String

## Text - Category for text/string manipulation functions

Name	Description	Parameters
	Return type: java.lang.String	
LTRIM	Clear a string, removing leading whitespaces. Return type: java.lang.String	Text* - The text string to be trimmed. Type: java.lang.String
MID	Returns the text from the middle of a text string. Return type: java.lang.String	Text* - The input text. Type: java.lang.String Start* - The initial position to extract the text. Type: java.lang.Integer Characters num - The number of characters. Type: java.lang.Integer
PROPER	Capitalizes each words of the specified text. The remaining parts of words are in lowercase. Return type: java.lang.String	Text* - The input text. Type: java.lang.String
REPLACE	Replaces parts of a text string with a different one. Starting from a specified position, removes a certain number of characters and then insert the new text. Return type: java.lang.String	Original Text* - The input text to modify. Type: java.lang.String Start position* - The number of characters. Default (not specified) is 1. Type: java.lang.Integer Characters num* - The number of characters to remove. Type: java.lang.Integer New Text* - The text that will replace the old one. Type: java.lang.String
REPT	Replicates an input text string for a specified number of times. Return type: java.lang.String	Original Text* - The input text to replicate. Type: java.lang.String Number of copies* - The needed number of copies. Type: java.lang.Integer
RIGHT	Returns the specified number of characters (1 by default) from the right side of the input text. Return type: java.lang.String	Text* - The input text. Type: java.lang.String Characters num - The number of characters. Default (not specified) is 1. Type: java.lang.Integer
RTRIM	Clear a string, removing trailing whitespaces. Return type: java.lang.String	Text* - The text string to be trimmed. Type: java.lang.String
SEARCH	Returns the position of a string of text in another string. Search is not case-sensitive. Return type: java.lang.Integer	Find Text* - The text to find. Type: java.lang.String Text to search* - The text to look into. Type: java.lang.String Start position - The initial position. Type: java.lang.Integer
SUBSTITUTE	Substitutes new text for old text in a text string. When no occurrence is specified all occurrences are replaced. Return type: java.lang.String	Original Text* - The text to be modified. Type: java.lang.String Old Text* - The old text to be replaced. Type: java.lang.String New Text* - The new text that will replace the old one. Type: java.lang.String Occurrence - The occurrence of old text to be replaced. Type: java.lang.Integer
T	Returns the text string if the value is a string, otherwise an empty string is returned. Return type: java.lang.String	Generic value* - The object value to be tested. Type: java.lang.Object
TEXT	Converts a number into a text string according to a specified format. Return type: java.lang.String	Number* - The number to be formatted. Type: java.lang.Number Format* - The format pattern. Type: java.lang.String

**Text - Category for text/string manipulation functions**

Name	Description	Parameters
TRIM	Clear a string,removing leading and trailing whitespaces. Return type: java.lang.String	Text* - The text string to be trimmed. Type: java.lang.String
UPPER	Performs the upper case conversion of the specified text string. Return type: java.lang.String	Text* - The input text string. Type: java.lang.String

# Functions Examples

Date & Time Functions	Expression	Result
Current date	DATEFORMAT(NOW(), "EEEE, MMM d, yyyy")	joi, Oct 24, 2013
Current year	YEAR(TODAY())	2013
Four months later	DATEFORMAT(EDATE(TODAY(), 4), "MM/dd/yyyy")	02/24/2014
Five years ago	DATEFORMAT(EDATE(TODAY(), -60), "MMMM d, yyyy")	octombrie 24, 2008
Current week number	WEEKNUM(TODAY())	43
Next working day	DATEFORMAT(WORKDAY(TODAY(),1), "EEE, MMM dd")	V, Oct 25
Years since 2000	YEARS(DATE(2000, 1, 1), NOW())	13
Current time	DATEFORMAT(TODAY(), "hh:mm:ss a")	11:19:42 AM

Logical Functions	Expression	Result
This is true.	TRUE()	true
This is false.	FALSE()	false
3 < 5 AND 3 > 7	AND(3 < 5, 3 > 7)	false
3 < 5 OR 3 > 7	OR(3 < 5, 3 > 7)	true
Current month is before July	IF(MONTH(TODAY()) < 7, "YES", "NO")	NO

Numeric / Mathematical Functions	Expression	Result
-245   is even	ISEVEN(ABS(-245))	false
14.4 * 22.56 * 10 * 34	PRODUCT(14.4, 22.56, 10, 34)	110.453,76
Random number between 0 and 1000	1000 * RAND()	91,407
Factorial of 5	FACT(5)	120
SQRT(6.25)	SQRT(6.25)	2,50
Max value in [4,6,2,3,9,4,1,7,3,8]	MAX(4, 6, 2, 3, 9, 4, 1, 7, 3, 8)	9

Text Functions	Expression	Result
Concatenate "Hello", " ", "World"	CONCATENATE("Hello", " ", "World")	Hello World
Case sensitive search for "II" in "Hello"	"Found on position " + (FIND("Hello", "II") + 1)	Found on position 3
Case insensitive search for "LL" in "all"	"Found on position " + (SEARCH("LL", "all") + 1)	Found on position 2
Lower case for "Hello WORLD"	LOWER("Hello WORLD")	hello world
Capitalize each word	PROPER("Capitalize each word")	Capitalize Each Word
Replicates "Hello " three times	REPT("Hello ", 3)	Hello Hello Hello
Substitute "o" by "0" in "Hello World"	SUBSTITUTE("Hello World", "o", "0")	Hell0 W0rld
Extract "orl" from "Hello World"	MID("Hello World", 8, 3)	orl