

Function Name	Description
abs	Return the absolute value of the argument.
acos	The arccosine function.
add	Returns the sum of a sequence of numbers (NOT strings) plus the value of parameter 'start'. When the sequence is empty, returns start.
additive_order	Return the additive order of x.
asin	The arcsine function.
atan	The arctangent function.
binomial	Return the binomial coefficient.
ceil	The ceiling function.
combinations	A combination of a multiset (a list of objects which may contain the same object several times) mset is an unordered selection without repetitions and is represented by a sorted sublist of mset. Returns the set of all combinations of the multiset mset with k elements.
complex	Create a complex number from a real part and an optional imaginary part. This is equivalent to $(\text{real} + \text{imag} * 1j)$ where imag defaults to 0.
cos	The cosine function.
cosh	The hyperbolic cosine function.
coth	The hyperbolic cotangent function.
csch	The hyperbolic cosecant function.
denominator	Return the denominator of x.
derivative	The derivative of f.
det	Return the determinant of x.
diff	The derivative of f.
dir	Return an alphabetized list of names comprising (some of) the attributes of the given object, and of attributes reachable from it.
divisors	Returns a list of all positive integer divisors.
dumps	Dump obj to a string s. To recover obj, use loads(s).
e	The base of the natural logarithm.
eratosthenes	Return a list of the primes $\leq n$ .
exists	If S contains an element x such that P(x) is True, this function returns True and the element x. Otherwise it returns False and None.
exp	The exponential function, $\exp(x) = e^x$ .
expand	Returns the expanded form of a polynomial.
factor	Returns the factorization of the integer n as a sorted list of tuples (p,e).
factorial	Compute the factorial of n, which is the product of $1 * 2 * 3 \dots (n-1) n$ .

fibonacci	Returns then n-th Fibonacci number.
fibonacci_sequence	Returns an iterator over the Fibonacci sequence, for all fibonacci numbers $f_n$ from $n = \text{start}$ up to (but not including) $n = \text{stop}$ .
fibonacci_xrange	Returns an iterator over all of the Fibonacci numbers in the given range, including $f_n = \text{start}$ up to, but not including, $f_n = \text{stop}$ .
find_root	Numerically find a root of $f$ on the closed interval $[a,b]$ (or $[b,a]$ ) if possible, where $f$ is a function in the one variable.
floor	The floor function.
forall	If $P(x)$ is true every $x$ in $S$ , return True and None. If there is some element $x$ in $S$ such that $P$ is not True, return False and $x$ .
forget	Forget the given assumption, or call with no arguments to forget all assumptions. Here an assumption is some sort of symbolic constraint.
function	Create a formal symbolic function with the name $*s*$ .
gaussian_binomial	Return the gaussian binomial.
gcd	The greatest common divisor of $a$ and $b$ .
generic_power	The $m$ -th power of $a$ , where $m$ is a non-negative.
get_memory_usage	Return memory usage.
hex	Return the hexadecimal representation of an integer or long integer.
imag	Return the imaginary part of $x$ .
imaginary	Return the imaginary part of a complex number.
integer_ceil	Return the ceiling of $x$ .
integer_floor	Return the largest integer $\leq x$ .
integral	Return an indefinite integral of an object $x$ .
integrate	The integral of $f$ .
interval	Integers between $a$ and $b$ inclusive ( $a$ and $b$ integers).
is_AlgebraElement	Return True if $x$ is of type AlgebraElement.
is_commutative	
is_ComplexNumber	
is_even	Return whether or not an integer $x$ is even, e.g., divisible by 2.
is_Functor	
is_Infinite	
is_Integer	
is_odd	Return whether or not $x$ is odd. This is by definition the complement of <code>is_even</code> .
is_power_of_two	This function returns True if and only if $n$ is a power of 2
is_prime	Returns True if $x$ is prime, and False otherwise.
is_prime_power	Returns True if $x$ is a prime power, and False otherwise.

is_pseudoprime	Returns True if x is a pseudo-prime, and False otherwise.
is_RealNumber	Return True if x is of type RealNumber, meaning that it is an element of the MPFR real field with some precision.
is_Set	Returns true if x is a SAGE Set.
is_square	Returns whether or not n is square, and if n is a square also returns the square root. If n is not square, also returns None.
is_SymbolicExpression	
isqrt	Return an integer square root, i.e., the floor of a square root.
laplace	Attempts to compute and return the Laplace transform of self.
latex	Use latex(...) to typeset a SAGE object.
lcm	The least common multiple of a and b, or if a is a list and b is omitted the least common multiple of all elements of v.
len	Returns the number of items of a sequence or mapping.
lim	Return the limit as the variable v approaches a from the given direction.
limit	Return the limit as the variable v approaches a from the given direction.
list	list() -> new list, list(sequence) -> new list initialized from sequence's items
list_plot	list_plot takes a single list of data, in which case it forms a list of tuples (i,di) where i goes from 0 to len(data)-1 and di is the ith data value, and puts points at those tuple values. list_plot also takes a list of tuples (dxi, dyi) where dxi is the ith data representing the x-value, and dyi is the ith y-value if plotjoined=True, then a line spanning all the data is drawn instead.
load	Load SAGE object from the file with name filename, which will have an .sobj extension added if it doesn't have one. NOTE: There is also a special SAGE command (that is not available in Python) called load that you use by typing sage: load filename.sage
loads	Recover an object x that has been dumped to a string s using s = dumps(x).
log	The natural logarithm of the real number 2.
matrix	Create a matrix.
max	With a single iterable argument, return its largest item. With two or more arguments, return the largest argument.
min	With a single iterable argument, return its smallest item. With two or more arguments, return the smallest argument.
minimal_polynomial	Return the minimal polynomial of x.
mod	
mrangle	Return the multirange list with given sizes and type.
mul	Return the product of the elements in the list x.
next_prime	The next prime greater than the integer n.
next_prime_power	The next prime power greater than the integer n. If n is a prime

norm	Return the norm of x.
normalvariate	Normal distribution.
nth_prime	
number_of_arrangements	Returns the size of arrangements(mset,k).
number_of_combinations	Returns the size of combinations(mset,k).
number_of_derangements	Returns the size of derangements(mset).
number_of_divisors	Return the number of divisors of the integer n.
number_of_permutations	Returns the size of permutations(mset).
numerator	Return the numerator of x.
numerical_integral	Returns the numerical integral of the function on the interval from xmin to xmax and an error bound.
numerical_sqrt	Return a square root of x.
oct	Return the octal representation of an integer or long integer.
order	Return the order of x. If x is a ring or module element, this is the additive order of x.
parametric_plot	parametric_plot takes two functions as a list or a tuple and make a plot with the first function giving the x coordinates and the second function giving the y coordinates.
parent	Return x.parent() if defined, or type(x) if not.
permutations	A permutation is represented by a list that contains exactly the same elements as mset, but possibly in different order.
pg	Permutation groups. In SAGE a permutation is represented as either a string that defines a permutation using disjoint cycle notation, or a list of tuples, which represent disjoint cycles.
pi	The ratio of a circle's circumference to its diameter.
plot	
pow	With two arguments, equivalent to $x^y$ . With three arguments, equivalent to $(x^y) \% z$ , but may be more efficient (e.g. for longs)
power_mod	The m-th power of a modulo the integer n.
prange	List of all primes between start and stop-1, inclusive.
previous_prime	The largest prime $< n$ .
previous_prime_power	The largest prime power $< n$ .
prime_divisors	The prime divisors of the integer n, sorted in increasing order.
prime_factors	The prime divisors of the integer n, sorted in increasing order.
prime_powers	List of all positive primes powers between start and stop-1, inclusive.
primes	Returns an iterator over all primes between start and stop-1, inclusive.
primes_first_n	Return the first n primes.
prod	Return the product of the elements in the list x.
quo	Return the quotient object x/y, e.g., a quotient of numbers or of a

	polynomial ring $x$ by the ideal generated by $y$ , etc.
quotient	Return the quotient object $x/y$ , e.g., a quotient of numbers or of a polynomial ring $x$ by the ideal generated by $y$ , etc.
random	Returns a random number in the interval $[0, 1]$ .
random_prime	Returns a random prime $p$ between 2 and $n$ (i.e. $2 \leq p \leq n$ ).
randrange	Choose a random item from <code>range(start, stop[, step])</code> .
range	Returns a list containing an arithmetic progression of integers.
rational_reconstruction	This function tries to compute $x/y$ , where $x/y$ is rational number.
real	Return the real part of $x$ .
reduce	Apply a function of two arguments cumulatively to the items of a sequence, from left to right, so as to reduce the sequence to a single value.
repr	Return the canonical string representation of the object.
reset	Delete all user defined variables, reset all global variables back to their default state, and reset all interfaces to other computer algebra systems. If <code>vars</code> is specified, just restore the value of <code>vars</code> and leave all other variables alone (i.e., call <code>restore</code> ).
restore	Restore predefined global variables to their default values.
round	Round a number to a given precision in decimal digits (default 0 digits). This always returns a real double field element.
sample	Chooses $k$ unique random elements from a population sequence.
save	Save <code>obj</code> to the file with name <code>filename</code> , which will have an <code>.sobj</code> extension added if it doesn't have one. This will <i>replace</i> the contents of <code>filename</code> .
save_session	Save all variables that can be saved wto the given filename.
search	Return <code>(True,i)</code> where $i$ is such that $v[i] == x$ if there is such an $i$ , or <code>(False,j)</code> otherwise, where $j$ is the position that a should be inserted so that $v$ remains sorted.
search_doc	Full text search of the SAGE HTML documentation for lines containing <code>s</code> .
search_src	Search sage source code for lines containing <code>s</code> .
sec	The secant function.
sech	The hyperbolic secant function.
seed	
seq	A mutable list of elements with a common guaranteed universe, which can be set immutable.
set	Build an unordered collection of unique elements.
show	Show a graphics object $x$ .
show_default	Set the default for showing plots using the following commands: <code>plot</code> , <code>parametric_plot</code> , <code>polar_plot</code> , and <code>list_plot</code> .
shuffle	
sigma	Return the sum of the $k$ -th powers of the divisors of $n$ .

simplify	Simplify the expression f.
sin	The sine function.
sinh	The hyperbolic sine function.
sleep	
slice	Create a slice object. This is used for extended slicing (e.g. <code>a[0:10:2]</code> ).
slide	Use <code>latex(...)</code> to typeset a SAGE object. Use <code>%slide</code> instead to typeset slides.
solve	Algebraically solve an equation or system of equations for given variables.
sorted	
sqrt	The square root function. This is a symbolic square root.
square_free_part	Return the square free part of x, i.e., a divisor z such that $x = zy^2$ , for a perfect square $y^2$ .
srange	Return list of numbers <code>\code{a, a+step, ..., a+k*step}</code> , where $a + k*step < b$ and $a + (k+1)*step > b$ . The type of the entries in the list are the type of the starting value.
str	Return a nice string representation of the object.
subfactorial	Subfactorial or rencontres numbers, or derangements: number of permutations of $n$ elements with no fixed points.
sum	Returns the sum of a sequence of numbers (NOT strings) plus the value of parameter 'start'
super	Typically used to call a cooperative superclass method.
symbolic_expression	
sys	This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.
tan	The tangent function.
tanh	The hyperbolic tangent function.
taylor	Expands self in a truncated Taylor or Laurent series in the variable v around the point a, containing terms through $(x - a)^n$ .
transpose	
trial_division	Return the smallest prime divisor $\leq$ bound of the positive integer n, or n if there is no such prime.
two_squares	Write the integer n as a sum of two integer squares if possible; otherwise raise a ValueError.
type	Returns an object's type.
union	Return the union of x and y, as a list.
uniq	Return the sublist of all elements in the list x that is sorted and is such that the entries in the sublist are unique.
valuation	The exact power of $p > 0$ that divides the integer m.
var	Create a symbolic variable with the name *s*.

vars	Without arguments, equivalent to locals(). With an argument, equivalent to object.__dict__.
vector	Return a vector over R with given entries.
version	Return the version of SAGE.
view	Compute a latex representation of each object in objects. NOTE: In notebook mode this function simply embeds a png image in the output
walltime	Return the wall time.
xgcd	Returns triple of integers (g,s,t) such that $g = s*a+t*b = \text{gcd}(a,b)$ .
xinterval	Iterator over the integers between a and b, inclusive.
xrange	Like range(), but instead of returning a list, returns an object that generates the numbers in the range on demand.
zip	Return a list of tuples, where each tuple contains the i-th element from each of the argument sequences.

Table 3: Subset of SAGE functions

## 1079 3.22 Obtaining Information On SAGE Functions

1080 Table 3 includes a list of functions along with a short description of what each  
 1081 one does. This is not enough information, however, to show how to actually use  
 1082 these functions. One way to obtain additional information on any function is to  
 1083 type its name followed by a question mark '?' into a worksheet cell then press the  
 1084 <tab> key:

1085 is\_even?<tab>

1086 |

1087 File: /opt/sage-2.7.1-debian-32bit-i686-

1088 Linux/local/lib/python2.5/site-packages/sage/misc/functional.py

1089 Type: <type 'function'>

1090 Definition: is\_even(x)

1091 Docstring:

1092 Return whether or not an integer x is even, e.g., divisible by 2.

1093 EXAMPLES:

1094 sage: is\_even(-1)

1095 False

1096 sage: is\_even(4)

1097 True

1098 sage: is\_even(-2)

1099 True

1100 A gray window will then be shown which contains the following information  
 1101 about the function:

1102 **File:** Gives the name of the file that contains the source code that implements  
1103 the function. This is useful if you would like to locate the file to see how the  
1104 function is implemented or to edit it.

1105 **Type:** Indicates the type of the object that the name passed to the information  
1106 service refers to.

1107 **Definition:** Shows how the function is called.

1108 **Docstring:** Displays the documentation string that has been placed into the  
1109 source code of this function.

1110 You may obtain help on any of the functions listed in Table 3, or the SAGE  
1111 reference manual, using this technique. **Also, if you place two question marks**  
1112 **'??' after a function name and press the <tab> key, the function's source code**  
1113 **will be displayed.**

### 1114 **3.23 Information Is Also Available On User-Entered** 1115 **Functions**

1116 The information service can also be used to obtain information on user-entered  
1117 functions and a better understanding of how the information service works can  
1118 be gained by trying this at least once.

1119 If you have not already done so in your current worksheet, type in the addnums  
1120 function again and execute it:

```
1121 def addnums(num1, num2):  
1122     """  
1123     Returns the sum of num1 and num2.  
1124     """  
1125     answer = num1 + num2  
1126     return answer  
  
1127 #Call the function and have it add 2 to 3.  
1128 a = addnums(2, 3)  
1129 print a  
1130 |  
1131     5
```

1132 Then obtain information on this newly-entered function using the technique from  
1133 the previous section:

```
1134 addnums?<tab>  
1135 |
```