

Math	Power & Logarithmic	Angular Conversion	Constants
Number Theoretic	exp(x) log(x[, base]) log1p(x) log10(x) pow(x,y) sqrt(x)	degrees(x) radians(x)	<u>math.pi</u> The mathematical constant of pie = 3.141592.... up to the available precision <u>math.e</u> The mathematical constant e = 2.718281.... up to the available precision
ceil(x) copysign(x,y) fabs(x) factorial(x) floor(x) fmod(x,y) frexp(x) fsum(iterable) isinf(x) isnan(x) ldexp(x,i) modf() trunc()	Trigonometric Functions acos(x) asin(x) atan(x) atan2(y,x) cos(x) hypot(x,y) sin(x) tan(x)	Hyperbolic Functions acosh(x) asinh(x) atanh(x) cosh(x) sinh(x) tanh(x)	

String Formatting	Formatting Operations
'd' Signed integer decimal 'i' Signed integer decimal 'o' Signed octal value 'u' Obsolete type - it was identical to 'd' 'x' Signed hexadecimal (lowercase) 'X' Signed hexadecimal (uppercase) 'e' Floating point exponential format (lowercase) 'E' Floating point exponential format (uppercase) 'f' Floating point decimal format 'F' Floating point decimal format 'g' Floating point format. Uses the lowercase exponential format if the exponent is less than -4 or not less than precision, otherwise it uses the decimal format 'G' Floating point format. Uses the uppercase exponential format if the exponent is less than -4 or not less than precision, otherwise it uses the decimal format 'c' Single character (accepts either integer or single character string) 'r' String (converts any Python object using repr()) 's' String (converts any Python object using str()) '%' No argument is converted, adds a % character in the end result	

File	Class
Methods close() flush() fileno() isatty() next() read([size]) readline ([size]) readlines([size]) xreadlines() seek(offset[, whence]) tell() truncate([size]) write(str) writelines(sequence)	Special Methods __new__(cls) __lt__(self, other) __init__(self, args) __le__(self, other) __del__(self) __gt__(self, other) __repr__(self) __ge__(self, other) __str__(self) __eq__(self, other) __cmp__(self, other) __ne__(self, other) __index__(self) __nonzero__(self) __hash__(self) __getattr__(self, name) __getattribute__(self, name) __setattr__(self, name, attr) __delattr__(self, name) __call__(self, args, kwargs)
Attributes closed encoding errors mode name newlines softspace	

Random Functions	Array	Indexes & Slices
seed([x]) getstate() vonmisesvariate(mu,kappa) setstate(state) jumpahead(n) paretovariate(alpha) getrandbits(k) randint(a,b) weibullvariate(alpha,beta) randrange([start], stop[, step]) lognormvariate(mu,sigma) choice(seq) shuffle(x[, random]) normalvariate(mu, sigma) sample(population,k) random() gammevariate(alpha,beta) uniform(a,b) triangular(low,high,mode) gauss(mu,sigma) betavariate(alpha,beta) expovariate(lambd)	Array Methods append(x) buffer_info() byteswap() count(x) extend(iterable) fromfile(f,n) fromlist(list) fromstring(s) fromunicode(s) index(x) insert(i,x) pop([i]) remove(x) reverse() tofile(f) tolist() tostring() tounicode()	a=[0,1,2,3,4,5] b=a[:] Shallow copy of a a[1:] [1,2,3,4,5] a[5:] [0,1,2,3,4] a[-2:] [0,1,2,3] len(a) 6 a[1:3] [1,2] a[0] 0 a[1:-1] [1,2,3,4] a[5] 5 a[-1] 5 a[-2] 4

OS	SYS
OS Variables altsep Alternative separator curdir Current dir string defpath Default search path devnull Path of null device extsep Extension separator pardir Parent dir string pathsep Patch separator sep Path separator name name of OS linesep Line separator	SYS Variables argv Command line args builtin_module_names Linked C modules check_interval Signal check frequency exec_prefix Root directory executable Name of Executable exitfunc Exit function name modules Loaded modules path Search path platform Current platform stdin, stdout, stderr File objects for I/O version_info Python version info winver Version number SYS Arg V sys.argv[0] foo.py sys.argv[1] bar sys.argv[2] -c sys.argv[3] qux sys.argv[4] -h

String	String Methods	Set & Mapping	Mapping Types
	capitalize() center(width[, fillchar]) count(sub[, start[, end]]) decode encode([encoding[, errors]]) isalnum() endswith(suffix[, start[, end]]) expandtabs([tabsize]) find(sub[, start[, end]]) format(*args, **kwargs) isalpha() index(sub[, start[, end]]) isdigit() islower() isspace() istitle() isupper() join(iterable) ljust(width[, fillchar]) lower() lstrip([chars]) partition(sep) replace(old, new[, count]) rfind(sub[, start[, end]]) rindex(sub[, start[, end]]) rjust(width[, fillchar]) rpartition(sep) rsplit([sep[, maxsplit]]) rstrip([chars]) split([sep[, maxsplit]]) splitlines([keepends]) startswith(prefix[, start[, end]]) strip([chars]), swapcase, title() translate(table[, deletechars]), upper() zfill(width) isnumeric() isdecimal()	Set Types len(s) x in s x not in s isdisjoint(other) issubset(others) issuperset union(other...) intersection(other...) difference(other...) symmetric_difference(other) copy() update() intersection_update() difference_update() symmetric_difference_update() add(elem) remove() discard(elem) pop() clear()	len(d) d[key] d[key]=value del d[key] key in d key not in d iter(d) clear() copy() items() fromkeys(seq[, value]) keys() get(key[, default]) has_key(key) iteritems() iterkeys() itervalues() popitem() pop(key[, default]) setdefault(key[, default]) update([other]) values

Date Time	Date Object	Date Formatting
Time Object replace([hour[, minute[, second[, microsecond[, tzinfo]]]]) isoformat() __str__() strftime() utcoffset() dst() tzname()	replace((year,month,day)) timetuple() toordinal() weekday() isoweekday() isocalendar() isoformat() __str__() ctime() strftime()	%a Abbreviated weekday (Mon) %A Weekday (Monday) %b Abbreviated month name (Nov) %B Month name (November) %c Date and time %d Day (leading zeros) (01 to 31) %H 24 hour (leading zeros) (00 to 23) %I 12 hour (leading zeros) (01 to 12) %j Day of year (001 to 366) %m Month (01 to 12) %M Minute (00 to 59) %p AM or PM %S Second (00 to 61?) %U Week number1 (00 to 53) %w Weekday2 (0 to 6) %W Week number3 (00 to 53) %x Date %X Time %y Year without century (00 to 99) %Y Year (2016) %Z Time zone (EST) %% A literal "%" character (%)
Datetime Object date() time() timetz() toordinal() weekday() isoweekday() isocalendar() replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]) astimezone(tz) utcoffset() dst() tzname() timetuple() utctimetuple() isoformat() __str__() ctime() strftime()		