# a

| 函数 | 说明 |
|---|---|
| abs | 绝对值 |
| acos | 反余弦 |
| acosh | 反双曲余弦 |
| addcslashes | 以 C 语言风格使用反斜线转义字符串中的字符 |
| addslashes | 使用反斜线引用字符串 |
| apache_child_terminate | 在本次请求结束后终止 apache 子进程 |
| apache_getenv | 获取 Apache subprocess_env 变量 |
| apache_get_modules | 获得已加载的 Apache 模块列表 |
| apache_get_version | 获得 Apache 版本信息 |
| apache_lookup_uri | 对指定的 URI 执行部分请求并返回所有有关信息 |
| apache_note | 取得或设置 apache 请求记录 |
| apache_request_headers | 获取全部 HTTP 请求头信息 |
| apache_reset_timeout | 重置 Apache 写入计时器 |
| apache_response_headers | 获得全部 HTTP 响应头信息 |
| apache_setenv | 设置 Apache 子进程环境变量 |
| apcu_add | Cache a new variable in the data store |
| apcu_cache_info | Retrieves cached information from APCu's data store |
| apcu_cas | Updates an old value with a new value |
| apcu_clear_cache | Clears the APCu cache |
| apcu_dec | Decrease a stored number |
| apcu_delete | Removes a stored variable from the cache |
| apcu_entry | Atomically fetch or generate a cache entry |
| apcu_exists | Checks if entry exists |
| apcu_fetch | Fetch a stored variable from the cache |
| apcu_inc | Increase a stored number |
| apcu_sma_info | Retrieves APCu Shared Memory Allocation information |
| apcu_store | Cache a variable in the data store |
| apc_add | 缓存一个变量到数据存储 |
| apc_bin_dump | Get a binary dump of the given files and user variables |
| apc_bin_dumpfile | Output a binary dump of cached files and user variables to a file |
| apc_bin_load | Load a binary dump into the APC file/user cache |
| apc_bin_loadfile | Load a binary dump from a file into the APC file/user cache |
| apc_cache_info | Retrieves cached information from APC's data store |
| apc_cas | 用新值更新旧值 |
| apc_clear_cache | 清除 APC 缓存 |
| apc_compile_file | Stores a file in the bytecode cache, bypassing all filters. |
| apc_dec | Decrease a stored number |
| apc_define_constants | Defines a set of constants for retrieval and mass-definition |
| apc_delete | 从用户缓存中删除某个变量 |
| apc_delete_file | Deletes files from the opcode cache |

| | |
|---|---|
| *apc_exists* | 检查 APC 中是否存在某个或者某些 key |
| *apc_fetch* | 从缓存中取出存储的变量 |
| *apc_inc* | 递增一个储存的数字 |
| *apc_load_constants* | Loads a set of constants from the cache |
| *apc_sma_info* | Retrieves APC's Shared Memory Allocation information |
| *apc_store* | Cache a variable in the data store |
| *apd_breakpoint* | Stops the interpreter and waits on a CR from the socket |
| *apd_callstack* | Returns the current call stack as an array |
| *apd_clunk* | Throw a warning and a callstack |
| *apd_continue* | Restarts the interpreter |
| *apd_croak* | Throw an error, a callstack and then exit |
| *apd_dump_function_table* | Outputs the current function table |
| *apd_dump_persistent_resources* | Return all persistent resources as an array |
| *apd_dump_regular_resources* | Return all current regular resources as an array |
| *apd_echo* | Echo to the debugging socket |
| *apd_get_active_symbols* | Get an array of the current variables names in the local scope |
| *apd_set_pprof_trace* | Starts the session debugging |
| *apd_set_session* | Changes or sets the current debugging level |
| *apd_set_session_trace* | Starts the session debugging |
| *apd_set_session_trace_socket* | Starts the remote session debugging |
| *array* | 新建一个数组 |
| *array_change_key_case* | 返回字符串键名全为小写或大写的数组 |
| *array_chunk* | 将一个数组分割成多个 |
| *array_column* | 返回数组中指定的一列 |
| *array_combine* | 创建一个数组，用一个数组的值作为其键名，另一个数组的值作为其值 |
| *array_count_values* | 统计数组中所有的值出现的次数 |
| *array_diff* | 计算数组的差集 |
| *array_diff_assoc* | 带索引检查计算数组的差集 |
| *array_diff_key* | 使用键名比较计算数组的差集 |
| *array_diff_uassoc* | 用用户提供的回调函数做索引检查来计算数组的差集 |
| *array_diff_ukey* | 用回调函数对键名比较计算数组的差集 |
| *array_fill* | 用给定的值填充数组 |
| *array_fill_keys* | 使用指定的键和值填充数组 |
| *array_filter* | 用回调函数过滤数组中的单元 |
| *array_flip* | 交换数组中的键和值 |
| *array_intersect* | 计算数组的交集 |
| *array_intersect_assoc* | 带索引检查计算数组的交集 |
| *array_intersect_key* | 使用键名比较计算数组的交集 |
| *array_intersect_uassoc* | 带索引检查计算数组的交集，用回调函数比较索引 |
| *array_intersect_ukey* | 用回调函数比较键名来计算数组的交集 |
| *array_keys* | 返回数组中部分的或所有的键名 |
| *array_key_exists* | 检查给定的键名或索引是否存在于数组中 |
| *array_map* | 将回调函数作用到给定数组的单元上 |
| *array_merge* | 合并一个或多个数组 |

| array_merge_recursive | 递归地合并一个或多个数组 |
|---|---|
| array_multisort | 对多个数组或多维数组进行排序 |
| array_pad | 用值将数组填补到指定长度 |
| array_pop | 将数组最后一个单元弹出（出栈） |
| array_product | 计算数组中所有值的乘积 |
| array_push | 将一个或多个单元压入数组的末尾（入栈） |
| array_rand | 从数组中随机取出一个或多个单元 |
| array_reduce | 用回调函数迭代地将数组简化为单一的值 |
| array_replace | 使用传递的数组替换第一个数组的元素 |
| array_replace_recursive | 使用传递的数组递归替换第一个数组的元素 |
| array_reverse | 返回一个单元顺序相反的数组 |
| array_search | 在数组中搜索给定的值，如果成功则返回相应的键名 |
| array_shift | 将数组开头的单元移出数组 |
| array_slice | 从数组中取出一段 |
| array_splice | 把数组中的一部分去掉并用其它值取代 |
| array_sum | 计算数组中所有值的和 |
| array_udiff | 用回调函数比较数据来计算数组的差集 |
| array_udiff_assoc | 带索引检查计算数组的差集，用回调函数比较数据 |
| array_udiff_uassoc | 带索引检查计算数组的差集，用回调函数比较数据和索引 |
| array_uintersect | 计算数组的交集，用回调函数比较数据 |
| array_uintersect_assoc | 带索引检查计算数组的交集，用回调函数比较数据 |
| array_uintersect_uassoc | 带索引检查计算数组的交集，用回调函数比较数据和索引 |
| array_unique | 移除数组中重复的值 |
| array_unshift | 在数组开头插入一个或多个单元 |
| array_values | 返回数组中所有的值 |
| array_walk | 使用用户自定义函数对数组中的每个元素做回调处理 |
| array_walk_recursive | 对数组中的每个成员递归地应用用户函数 |
| arsort | 对数组进行逆向排序并保持索引关系 |
| asin | 反正弦 |
| asinh | 反双曲正弦 |
| asort | 对数组进行排序并保持索引关系 |
| assert | 检查一个断言是否为 FALSE |
| assert_options | 设置/获取断言的各种标志 |
| atan | 反正切 |
| atan2 | 两个参数的反正切 |
| atanh | 反双曲正切 |

**b**

| 函数 | 说明 |
|---|---|
| base64_decode | 对使用 MIME base64 编码的数据进行解码 |
| base64_encode | 使用 MIME base64 对数据进行编码 |
| basename | 返回路径中的文件名部分 |
| base_convert | 在任意进制之间转换数字 |

| | |
|---|---|
| *bbcode_add_element* | Adds a bbcode element |
| *bbcode_add_smiley* | Adds a smiley to the parser |
| *bbcode_create* | Create a BBCode Resource |
| *bbcode_destroy* | Close BBCode_container resource |
| *bbcode_parse* | Parse a string following a given rule set |
| *bbcode_set_arg_parser* | Attach another parser in order to use another rule set for argument parsing |
| *bbcode_set_flags* | Set or alter parser options |
| *bcadd* | 2 个任意精度数字的加法计算 |
| *bccomp* | 比较两个任意精度的数字 |
| *bcdiv* | 2 个任意精度的数字除法计算 |
| *bcmod* | 对一个任意精度数字取模 |
| *bcmul* | 2 个任意精度数字乘法计算 |
| *bcompiler_load* | 从一个 bz 压缩过的文件中读取并创建类 |
| *bcompiler_load_exe* | 从一个 bcompiler exe 文件中读取并创建类 |
| *bcompiler_parse_class* | 读取一个类的字节码并回调一个用户的函数 |
| *bcompiler_read* | 从一个文件句柄中读取并创建类 |
| *bcompiler_write_class* | 写入定义过的类的字节码 |
| *bcompiler_write_constant* | 写入定义过的常量的字节码 |
| *bcompiler_write_exe_footer* | 写入开始位置以及 exe 类型文件的结尾信号 |
| *bcompiler_write_file* | 写入 PHP 源码文件的字节码 |
| *bcompiler_write_footer* | 写入单个字符 \x00 用于标识编译数据的结尾 |
| *bcompiler_write_function* | 以字节码写入定义过的函数 |
| *bcompiler_write_functions_from_file* | 以字节码写入一个文件中定义过的所以函数 |
| *bcompiler_write_header* | 写入 bcompiler 头 |
| *bcompiler_write_included_filename* | 写入一个包含的文件的字节码 |
| *bcpow* | 任意精度数字的成方 |
| *bcpowmod* | Raise an arbitrary precision number to another, reduced by a specified modulus |
| *bcscale* | 设置所有 bc 数学函数的默认小数点保留位数 |
| *bcsqrt* | 任意精度数字的二次方根 |
| *bcsub* | 2 个任意精度数字的减法 |
| *bin2hex* | 函数把 ASCII 字符的字符串转换为十六进制值 |
| *bindec* | 二进制转换为十进制 |
| *bindtextdomain* | Sets the path for a domain |
| *bind_textdomain_codeset* | Specify the character encoding in which the messages from the DOMAIN message catalog will be returned |
| *blenc_encrypt* | Encrypt a PHP script with BLENC. |
| *boolval* | Get the boolean value of a variable |
| *bson_decode* | 反序列化一个 BSON 对象为 PHP 数组 |
| *bson_encode* | 序列化一个 PHP 变量为 BSON 字符串 |
| *bzclose* | 关闭一个 bzip2 文件 |
| *bzcompress* | 把一个字符串压缩成 bzip2 编码数据 |
| *bzdecompress* | 解压经 bzip2 编码过的数据 |

| | |
|---|---|
| *bzerrno* | 返回一个 bzip2 错误码 |
| *bzerror* | 返回包含 bzip2 错误号和错误字符串的一个 array |
| *bzerrstr* | 返回一个 bzip2 的错误字符串 |
| *bzflush* | 强制写入所有写缓冲区的数据 |
| *bzopen* | 打开一个经 bzip2 压缩过的文件 |
| *bzread* | bzip2 文件二进制安全地读取 |
| *bzwrite* | 二进制安全地写入 bzip2 文件 |

## C

| *函数* | 说明 |
|---|---|
| *cairo_create* | Returns a new CairoContext object on the requested surface. |
| *cairo_font_face_get_type* | Description |
| *cairo_font_options_create* | Description |
| *cairo_font_options_equal* | Description |
| *cairo_font_options_get_antialias* | Description |
| *cairo_font_options_get_hint_metrics* | Description |
| *cairo_font_options_get_hint_style* | Description |
| *cairo_font_options_get_subpixel_order* | Description |
| *cairo_font_options_hash* | Description |
| *cairo_font_options_merge* | Description |
| *cairo_font_options_set_antialias* | Description |
| *cairo_font_options_set_hint_metrics* | Description |
| *cairo_font_options_set_hint_style* | Description |
| *cairo_font_options_set_subpixel_order* | Description |
| *cairo_font_options_status* | Description |
| *cairo_format_stride_for_width* | Description |
| *cairo_image_surface_create* | Description |
| *cairo_image_surface_create_for_data* | Description |
| *cairo_image_surface_create_from_png* | Description |
| *cairo_image_surface_get_data* | Description |
| *cairo_image_surface_get_format* | Description |
| *cairo_image_surface_get_height* | Description |
| *cairo_image_surface_get_stride* | Description |
| *cairo_image_surface_get_width* | Description |
| *cairo_matrix_invert* | Description |
| *cairo_matrix_multiply* | Description |
| *cairo_matrix_rotate* | Description |
| *cairo_matrix_transform_distance* | Description |
| *cairo_matrix_transform_point* | Description |
| *cairo_matrix_translate* | Description |
| *cairo_pattern_add_color_stop_rgb* | Description |
| *cairo_pattern_add_color_stop_rgba* | Description |
| *cairo_pattern_create_for_surface* | Description |

| | |
|---|---|
| *cairo_pattern_create_linear* | Description |
| *cairo_pattern_create_radial* | Description |
| *cairo_pattern_create_rgb* | Description |
| *cairo_pattern_create_rgba* | Description |
| *cairo_pattern_get_color_stop_count* | Description |
| *cairo_pattern_get_color_stop_rgba* | Description |
| *cairo_pattern_get_extend* | Description |
| *cairo_pattern_get_filter* | Description |
| *cairo_pattern_get_linear_points* | Description |
| *cairo_pattern_get_matrix* | Description |
| *cairo_pattern_get_radial_circles* | Description |
| *cairo_pattern_get_rgba* | Description |
| *cairo_pattern_get_surface* | Description |
| *cairo_pattern_get_type* | Description |
| *cairo_pattern_set_extend* | Description |
| *cairo_pattern_set_filter* | Description |
| *cairo_pattern_set_matrix* | Description |
| *cairo_pattern_status* | Description |
| *cairo_pdf_surface_create* | Description |
| *cairo_pdf_surface_set_size* | Description |
| *cairo_ps_get_levels* | Description |
| *cairo_ps_level_to_string* | Description |
| *cairo_ps_surface_create* | Description |
| *cairo_ps_surface_dsc_begin_page_setup* | Description |
| *cairo_ps_surface_dsc_begin_setup* | Description |
| *cairo_ps_surface_dsc_comment* | Description |
| *cairo_ps_surface_get_eps* | Description |
| *cairo_ps_surface_restrict_to_level* | Description |
| *cairo_ps_surface_set_eps* | Description |
| *cairo_ps_surface_set_size* | Description |
| *cairo_scaled_font_create* | Description |
| *cairo_scaled_font_extents* | Description |
| *cairo_scaled_font_get_ctm* | Description |
| *cairo_scaled_font_get_font_face* | Description |
| *cairo_scaled_font_get_font_matrix* | Description |
| *cairo_scaled_font_get_font_options* | Description |
| *cairo_scaled_font_get_scale_matrix* | Description |
| *cairo_scaled_font_get_type* | Description |
| *cairo_scaled_font_glyph_extents* | Description |
| *cairo_scaled_font_status* | Description |
| *cairo_scaled_font_text_extents* | Description |
| *cairo_surface_copy_page* | Description |
| *cairo_surface_create_similar* | Description |
| *cairo_surface_finish* | Description |

| | |
|---|---|
| *cairo_surface_flush* | Description |
| *cairo_surface_get_content* | Description |
| *cairo_surface_get_device_offset* | Description |
| *cairo_surface_get_font_options* | Description |
| *cairo_surface_get_type* | Description |
| *cairo_surface_mark_dirty* | Description |
| *cairo_surface_mark_dirty_rectangle* | Description |
| *cairo_surface_set_device_offset* | Description |
| *cairo_surface_set_fallback_resolution* | Description |
| *cairo_surface_show_page* | Description |
| *cairo_surface_status* | Description |
| *cairo_surface_write_to_png* | Description |
| *cairo_svg_surface_create* | Description |
| *cairo_svg_surface_restrict_to_version* | Description |
| *cairo_svg_version_to_string* | Description |
| *calculhmac* | Obtain a hmac key (needs 2 arguments) |
| *calcul_hmac* | Obtain a hmac key (needs 8 arguments) |
| *call_user_func* | 把第一个参数作为回调函数调用 |
| *call_user_func_array* | 调用回调函数，并把一个数组参数作为回调函数的参数 |
| *call_user_method* | 对特定对象调用用户方法（已废弃） |
| *call_user_method_array* | 调用一个用户方法，同时传递参数数组（已废弃） |
| *cal_days_in_month* | 返回某个历法中某年中某月的天数 |
| *cal_from_jd* | 转换 Julian Day 计数到一个支持的历法。 |
| *cal_info* | 返回选定历法的信息 |
| *cal_to_jd* | 从一个支持的历法转变为 Julian Day 计数。 |
| *ceil* | 进一法取整 |
| *chdb_create* | Creates a chdb file |
| *chdir* | 改变目录 |
| *checkdate* | 验证一个格里高里日期 |
| *checkdnsrr* | 给指定的主机（域名）或者 IP 地址做 DNS 通信检查 |
| *chgrp* | 改变文件所属的组 |
| *chmod* | 改变文件模式 |
| *chop* | rtrim 的别名 |
| *chown* | 改变文件的所有者 |
| *chr* | 返回指定的字符 |
| *chroot* | 改变根目录 |
| *chunk_split* | 将字符串分割成小块 |
| *classkit_import* | Import new class method definitions from a file |
| *classkit_method_add* | Dynamically adds a new method to a given class |
| *classkit_method_copy* | Copies a method from class to another |
| *classkit_method_redefine* | Dynamically changes the code of the given method |
| *classkit_method_remove* | Dynamically removes the given method |
| *classkit_method_rename* | Dynamically changes the name of the given method |
| *class_alias* | 为一个类创建别名 |

| | |
|---|---|
| *class_exists* | 检查类是否已定义 |
| *class_implements* | 返回指定的类实现的所有接口。 |
| *class_parents* | 返回指定类的父类。 |
| *class_uses* | Return the traits used by the given class |
| *clearstatcache* | 清除文件状态缓存 |
| *cli_get_process_title* | Returns the current process title |
| *cli_set_process_title* | Sets the process title |
| *closedir* | 关闭目录句柄 |
| *closelog* | 关闭系统日志链接 |
| *compact* | 建立一个数组，包括变量名和它们的值 |
| *com_create_guid* | Generate a globally unique identifier (GUID) |
| *com_event_sink* | Connect events from a COM object to a PHP object |
| *com_get_active_object* | Returns a handle to an already running instance of a COM object |
| *com_load_typelib* | 装载一个 Typelib |
| *com_message_pump* | Process COM messages, sleeping for up to timeoutms milliseconds |
| *com_print_typeinfo* | Print out a PHP class definition for a dispatchable interface |
| *connection_aborted* | 检查客户端是否已经断开 |
| *connection_status* | 返回连接的状态位 |
| *constant* | 返回一个常量的值 |
| *Constants for PDO_4D* | Constants for PDO_4D |
| *Context 参数* | Context 参数列表 |
| *convert_cyr_string* | 将字符由一种 Cyrillic 字符转换成另一种 |
| *convert_uudecode* | 解码一个 uuencode 编码的字符串 |
| *convert_uuencode* | 使用 uuencode 编码一个字符串 |
| *copy* | 拷贝文件 |
| *cos* | 余弦 |
| *cosh* | 双曲余弦 |
| *count* | 计算数组中的单元数目或对象中的属性个数 |
| *counter_bump* | 修改简单计数器的当前值。 |
| *counter_bump_value* | 更新计数器资源的当前值。 |
| *counter_create* | 创建一个包含单个数值的计数器。 |
| *counter_get* | 获取简单计数器的当前值。 |
| *counter_get_meta* | 返回计数器资源的部分元信息。 |
| *counter_get_named* | 按名称查询一个已存在的计数器，并作为资源返回。 |
| *counter_get_value* | 获取计数器资源的当前值。 |
| *counter_reset* | 重置简单计数器的当前值。 |
| *counter_reset_value* | 重置计数器资源的当前值。 |
| *count_chars* | 返回字符串所用字符的信息 |
| *crack_check* | Performs an obscure check with the given password |
| *crack_closedict* | Closes an open CrackLib dictionary |
| *crack_getlastmessage* | Returns the message from the last obscure check |
| *crack_opendict* | Opens a new CrackLib dictionary |
| *crc32* | 计算一个字符串的 crc32 多项式 |
| *create_function* | Create an anonymous (lambda-style) function |

| | |
|---|---|
| *crypt* | 单向字符串散列 |
| *ctype_alnum* | 做字母和数字字符检测 |
| *ctype_alpha* | 做纯字符检测 |
| *ctype_cntrl* | 做控制字符检测 |
| *ctype_digit* | 做纯数字检测 |
| *ctype_graph* | 做可打印字符串检测，空格除外 |
| *ctype_lower* | 做小写字符检测 |
| *ctype_print* | 做可打印字符检测 |
| *ctype_punct* | 检测可打印的字符是不是不包含空白、数字和字母 |
| *ctype_space* | 做空白字符检测 |
| *ctype_upper* | 做大写字母检测 |
| *ctype_xdigit* | 检测字符串是否只包含十六进制字符 |
| *cubrid_affected_rows* | Return the number of rows affected by the last SQL statement |
| *cubrid_bind* | Bind variables to a prepared statement as parameters |
| *cubrid_client_encoding* | Return the current CUBRID connection charset |
| *cubrid_close* | Close CUBRID connection |
| *cubrid_close_prepare* | Close the request handle |
| *cubrid_close_request* | Close the request handle |
| *cubrid_column_names* | Get the column names in result |
| *cubrid_column_types* | Get column types in result |
| *cubrid_col_get* | Get contents of collection type column using OID |
| *cubrid_col_size* | Get the number of elements in collection type column using OID |
| *cubrid_commit* | Commit a transaction |
| *cubrid_connect* | Open a connection to a CUBRID Server |
| *cubrid_connect_with_url* | Establish the environment for connecting to CUBRID server |
| *cubrid_current_oid* | Get OID of the current cursor location |
| *cubrid_data_seek* | Move the internal row pointer of the CUBRID result |
| *cubrid_db_name* | Get db name from results of cubrid_list_dbs |
| *cubrid_disconnect* | Close a database connection |
| *cubrid_drop* | Delete an instance using OID |
| *cubrid_errno* | Return the numerical value of the error message from previous CUBRID operation |
| *cubrid_error* | Get the error message |
| *cubrid_error_code* | Get error code for the most recent function call |
| *cubrid_error_code_facility* | Get the facility code of error |
| *cubrid_error_msg* | Get last error message for the most recent function call |
| *cubrid_execute* | Execute a prepared SQL statement |
| *cubrid_fetch* | Fetch the next row from a result set |
| *cubrid_fetch_array* | Fetch a result row as an associative array, a numeric array, or both |
| *cubrid_fetch_assoc* | Return the associative array that corresponds to the fetched row |
| *cubrid_fetch_field* | Get column information from a result and return as an object |
| *cubrid_fetch_lengths* | Return an array with the lengths of the values of each field from the current row |

| | |
|---|---|
| *cubrid_fetch_object* | Fetche the next row and returns it as an object |
| *cubrid_fetch_row* | Return a numerical array with the values of the current row |
| *cubrid_field_flags* | Return a string with the flags of the given field offset |
| *cubrid_field_len* | Get the maximum length of the specified field |
| *cubrid_field_name* | Return the name of the specified field index |
| *cubrid_field_seek* | Move the result set cursor to the specified field offset |
| *cubrid_field_table* | Return the name of the table of the specified field |
| *cubrid_field_type* | Return the type of the column corresponding to the given field offset |
| *cubrid_free_result* | Free the memory occupied by the result data |
| *cubrid_get* | Get a column using OID |
| *cubrid_get_autocommit* | Get auto-commit mode of the connection |
| *cubrid_get_charset* | Return the current CUBRID connection charset |
| *cubrid_get_class_name* | Get the class name using OID |
| *cubrid_get_client_info* | Return the client library version |
| *cubrid_get_db_parameter* | Returns the CUBRID database parameters |
| *cubrid_get_query_timeout* | Get the query timeout value of the request |
| *cubrid_get_server_info* | Return the CUBRID server version |
| *cubrid_insert_id* | Return the ID generated for the last updated AUTO_INCREMENT column |
| *cubrid_is_instance* | Check whether the instance pointed by OID exists |
| *cubrid_list_dbs* | Return an array with the list of all existing CUBRID databases |
| *cubrid_load_from_glo* | Read data from a GLO instance and save it in a file |
| *cubrid_lob2_bind* | Bind a lob object or a string as a lob object to a prepared statement as parameters. |
| *cubrid_lob2_close* | Close LOB object. |
| *cubrid_lob2_export* | Export the lob object to a file. |
| *cubrid_lob2_import* | Import BLOB/CLOB data from a file. |
| *cubrid_lob2_new* | Create a lob object. |
| *cubrid_lob2_read* | Read from BLOB/CLOB data. |
| *cubrid_lob2_seek* | Move the cursor of a lob object. |
| *cubrid_lob2_seek64* | Move the cursor of a lob object. |
| *cubrid_lob2_size* | Get a lob object's size. |
| *cubrid_lob2_size64* | Get a lob object's size. |
| *cubrid_lob2_tell* | Tell the cursor position of the LOB object. |
| *cubrid_lob2_tell64* | Tell the cursor position of the LOB object. |
| *cubrid_lob2_write* | Write to a lob object. |
| *cubrid_lob_close* | Close BLOB/CLOB data |
| *cubrid_lob_export* | Export BLOB/CLOB data to file |
| *cubrid_lob_get* | Get BLOB/CLOB data |
| *cubrid_lob_send* | Read BLOB/CLOB data and send straight to browser |
| *cubrid_lob_size* | Get BLOB/CLOB data size |
| *cubrid_lock_read* | Set a read lock on the given OID |
| *cubrid_lock_write* | Set a write lock on the given OID |
| *cubrid_move_cursor* | Move the cursor in the result |

| | |
|---|---|
| *cubrid_new_glo* | Create a glo instance |
| *cubrid_next_result* | Get result of next query when executing multiple SQL statements |
| *cubrid_num_cols* | Return the number of columns in the result set |
| *cubrid_num_fields* | Return the number of columns in the result set |
| *cubrid_num_rows* | Get the number of rows in the result set |
| *cubrid_pconnect* | Open a persistent connection to a CUBRID server |
| *cubrid_pconnect_with_url* | Open a persistent connection to CUBRID server |
| *cubrid_ping* | Ping a server connection or reconnect if there is no connection |
| *cubrid_prepare* | Prepare a SQL statement for execution |
| *cubrid_put* | Update a column using OID |
| *cubrid_query* | Send a CUBRID query |
| *cubrid_real_escape_string* | Escape special characters in a string for use in an SQL statement |
| *cubrid_result* | Return the value of a specific field in a specific row |
| *cubrid_rollback* | Roll back a transaction |
| *cubrid_save_to_glo* | Save requested file in a GLO instance |
| *cubrid_schema* | Get the requested schema information |
| *cubrid_send_glo* | Read data from glo and send it to std output |
| *cubrid_seq_drop* | Delete an element from sequence type column using OID |
| *cubrid_seq_insert* | Insert an element to a sequence type column using OID |
| *cubrid_seq_put* | Update the element value of sequence type column using OID |
| *cubrid_set_add* | Insert a single element to set type column using OID |
| *cubrid_set_autocommit* | Set autocommit mode of the connection |
| *cubrid_set_db_parameter* | Sets the CUBRID database parameters |
| *cubrid_set_drop* | Delete an element from set type column using OID |
| *cubrid_set_query_timeout* | Set the timeout time of query execution |
| *cubrid_unbuffered_query* | Perform a query without fetching the results into memory |
| *cubrid_version* | Get the CUBRID PHP module's version |
| *CURL context options* | CURL 上下文选项列表 |
| *curl_close* | 关闭一个 cURL 会话 |
| *curl_copy_handle* | 复制一个 cURL 句柄和它的所有选项 |
| *curl_errno* | 返回最后一次的错误号 |
| *curl_error* | 返回一个保护当前会话最近一次错误的字符串 |
| *curl_escape* | 使用 URL 编码给定的字符串 |
| *curl_exec* | 执行一个 cURL 会话 |
| *curl_file_create* | 创建一个 CURLFile 对象 |
| *curl_getinfo* | 获取一个 cURL 连接资源句柄的信息 |
| *curl_init* | 初始化一个 cURL 会话 |
| *curl_multi_add_handle* | 向 curl 批处理会话中添加单独的 curl 句柄 |
| *curl_multi_close* | 关闭一组 cURL 句柄 |
| *curl_multi_exec* | 运行当前 cURL 句柄的子连接 |
| *curl_multi_getcontent* | 如果设置了 CURLOPT_RETURNTRANSFER，则返回获取的输出的文本流 |
| *curl_multi_info_read* | 获取当前解析的 cURL 的相关传输信息 |
| *curl_multi_init* | 返回一个新 cURL 批处理句柄 |
| *curl_multi_remove_handle* | 移除 curl 批处理句柄资源中的某个句柄资源 |

| | |
|---|---|
| *curl_multi_select* | 等待所有 cURL 批处理中的活动连接 |
| *curl_multi_setopt* | 为 cURL 并行处理设置一个选项 |
| *curl_multi_strerror* | Return string describing error code |
| *curl_pause* | Pause and unpause a connection |
| *curl_reset* | Reset all options of a libcurl session handle |
| *curl_setopt* | 设置一个 cURL 传输选项 |
| *curl_setopt_array* | 为 cURL 传输会话批量设置选项 |
| *curl_share_close* | Close a cURL share handle |
| *curl_share_init* | Initialize a cURL share handle |
| *curl_share_setopt* | Set an option for a cURL share handle. |
| *curl_strerror* | Return string describing the given error code |
| *curl_unescape* | 解码给定的 URL 编码的字符串 |
| *curl_version* | 获取 cURL 版本信息 |
| *current* | 返回数组中的当前单元 |
| *cyrus_authenticate* | Authenticate against a Cyrus IMAP server |
| *cyrus_bind* | Bind callbacks to a Cyrus IMAP connection |
| *cyrus_close* | Close connection to a Cyrus IMAP server |
| *cyrus_connect* | Connect to a Cyrus IMAP server |
| *cyrus_query* | Send a query to a Cyrus IMAP server |
| | |
| *cyrus_unbind* | Unbind … |

# d

| *函数* | 说明 |
|---|---|
| *date* | 格式化一个本地时间／日期 |
| *date_default_timezone_get* | 取得一个脚本中所有日期时间函数所使用的默认时区 |
| *date_default_timezone_set* | 设定用于一个脚本中所有日期时间函数的默认时区 |
| *date_parse* | Returns associative array with detailed info about given date |
| *date_parse_from_format* | Get info about given date formatted according to the specified format |
| *date_sunrise* | 返回给定的日期与地点的日出时间 |
| *date_sunset* | 返回给定的日期与地点的日落时间 |
| *date_sun_info* | Returns an array with information about sunset/sunrise and twilight begin/end |
| *db2_autocommit* | Returns or sets the AUTOCOMMIT state for a database connection |
| *db2_bind_param* | Binds a PHP variable to an SQL statement parameter |
| *db2_client_info* | Returns an object with properties that describe the DB2 database client |
| *db2_close* | Closes a database connection |
| *db2_columns* | Returns a result set listing the columns and associated metadata for a table |
| *db2_column_privileges* | Returns a result set listing the columns and associated privileges for a table |

| | |
|---|---|
| *db2_commit* | Commits a transaction |
| *db2_connect* | Returns a connection to a database |
| *db2_conn_error* | Returns a string containing the SQLSTATE returned by the last connection attempt |
| *db2_conn_errormsg* | Returns the last connection error message and SQLCODE value |
| *db2_cursor_type* | Returns the cursor type used by a statement resource |
| *db2_escape_string* | Used to escape certain characters |
| *db2_exec* | Executes an SQL statement directly |
| *db2_execute* | Executes a prepared SQL statement |
| *db2_fetch_array* | Returns an array, indexed by column position, representing a row in a result set |
| *db2_fetch_assoc* | Returns an array, indexed by column name, representing a row in a result set |
| *db2_fetch_both* | Returns an array, indexed by both column name and position, representing a row in a result set |
| *db2_fetch_object* | Returns an object with properties representing columns in the fetched row |
| *db2_fetch_row* | Sets the result set pointer to the next row or requested row |
| *db2_field_display_size* | Returns the maximum number of bytes required to display a column |
| *db2_field_name* | Returns the name of the column in the result set |
| *db2_field_num* | Returns the position of the named column in a result set |
| *db2_field_precision* | Returns the precision of the indicated column in a result set |
| *db2_field_scale* | Returns the scale of the indicated column in a result set |
| *db2_field_type* | Returns the data type of the indicated column in a result set |
| *db2_field_width* | Returns the width of the current value of the indicated column in a result set |
| *db2_foreign_keys* | Returns a result set listing the foreign keys for a table |
| *db2_free_result* | Frees resources associated with a result set |
| *db2_free_stmt* | Frees resources associated with the indicated statement resource |
| *db2_get_option* | Retrieves an option value for a statement resource or a connection resource |
| *db2_last_insert_id* | Returns the auto generated ID of the last insert query that successfully executed on this connection |
| *db2_lob_read* | Gets a user defined size of LOB files with each invocation |
| *db2_next_result* | Requests the next result set from a stored procedure |
| *db2_num_fields* | Returns the number of fields contained in a result set |
| *db2_num_rows* | Returns the number of rows affected by an SQL statement |
| *db2_pclose* | Closes a persistent database connection |
| *db2_pconnect* | Returns a persistent connection to a database |
| *db2_prepare* | Prepares an SQL statement to be executed |
| *db2_primary_keys* | Returns a result set listing primary keys for a table |
| *db2_procedures* | Returns a result set listing the stored procedures registered in a database |
| *db2_procedure_columns* | Returns a result set listing stored procedure parameters |
| *db2_result* | Returns a single column from a row in the result set |

| | |
|---|---|
| db2_rollback | Rolls back a transaction |
| db2_server_info | Returns an object with properties that describe the DB2 database server |
| db2_set_option | Set options for connection or statement resources |
| db2_special_columns | Returns a result set listing the unique row identifier columns for a table |
| db2_statistics | Returns a result set listing the index and statistics for a table |
| db2_stmt_error | Returns a string containing the SQLSTATE returned by an SQL statement |
| db2_stmt_errormsg | Returns a string containing the last SQL statement error message |
| db2_tables | Returns a result set listing the tables and associated metadata in a database |
| db2_table_privileges | Returns a result set listing the tables and associated privileges in a database |
| dbase_add_record | Adds a record to a database |
| dbase_close | Closes a database |
| dbase_create | Creates a database |
| dbase_delete_record | Deletes a record from a database |
| dbase_get_header_info | Gets the header info of a database |
| dbase_get_record | Gets a record from a database as an indexed array |
| dbase_get_record_with_names | Gets a record from a database as an associative array |
| dbase_numfields | Gets the number of fields of a database |
| dbase_numrecords | Gets the number of records in a database |
| dbase_open | Opens a database |
| dbase_pack | Packs a database |
| dbase_replace_record | Replaces a record in a database |
| dba_close | Close a DBA database |
| dba_delete | Delete DBA entry specified by key |
| dba_exists | Check whether key exists |
| dba_fetch | Fetch data specified by key |
| dba_firstkey | Fetch first key |
| dba_handlers | List all the handlers available |
| dba_insert | Insert entry |
| dba_key_split | Splits a key in string representation into array representation |
| dba_list | List all open database files |
| dba_nextkey | Fetch next key |
| dba_open | Open database |
| dba_optimize | Optimize database |
| dba_popen | Open database persistently |
| dba_replace | Replace or insert entry |
| dba_sync | Synchronize database |
| dbplus_add | Add a tuple to a relation |
| dbplus_aql | Perform AQL query |
| dbplus_chdir | Get/Set database virtual current directory |
| dbplus_close | Close a relation |

| | |
|---|---|
| *dbplus_curr* | Get current tuple from relation |
| *dbplus_errcode* | Get error string for given errorcode or last error |
| *dbplus_errno* | Get error code for last operation |
| *dbplus_find* | Set a constraint on a relation |
| *dbplus_first* | Get first tuple from relation |
| *dbplus_flush* | Flush all changes made on a relation |
| *dbplus_freealllocks* | Free all locks held by this client |
| *dbplus_freelock* | Release write lock on tuple |
| *dbplus_freerlocks* | Free all tuple locks on given relation |
| *dbplus_getlock* | Get a write lock on a tuple |
| *dbplus_getunique* | Get an id number unique to a relation |
| *dbplus_info* | Get information about a relation |
| *dbplus_last* | Get last tuple from relation |
| *dbplus_lockrel* | Request write lock on relation |
| *dbplus_next* | Get next tuple from relation |
| *dbplus_open* | Open relation file |
| *dbplus_prev* | Get previous tuple from relation |
| *dbplus_rchperm* | Change relation permissions |
| *dbplus_rcreate* | Creates a new DB++ relation |
| *dbplus_rcrtexact* | Creates an exact but empty copy of a relation including indices |
| *dbplus_rcrtlike* | Creates an empty copy of a relation with default indices |
| *dbplus_resolve* | Resolve host information for relation |
| *dbplus_restorepos* | Restore position |
| *dbplus_rkeys* | Specify new primary key for a relation |
| *dbplus_ropen* | Open relation file local |
| *dbplus_rquery* | Perform local (raw) AQL query |
| *dbplus_rrename* | Rename a relation |
| *dbplus_rsecindex* | Create a new secondary index for a relation |
| *dbplus_runlink* | Remove relation from filesystem |
| *dbplus_rzap* | Remove all tuples from relation |
| *dbplus_savepos* | Save position |
| *dbplus_setindex* | Set index |
| *dbplus_setindexbynumber* | Set index by number |
| *dbplus_sql* | Perform SQL query |
| *dbplus_tcl* | Execute TCL code on server side |
| *dbplus_tremove* | Remove tuple and return new current tuple |
| *dbplus_undo* | Undo |
| *dbplus_undoprepare* | Prepare undo |
| *dbplus_unlockrel* | Give up write lock on relation |
| *dbplus_unselect* | Remove a constraint from relation |
| *dbplus_update* | Update specified tuple in relation |
| *dbplus_xlockrel* | Request exclusive lock on relation |
| *dbplus_xunlockrel* | Free exclusive lock on relation |
| *dbx_close* | Close an open connection/database |

| | |
|---|---|
| *dbx_compare* | Compare two rows for sorting purposes |
| *dbx_connect* | Open a connection/database |
| *dbx_error* | Report the error message of the latest function call in the module |
| *dbx_escape_string* | Escape a string so it can safely be used in an sql-statement |
| *dbx_fetch_row* | Fetches rows from a query-result that had the DBX_RESULT_UNBUFFERED flag set |
| *dbx_query* | Send a query and fetch all results (if any) |
| *dbx_sort* | Sort a result from a dbx_query by a custom sort function |
| *dcgettext* | Overrides the domain for a single lookup |
| *dcngettext* | Plural version of dcgettext |
| *debug_backtrace* | 产生一条回溯跟踪(backtrace) |
| *debug_print_backtrace* | 打印一条回溯。 |
| *debug_zval_dump* | Dumps a string representation of an internal zend value to output |
| *decbin* | 十进制转换为二进制 |
| *dechex* | 十进制转换为十六进制 |
| *decoct* | 十进制转换为八进制 |
| *define* | 定义一个常量 |
| *defined* | 检查某个名称的常量是否存在 |
| *define_syslog_variables* | Initializes all syslog related variables |
| *deg2rad* | 将角度转换为弧度 |
| *delete* | 参见 unlink 或 unset |
| *dgettext* | Override the current domain |
| *die* | 等同于 exit |
| *dio_close* | Closes the file descriptor given by fd |
| *dio_fcntl* | Performs a c library fcntl on fd |
| *dio_open* | Opens a file (creating it if necessary) at a lower level than the C library input/ouput stream functions allow. |
| *dio_read* | Reads bytes from a file descriptor |
| *dio_seek* | Seeks to pos on fd from whence |
| *dio_stat* | Gets stat information about the file descriptor fd |
| *dio_tcsetattr* | Sets terminal attributes and baud rate for a serial port |
| *dio_truncate* | Truncates file descriptor fd to offset bytes |
| *dio_write* | Writes data to fd with optional truncation at length |
| *dir* | 返回一个 Directory 类实例 |
| *dirname* | 返回路径中的目录部分 |
| *diskfreespace* | disk_free_space 的别名 |
| *disk_free_space* | 返回目录中的可用空间 |
| *disk_total_space* | 返回一个目录的磁盘总大小 |
| *dl* | 运行时载入一个 PHP 扩展 |
| *dngettext* | Plural version of dgettext |
| *dns_check_record* | 别名 checkdnsrr |
| *dns_get_mx* | 别名 getmxrr |
| *dns_get_record* | 获取指定主机的 DNS 记录 |
| *dom_import_simplexml* | Gets a DOMElement object from a SimpleXMLElement object |

| | |
|---|---|
| *doubleval* | floatval 的别名 |

# e

| *函数* | 说明 |
|---|---|
| *each* | 返回数组中当前的键 / 值对并将数组指针向前移动一步 |
| *easter_date* | 得到指定年份的复活节午夜时的 Unix 时间戳。 |
| *easter_days* | 得到指定年份的 3 月 21 日到复活节之间的天数 |
| *echo* | 输出一个或多个字符串 |
| *eio_busy* | Artificially increase load. Could be useful in tests, benchmarking. |
| *eio_cancel* | Cancels a request |
| *eio_chmod* | Change file/direcrory permissions. |
| *eio_chown* | Change file/direcrory permissions. |
| *eio_close* | Close file |
| *eio_custom* | Execute custom request like any other eio_* call. |
| *eio_dup2* | Duplicate a file descriptor |
| *eio_event_loop* | Polls libeio until all requests proceeded |
| *eio_fallocate* | Allows the caller to directly manipulate the allocated disk space for a file |
| *eio_fchmod* | Change file permissions. |
| *eio_fchown* | Change file ownership |
| *eio_fdatasync* | Synchronize a file's in-core state with storage device. |
| *eio_fstat* | Get file status |
| *eio_fstatvfs* | Get file system statistics |
| *eio_fsync* | Synchronize a file's in-core state with storage device |
| *eio_ftruncate* | Truncate a file |
| *eio_futime* | Change file last access and modification times |
| *eio_get_event_stream* | Get stream representing a variable used in internal communications with libeio. |
| *eio_get_last_error* | Returns string describing the last error associated with a request resource |
| *eio_grp* | Createsa request group. |
| *eio_grp_add* | Adds a request to the request group. |
| *eio_grp_cancel* | Cancels a request group |
| *eio_grp_limit* | Set group limit |
| *eio_init* | (Re-)initialize Eio |
| *eio_link* | Create a hardlink for file |
| *eio_lstat* | Get file status |
| *eio_mkdir* | Create directory |
| *eio_mknod* | Create a special or ordinary file. |
| *eio_nop* | Does nothing, except go through the whole request cycle. |
| *eio_npending* | Returns number of finished, but unhandled requests |

| | |
|---|---|
| *eio_nready* | Returns number of not-yet handled requests |
| *eio_nreqs* | Returns number of requests to be processed |
| *eio_nthreads* | Returns number of threads currently in use |
| *eio_open* | Opens a file |
| *eio_poll* | Can be to be called whenever there are pending requests that need finishing. |
| *eio_read* | Read from a file descriptor at given offset. |
| *eio_readahead* | Perform file readahead into page cache |
| *eio_readdir* | Reads through a whole directory |
| *eio_readlink* | Read value of a symbolic link. |
| *eio_realpath* | Get the canonicalized absolute pathname. |
| *eio_rename* | Change the name or location of a file. |
| *eio_rmdir* | Remove a directory |
| *eio_seek* | Repositions the offset of the open file associated with the fd argument to the argument offset according to the directive whence |
| *eio_sendfile* | Transfer data between file descriptors |
| *eio_set_max_idle* | Set maximum number of idle threads. |
| *eio_set_max_parallel* | Set maximum parallel threads |
| *eio_set_max_poll_reqs* | Set maximum number of requests processed in a poll. |
| *eio_set_max_poll_time* | Set maximum poll time |
| *eio_set_min_parallel* | Set minimum parallel thread number |
| *eio_stat* | Get file status |
| *eio_statvfs* | Get file system statistics |
| *eio_symlink* | Create a symbolic link |
| *eio_sync* | Commit buffer cache to disk |
| *eio_syncfs* | Calls Linux' syncfs syscall, if available |
| *eio_sync_file_range* | Sync a file segment with disk |
| *eio_truncate* | Truncate a file |
| *eio_unlink* | Delete a name and possibly the file it refers to |
| *eio_utime* | Change file last access and modification times. |
| *eio_write* | Write to file |
| *empty* | 检查一个变量是否为空 |
| *enchant_broker_describe* | Enumerates the Enchant providers |
| *enchant_broker_dict_exists* | Whether a dictionary exists or not. Using non-empty tag |
| *enchant_broker_free* | Free the broker resource and its dictionnaries |
| *enchant_broker_free_dict* | Free a dictionary resource |
| *enchant_broker_get_dict_path* | Get the directory path for a given backend |
| *enchant_broker_get_error* | Returns the last error of the broker |
| *enchant_broker_init* | create a new broker object capable of requesting |
| *enchant_broker_list_dicts* | Returns a list of available dictionaries |
| *enchant_broker_request_dict* | create a new dictionary using a tag |
| *enchant_broker_request_pwl_dict* | creates a dictionary using a PWL file |
| *enchant_broker_set_dict_path* | Set the directory path for a given backend |
| *enchant_broker_set_ordering* | Declares a preference of dictionaries to use for the language |

| | |
|---|---|
| *enchant_dict_add_to_personal* | add a word to personal word list |
| *enchant_dict_add_to_session* | add 'word' to this spell-checking session |
| *enchant_dict_check* | Check whether a word is correctly spelled or not |
| *enchant_dict_describe* | Describes an individual dictionary |
| *enchant_dict_get_error* | Returns the last error of the current spelling-session |
| *enchant_dict_is_in_session* | whether or not 'word' exists in this spelling-session |
| *enchant_dict_quick_check* | Check the word is correctly spelled and provide suggestions |
| *enchant_dict_store_replacement* | Add a correction for a word |
| *enchant_dict_suggest* | Will return a list of values if any of those pre-conditions are not met |
| *end* | 将数组的内部指针指向最后一个单元 |
| *ereg* | 正则表达式匹配 |
| *eregi* | 不区分大小写的正则表达式匹配 |
| *eregi_replace* | 不区分大小写的正则表达式替换 |
| *ereg_replace* | 正则表达式替换 |
| *error_clear_last* | Clear the most recent error |
| *error_get_last* | 获取最后发生的错误 |
| *error_log* | 发送错误信息到某个地方 |
| *error_reporting* | 设置应该报告何种 PHP 错误 |
| *escapeshellarg* | 把字符串转码为可以在 shell 命令里使用的参数 |
| *escapeshellcmd* | shell 元字符转义 |
| *eval* | 把字符串作为 PHP 代码执行 |
| *event_base_free* | Destroy event base |
| *event_base_loop* | Handle events |
| *event_base_loopbreak* | Abort event loop |
| *event_base_loopexit* | Exit loop after a time |
| *event_base_new* | Create and initialize new event base |
| *event_base_priority_init* | Set the number of event priority levels |
| *event_base_reinit* | Reinitialize the event base after a fork |
| *event_base_set* | Associate event base with an event |
| *event_buffer_base_set* | Associate buffered event with an event base |
| *event_buffer_disable* | Disable a buffered event |
| *event_buffer_enable* | Enable a buffered event |
| *event_buffer_fd_set* | Change a buffered event file descriptor |
| *event_buffer_free* | Destroy buffered event |
| *event_buffer_new* | Create new buffered event |
| *event_buffer_priority_set* | Assign a priority to a buffered event |
| *event_buffer_read* | Read data from a buffered event |
| *event_buffer_set_callback* | Set or reset callbacks for a buffered event |
| *event_buffer_timeout_set* | Set read and write timeouts for a buffered event |
| *event_buffer_watermark_set* | Set the watermarks for read and write events |
| *event_buffer_write* | Write data to a buffered event |
| *event_new* | Create new event |
| *event_priority_set* | Assign a priority to an event. |

| event_timer_add | 别名 event_add |
|---|---|
| event_timer_del | 别名 event_del |
| event_timer_new | 别名 event_new |
| event_timer_set | Prepare a timer event |
| Examples with PDO_4D | Examples PDO_4D |
| exec | 执行一个外部程序 |
| exif_imagetype | 判断一个图像的类型 |
| exif_read_data | 从 JPEG 或 TIFF 文件中读取 EXIF 头信息 |
| exif_tagname | 获取指定索引的头名称 |
| exif_thumbnail | 取得嵌入在 TIFF 或 JPEG 图像中的缩略图 |
| exit | 输出一个消息并且退出当前脚本 |
| exp | 计算 e 的指数 |
| expect_expectl | Waits until the output from a process matches one of the patterns, a specified time period has passed, or an EOF is seen |
| expect_popen | Execute command via Bourne shell, and open the PTY stream to the process |
| explode | 使用一个字符串分割另一个字符串 |
| expml | 返回 exp(number) |
| extension_loaded | 检查一个扩展是否已经加载 |
| extract | 从数组中将变量导入到当前的符号表 |
| ezmlm_hash | 计算 EZMLM 所需的散列值 |

# f

| 函数 | 说明 |
|---|---|
| fam_cancel_monitor | Terminate monitoring |
| fam_close | Close FAM connection |
| fam_monitor_collection | Monitor a collection of files in a directory for changes |
| fam_monitor_directory | Monitor a directory for changes |
| fam_monitor_file | Monitor a regular file for changes |
| fam_next_event | Get next pending FAM event |
| fam_open | Open connection to FAM daemon |
| fam_pending | Check for pending FAM events |
| fam_resume_monitor | Resume suspended monitoring |
| fam_suspend_monitor | Temporarily suspend monitoring |
| fann_cascadetrain_on_data | Trains on an entire dataset, for a period of time using the Cascade2 training algorithm |
| fann_cascadetrain_on_file | Trains on an entire dataset read from file, for a period of time using the Cascade2 training algorithm. |
| fann_clear_scaling_params | Clears scaling parameters |
| fann_copy | Creates a copy of a fann structure |
| fann_create_from_file | Constructs a backpropagation neural network from a configuration file |
| fann_create_shortcut | Creates a standard backpropagation neural network which is not |

| | |
|---|---|
| | fully connectected and has shortcut connections |
| *fann_create_shortcut_array* | Creates a standard backpropagation neural network which is not fully connectected and has shortcut connections |
| *fann_create_sparse* | Creates a standard backpropagation neural network, which is not fully connected |
| *fann_create_sparse_array* | Creates a standard backpropagation neural network, which is not fully connected using an array of layer sizes |
| *fann_create_standard* | Creates a standard fully connected backpropagation neural network |
| *fann_create_standard_array* | Creates a standard fully connected backpropagation neural network using an array of layer sizes |
| *fann_create_train* | Creates an empty training data struct |
| *fann_create_train_from_callback* | Creates the training data struct from a user supplied function |
| *fann_descale_input* | Scale data in input vector after get it from ann based on previously calculated parameters |
| *fann_descale_output* | Scale data in output vector after get it from ann based on previously calculated parameters |
| *fann_descale_train* | Descale input and output data based on previously calculated parameters |
| *fann_destroy* | Destroys the entire network and properly freeing all the associated memory |
| *fann_destroy_train* | Destructs the training data |
| *fann_duplicate_train_data* | Returns an exact copy of a fann train data |
| *fann_get_activation_function* | Returns the activation function |
| *fann_get_activation_steepness* | Returns the activation steepness for supplied neuron and layer number |
| *fann_get_bias_array* | Get the number of bias in each layer in the network |
| *fann_get_bit_fail* | The number of fail bits |
| *fann_get_bit_fail_limit* | Returns the bit fail limit used during training |
| *fann_get_cascade_activation_functions* | Returns the cascade activation functions |
| *fann_get_cascade_activation_functions_count* | Returns the number of cascade activation functions |
| *fann_get_cascade_activation_steepnesses* | Returns the cascade activation steepnesses |
| *fann_get_cascade_activation_steepnesses_count* | The number of activation steepnesses |
| *fann_get_cascade_candidate_change_fraction* | Returns the cascade candidate change fraction |
| *fann_get_cascade_candidate_limit* | Return the candidate limit |
| *fann_get_cascade_candidate_stagnation_epochs* | Returns the number of cascade candidate stagnation epochs |
| *fann_get_cascade_max_cand_epochs* | Returns the maximum candidate epochs |
| *fann_get_cascade_max_out_epochs* | Returns the maximum out epochs |
| *fann_get_cascade_min_cand_epochs* | Returns the minimum candidate epochs |
| *fann_get_cascade_min_out_epochs* | Returns the minimum out epochs |
| *fann_get_cascade_num_candidates* | Returns the number of candidates used during training |
| *fann_get_cascade_num_candidate_groups* | Returns the number of candidate groups |
| *fann_get_cascade_output_change_fraction* | Returns the cascade output change fraction |

| fann_get_cascade_output_stagnation_epochs | Returns the number of cascade output stagnation epochs |
|---|---|
| fann_get_cascade_weight_multiplier | Returns the weight multiplier |
| fann_get_connection_array | Get connections in the network |
| fann_get_connection_rate | Get the connection rate used when the network was created |
| fann_get_errno | Returns the last error number |
| fann_get_errstr | Returns the last errstr |
| fann_get_layer_array | Get the number of neurons in each layer in the network |
| fann_get_learning_momentum | Returns the learning momentum |
| fann_get_learning_rate | Returns the learning rate |
| fann_get_MSE | Reads the mean square error from the network |
| fann_get_network_type | Get the type of neural network it was created as |
| fann_get_num_input | Get the number of input neurons |
| fann_get_num_layers | Get the number of layers in the neural network |
| fann_get_num_output | Get the number of output neurons |
| fann_get_quickprop_decay | Returns the decay which is a factor that weights should decrease in each iteration during quickprop training |
| fann_get_quickprop_mu | Returns the mu factor |
| fann_get_rprop_decrease_factor | Returns the increase factor used during RPROP training |
| fann_get_rprop_delta_max | Returns the maximum step-size |
| fann_get_rprop_delta_min | Returns the minimum step-size |
| fann_get_rprop_delta_zero | Returns the initial step-size |
| fann_get_rprop_increase_factor | Returns the increase factor used during RPROP training |
| fann_get_sarprop_step_error_shift | Returns the sarprop step error shift |
| fann_get_sarprop_step_error_threshold_factor | Returns the sarprop step error threshold factor |
| fann_get_sarprop_temperature | Returns the sarprop temperature |
| fann_get_sarprop_weight_decay_shift | Returns the sarprop weight decay shift |
| fann_get_total_connections | Get the total number of connections in the entire network |
| fann_get_total_neurons | Get the total number of neurons in the entire network |
| fann_get_training_algorithm | Returns the training algorithm |
| fann_get_train_error_function | Returns the error function used during training |
| fann_get_train_stop_function | Returns the stop function used during training |
| fann_init_weights | Initialize the weights using Widrow + Nguyen's algorithm |
| fann_length_train_data | Returns the number of training patterns in the train data |
| fann_merge_train_data | Merges the train data |
| fann_num_input_train_data | Returns the number of inputs in each of the training patterns in the train data |
| fann_num_output_train_data | Returns the number of outputs in each of the training patterns in the train data |
| fann_print_error | Prints the error string |
| fann_randomize_weights | Give each connection a random weight between min_weight and max_weight |
| fann_read_train_from_file | Reads a file that stores training data |
| fann_reset_errno | Resets the last error number |

| | |
|---|---|
| *fann_reset_errstr* | Resets the last error string |
| *fann_reset_MSE* | Resets the mean square error from the network |
| *fann_run* | Will run input through the neural network |
| *fann_save* | Saves the entire network to a configuration file |
| *fann_save_train* | Save the training structure to a file |
| *fann_scale_input* | Scale data in input vector before feed it to ann based on previously calculated parameters |
| *fann_scale_input_train_data* | Scales the inputs in the training data to the specified range |
| *fann_scale_output* | Scale data in output vector before feed it to ann based on previously calculated parameters |
| *fann_scale_output_train_data* | Scales the outputs in the training data to the specified range |
| *fann_scale_train* | Scale input and output data based on previously calculated parameters |
| *fann_scale_train_data* | Scales the inputs and outputs in the training data to the specified range |
| *fann_set_activation_function* | Sets the activation function for supplied neuron and layer |
| *fann_set_activation_function_hidden* | Sets the activation function for all of the hidden layers |
| *fann_set_activation_function_layer* | Sets the activation function for all the neurons in the supplied layer. |
| *fann_set_activation_function_output* | Sets the activation function for the output layer |
| *fann_set_activation_steepness* | Sets the activation steepness for supplied neuron and layer number |
| *fann_set_activation_steepness_hidden* | Sets the steepness of the activation steepness for all neurons in the all hidden layers |
| *fann_set_activation_steepness_layer* | Sets the activation steepness for all of the neurons in the supplied layer number |
| *fann_set_activation_steepness_output* | Sets the steepness of the activation steepness in the output layer |
| *fann_set_bit_fail_limit* | Set the bit fail limit used during training |
| *fann_set_callback* | Sets the callback function for use during training |
| *fann_set_cascade_activation_functions* | Sets the array of cascade candidate activation functions |
| *fann_set_cascade_activation_steepnesses* | Sets the array of cascade candidate activation steepnesses |
| *fann_set_cascade_candidate_change_fraction* | Sets the cascade candidate change fraction |
| *fann_set_cascade_candidate_limit* | Sets the candidate limit |
| *fann_set_cascade_candidate_stagnation_epochs* | Sets the number of cascade candidate stagnation epochs |
| *fann_set_cascade_max_cand_epochs* | Sets the max candidate epochs |
| *fann_set_cascade_max_out_epochs* | Sets the maximum out epochs |
| *fann_set_cascade_min_cand_epochs* | Sets the min candidate epochs |
| *fann_set_cascade_min_out_epochs* | Sets the minimum out epochs |
| *fann_set_cascade_num_candidate_groups* | Sets the number of candidate groups |
| *fann_set_cascade_output_change_fraction* | Sets the cascade output change fraction |
| *fann_set_cascade_output_stagnation_epochs* | Sets the number of cascade output stagnation epochs |
| *fann_set_cascade_weight_multiplier* | Sets the weight multiplier |
| *fann_set_error_log* | Sets where the errors are logged to |

| | |
|---|---|
| *fann_set_input_scaling_params* | Calculate input scaling parameters for future use based on training data |
| *fann_set_learning_momentum* | Sets the learning momentum |
| *fann_set_learning_rate* | Sets the learning rate |
| *fann_set_output_scaling_params* | Calculate output scaling parameters for future use based on training data |
| *fann_set_quickprop_decay* | Sets the quickprop decay factor |
| *fann_set_quickprop_mu* | Sets the quickprop mu factor |
| *fann_set_rprop_decrease_factor* | Sets the decrease factor used during RPROP training |
| *fann_set_rprop_delta_max* | Sets the maximum step-size |
| *fann_set_rprop_delta_min* | Sets the minimum step-size |
| *fann_set_rprop_delta_zero* | Sets the initial step-size |
| *fann_set_rprop_increase_factor* | Sets the increase factor used during RPROP training |
| *fann_set_sarprop_step_error_shift* | Sets the sarprop step error shift |
| *fann_set_sarprop_step_error_threshold_factor* | Sets the sarprop step error threshold factor |
| *fann_set_sarprop_temperature* | Sets the sarprop temperature |
| *fann_set_sarprop_weight_decay_shift* | Sets the sarprop weight decay shift |
| *fann_set_scaling_params* | Calculate input and output scaling parameters for future use based on training data |
| *fann_set_training_algorithm* | Sets the training algorithm |
| *fann_set_train_error_function* | Sets the error function used during training |
| *fann_set_train_stop_function* | Sets the stop function used during training |
| *fann_set_weight* | Set a connection in the network |
| *fann_set_weight_array* | Set connections in the network |
| *fann_shuffle_train_data* | Shuffles training data, randomizing the order |
| *fann_subset_train_data* | Returns an copy of a subset of the train data |
| *fann_test* | Test with a set of inputs, and a set of desired outputs |
| *fann_test_data* | Test a set of training data and calculates the MSE for the training data |
| *fann_train* | Train one iteration with a set of inputs, and a set of desired outputs |
| *fann_train_epoch* | Train one epoch with a set of training data |
| *fann_train_on_data* | Trains on an entire dataset for a period of time |
| *fann_train_on_file* | Trains on an entire dataset, which is read from file, for a period of time |
| *fastcgi_finish_request* | 冲刷(flush)所有响应的数据给客户端 |
| *fbsql_affected_rows* | Get number of affected rows in previous FrontBase operation |
| *fbsql_autocommit* | Enable or disable autocommit |
| *fbsql_blob_size* | Get the size of a BLOB |
| *fbsql_change_user* | Change logged in user of the active connection |
| *fbsql_clob_size* | Get the size of a CLOB |
| *fbsql_close* | Close FrontBase connection |
| *fbsql_commit* | Commits a transaction to the database |
| *fbsql_connect* | Open a connection to a FrontBase Server |

| | |
|---|---|
| *fbsql_create_blob* | Create a BLOB |
| *fbsql_create_clob* | Create a CLOB |
| *fbsql_create_db* | Create a FrontBase database |
| *fbsql_database* | Get or set the database name used with a connection |
| *fbsql_database_password* | Sets or retrieves the password for a FrontBase database |
| *fbsql_data_seek* | Move internal result pointer |
| *fbsql_db_query* | Send a FrontBase query |
| *fbsql_db_status* | Get the status for a given database |
| *fbsql_drop_db* | Drop (delete) a FrontBase database |
| *fbsql_errno* | Returns the error number from previous operation |
| *fbsql_error* | Returns the error message from previous operation |
| *fbsql_fetch_array* | Fetch a result row as an associative array, a numeric array, or both |
| *fbsql_fetch_assoc* | Fetch a result row as an associative array |
| *fbsql_fetch_field* | Get column information from a result and return as an object |
| *fbsql_fetch_lengths* | Get the length of each output in a result |
| *fbsql_fetch_object* | Fetch a result row as an object |
| *fbsql_fetch_row* | Get a result row as an enumerated array |
| *fbsql_field_flags* | Get the flags associated with the specified field in a result |
| *fbsql_field_len* | Returns the length of the specified field |
| *fbsql_field_name* | Get the name of the specified field in a result |
| *fbsql_field_seek* | Set result pointer to a specified field offset |
| *fbsql_field_table* | Get name of the table the specified field is in |
| *fbsql_field_type* | Get the type of the specified field in a result |
| *fbsql_free_result* | Free result memory |
| *fbsql_get_autostart_info* | 说明 |
| *fbsql_hostname* | Get or set the host name used with a connection |
| *fbsql_insert_id* | Get the id generated from the previous INSERT operation |
| *fbsql_list_dbs* | List databases available on a FrontBase server |
| *fbsql_list_fields* | List FrontBase result fields |
| *fbsql_list_tables* | List tables in a FrontBase database |
| *fbsql_next_result* | Move the internal result pointer to the next result |
| *fbsql_num_fields* | Get number of fields in result |
| *fbsql_num_rows* | Get number of rows in result |
| *fbsql_password* | Get or set the user password used with a connection |
| *fbsql_pconnect* | Open a persistent connection to a FrontBase Server |
| *fbsql_query* | Send a FrontBase query |
| *fbsql_read_blob* | Read a BLOB from the database |
| *fbsql_read_clob* | Read a CLOB from the database |
| *fbsql_result* | Get result data |
| *fbsql_rollback* | Rollback a transaction to the database |
| *fbsql_rows_fetched* | Get the number of rows affected by the last statement |
| *fbsql_select_db* | Select a FrontBase database |
| *fbsql_set_characterset* | Change input/output character set |

| | |
|---|---|
| *fbsql_set_lob_mode* | Set the LOB retrieve mode for a FrontBase result set |
| *fbsql_set_password* | Change the password for a given user |
| *fbsql_set_transaction* | Set the transaction locking and isolation |
| *fbsql_start_db* | Start a database on local or remote server |
| *fbsql_stop_db* | Stop a database on local or remote server |
| *fbsql_tablename* | 别名 fbsql_table_name |
| *fbsql_table_name* | Get table name of field |
| *fbsql_username* | Get or set the username for the connection |
| *fbsql_warnings* | Enable or disable FrontBase warnings |
| *fclose* | 关闭一个已打开的文件指针 |
| *fdf_add_doc_javascript* | Adds javascript code to the FDF document |
| *fdf_add_template* | Adds a template into the FDF document |
| *fdf_close* | Close an FDF document |
| *fdf_create* | Create a new FDF document |
| *fdf_enum_values* | Call a user defined function for each document value |
| *fdf_errno* | Return error code for last fdf operation |
| *fdf_error* | Return error description for FDF error code |
| *fdf_get_ap* | Get the appearance of a field |
| *fdf_get_attachment* | Extracts uploaded file embedded in the FDF |
| *fdf_get_encoding* | Get the value of the /Encoding key |
| *fdf_get_file* | Get the value of the /F key |
| *fdf_get_flags* | Gets the flags of a field |
| *fdf_get_opt* | Gets a value from the opt array of a field |
| *fdf_get_status* | Get the value of the /STATUS key |
| *fdf_get_value* | Get the value of a field |
| *fdf_get_version* | Gets version number for FDF API or file |
| *fdf_header* | Sets FDF-specific output headers |
| *fdf_next_field_name* | Get the next field name |
| *fdf_open* | Open a FDF document |
| *fdf_open_string* | Read a FDF document from a string |
| *fdf_remove_item* | Sets target frame for form |
| *fdf_save* | Save a FDF document |
| *fdf_save_string* | Returns the FDF document as a string |
| *fdf_set_ap* | Set the appearance of a field |
| *fdf_set_encoding* | Sets FDF character encoding |
| *fdf_set_file* | Set PDF document to display FDF data in |
| *fdf_set_flags* | Sets a flag of a field |
| *fdf_set_javascript_action* | Sets an javascript action of a field |
| *fdf_set_on_import_javascript* | Adds javascript code to be executed when Acrobat opens the FDF |
| *fdf_set_opt* | Sets an option of a field |
| *fdf_set_status* | Set the value of the /STATUS key |
| *fdf_set_submit_form_action* | Sets a submit form action of a field |
| *fdf_set_target_frame* | Set target frame for form display |
| *fdf_set_value* | Set the value of a field |

| | |
|---|---|
| *fdf_set_version* | Sets version number for a FDF file |
| *feof* | 测试文件指针是否到了文件结束的位置 |
| *fflush* | 将缓冲内容输出到文件 |
| *fgetc* | 从文件指针中读取字符 |
| *fgetcsv* | 从文件指针中读入一行并解析 CSV 字段 |
| *fgets* | 从文件指针中读取一行 |
| *fgetss* | 从文件指针中读取一行并过滤掉 HTML 标记 |
| *file* | 把整个文件读入一个数组中 |
| *fileatime* | 取得文件的上次访问时间 |
| *filectime* | 取得文件的 inode 修改时间 |
| *filegroup* | 取得文件的组 |
| *fileinode* | 取得文件的 inode |
| *filemtime* | 取得文件修改时间 |
| *fileowner* | 取得文件的所有者 |
| *fileperms* | 取得文件的权限 |
| *filepro* | Read and verify the map file |
| *filepro_fieldcount* | Find out how many fields are in a filePro database |
| *filepro_fieldname* | Gets the name of a field |
| *filepro_fieldtype* | Gets the type of a field |
| *filepro_fieldwidth* | Gets the width of a field |
| *filepro_retrieve* | Retrieves data from a filePro database |
| *filepro_rowcount* | Find out how many rows are in a filePro database |
| *filesize* | 取得文件大小 |
| *filetype* | 取得文件类型 |
| *file_exists* | 检查文件或目录是否存在 |
| *file_get_contents* | 将整个文件读入一个字符串 |
| *file_put_contents* | 将一个字符串写入文件 |
| *filter_has_var* | Checks if variable of specified type exists |
| *filter_id* | 返回与某个特定名称的过滤器相关联的 id |
| *filter_input* | 通过名称获取特定的外部变量，并且可以通过过滤器处理它 |
| *filter_input_array* | 获取一系列外部变量，并且可以通过过滤器处理它们 |
| *filter_list* | 返回所支持的过滤器列表 |
| *filter_var* | 使用特定的过滤器过滤一个变量 |
| *filter_var_array* | 获取多个变量并且过滤它们 |
| *finfo_close* | 关闭 fileinfo 资源 |
| *finfo_open* | 创建一个 fileinfo 资源 |
| *floatval* | 获取变量的浮点值 |
| *flock* | 轻便的咨询文件锁定 |
| *floor* | 舍去法取整 |
| *flush* | 刷新输出缓冲 |
| *fmod* | 返回除法的浮点数余数 |
| *fnmatch* | 用模式匹配文件名 |
| *fopen* | 打开文件或者 URL |
| *forward_static_call* | Call a static method |

| | |
|---|---|
| *forward_static_call_array* | Call a static method and pass the arguments as array |
| *fpassthru* | 输出文件指针处的所有剩余数据 |
| *fprintf* | 将格式化后的字符串写入到流 |
| *fputcsv* | 将行格式化为 CSV 并写入文件指针 |
| *fputs* | fwrite 的别名 |
| *fread* | 读取文件（可安全用于二进制文件） |
| *FrenchToJD* | 从一个 French Republican 历法的日期得到 Julian Day 计数。 |
| *fribidi_log2vis* | Convert a logical string to a visual one |
| *fscanf* | 从文件中格式化输入 |
| *fseek* | 在文件指针中定位 |
| *fsockopen* | 打开一个网络连接或者一个 Unix 套接字连接 |
| *fstat* | 通过已打开的文件指针取得文件信息 |
| *ftell* | 返回文件指针读/写的位置 |
| *ftok* | Convert a pathname and a project identifier to a System V IPC key |
| *FTP context options* | FTP context option listing |
| *ftp_alloc* | 为要上传的文件分配空间 |
| *ftp_cdup* | 切换到当前目录的父目录 |
| *ftp_chdir* | 在 FTP 服务器上改变当前目录 |
| *ftp_chmod* | 设置 FTP 服务器上的文件权限 |
| *ftp_close* | 关闭一个 FTP 连接 |
| *ftp_connect* | 建立一个新的 FTP 连接 |
| *ftp_delete* | 删除 FTP 服务器上的一个文件 |
| *ftp_exec* | 请求运行一条 FTP 命令 |
| *ftp_fget* | 从 FTP 服务器上下载一个文件并保存到本地一个已经打开的文件中 |
| *ftp_fput* | 上传一个已经打开的文件到 FTP 服务器 |
| *ftp_get* | 从 FTP 服务器上下载一个文件 |
| *ftp_get_option* | 返回当前 FTP 连接的各种不同的选项设置 |
| *ftp_login* | 登录 FTP 服务器 |
| *ftp_mdtm* | 返回指定文件的最后修改时间 |
| *ftp_mkdir* | 建立新目录 |
| *ftp_nb_continue* | 连续获取 / 发送文件（non-blocking） |
| *ftp_nb_fget* | 从 FTP 服务器获取文件并写入到一个打开的文件（非阻塞） |
| *ftp_nb_fput* | 将文件存储到 FTP 服务器 （非阻塞） |
| *ftp_nb_get* | 从 FTP 服务器上获取文件并写入本地文件（non-blocking） |
| *ftp_nb_put* | 存储一个文件至 FTP 服务器（non-blocking） |
| *ftp_nlist* | 返回给定目录的文件列表 |
| *ftp_pasv* | 返回当前 FTP 被动模式是否打开 |
| *ftp_put* | 上传文件到 FTP 服务器 |
| *ftp_pwd* | 返回当前目录名 |
| *ftp_quit* | ftp_close 的 别名 |
| *ftp_raw* | 向 FTP 服务器发送命令 |
| *ftp_rawlist* | 返回指定目录下文件的详细列表 |
| *ftp_rename* | 更改 FTP 服务器上的文件或目录名 |
| *ftp_rmdir* | 删除 FTP 服务器上的一个目录 |

| | |
|---|---|
| *ftp_set_option* | 设置各种 FTP 运行时选项 |
| *ftp_site* | 向服务器发送 SITE 命令 |
| *ftp_size* | 返回指定文件的大小 |
| *ftp_ssl_connect* | 打开 SSL-FTP 连接 |
| *ftp_systype* | 返回远程 FTP 服务器的操作系统类型 |
| *ftruncate* | 将文件截断到给定的长度 |
| *function_exists* | 如果给定的函数已经被定义就返回 TRUE |
| *func_get_arg* | 返回参数列表的某一项 |
| *func_get_args* | 返回一个包含函数参数列表的数组 |
| *func_num_args* | Returns the number of arguments passed to the function |
| | |
| *fwrite* | 写入文件（可安全用于二进制文件） |

## g

| *函数* | 说明 |
|---|---|
| *gc_collect_cycles* | 强制收集所有现存的垃圾循环周期 |
| *gc_disable* | 停用循环引用收集器 |
| *gc_enable* | 激活循环引用收集器 |
| *gc_enabled* | 返回循环引用计数器的状态 |
| *gc_mem_caches* | Reclaims memory used by the Zend Engine memory manager |
| *gd_info* | 取得当前安装的 GD 库的信息 |
| *geoip_asnum_by_name* | Get the Autonomous System Numbers (ASN) |
| *geoip_continent_code_by_name* | Get the two letter continent code |
| *geoip_country_code3_by_name* | Get the three letter country code |
| *geoip_country_code_by_name* | Get the two letter country code |
| *geoip_country_name_by_name* | Get the full country name |
| *geoip_database_info* | Get GeoIP Database information |
| *geoip_db_avail* | Determine if GeoIP Database is available |
| *geoip_db_filename* | Returns the filename of the corresponding GeoIP Database |
| *geoip_db_get_all_info* | Returns detailed information about all GeoIP database types |
| *geoip_domain_by_name* | Get the second level domain name |
| *geoip_id_by_name* | Get the Internet connection type |
| *geoip_isp_by_name* | Get the Internet Service Provider (ISP) name |
| *geoip_netspeedcell_by_name* | Get the Internet connection speed |
| *geoip_org_by_name* | Get the organization name |
| *geoip_record_by_name* | Returns the detailed City information found in the GeoIP Database |
| *geoip_region_by_name* | Get the country code and region |
| *geoip_region_name_by_code* | Returns the region name for some country and region code combo |
| *geoip_setup_custom_directory* | Set a custom directory for the GeoIP database. |
| *geoip_time_zone_by_country_and_region* | Returns the time zone for some country and region code combo |
| *getallheaders* | 获取全部 HTTP 请求头信息 |
| *getcwd* | 取得当前工作目录 |

| | |
|---|---|
| *getdate* | 取得日期／时间信息 |
| *getenv* | 获取一个环境变量的值 |
| *gethostbyaddr* | 获取指定的 IP 地址对应的主机名 |
| *gethostbyname* | Get the IPv4 address corresponding to a given Internet host name |
| *gethostbynamel* | Get a list of IPv4 addresses corresponding to a given Internet host name |
| *gethostname* | Gets the host name |
| *getimagesize* | 取得图像大小 |
| *getimagesizefromstring* | 从字符串中获取图像尺寸信息 |
| *getlastmod* | 获取页面最后修改的时间 |
| *getmxrr* | Get MX records corresponding to a given Internet host name |
| *getmygid* | 获取当前 PHP 脚本拥有者的 GID |
| *getmyinode* | 获取当前脚本的索引节点（inode） |
| *getmypid* | 获取 PHP 进程的 ID |
| *getmyuid* | 获取 PHP 脚本所有者的 UID |
| *getopt* | 从命令行参数列表中获取选项 |
| *getprotobyname* | Get protocol number associated with protocol name |
| *getprotobynumber* | Get protocol name associated with protocol number |
| *getrandmax* | 显示随机数最大的可能值 |
| *getrusage* | 获取当前资源使用状况 |
| *getservbyname* | Get port number associated with an Internet service and protocol |
| *getservbyport* | Get Internet service which corresponds to port and protocol |
| *gettext* | Lookup a message in the current domain |
| *gettimeofday* | 取得当前时间 |
| *gettype* | 获取变量的类型 |
| *get_browser* | 获取浏览器具有的功能 |
| *get_called_class* | 后期静态绑定（"Late Static Binding"）类的名称 |
| *get_cfg_var* | 获取 PHP 配置选项的值 |
| *get_class* | 返回对象的类名 |
| *get_class_methods* | 返回由类的方法名组成的数组 |
| *get_class_vars* | 返回由类的默认属性组成的数组 |
| *get_current_user* | 获取当前 PHP 脚本所有者名称 |
| *get_declared_classes* | 返回由已定义类的名字所组成的数组 |
| *get_declared_interfaces* | 返回一个数组包含所有已声明的接口 |
| *get_declared_traits* | 返回所有已定义的 traits 的数组 |
| *get_defined_constants* | 返回所有常量的关联数组，键是常量名，值是常量值 |
| *get_defined_functions* | Returns an array of all defined functions |
| *get_defined_vars* | 返回由所有已定义变量所组成的数组 |
| *get_extension_funcs* | 返回模块函数名称的数组 |
| *get_headers* | 取得服务器响应一个 HTTP 请求所发送的所有标头 |
| *get_html_translation_table* | 返回使用 htmlspecialchars 和 htmlentities 后的转换表 |
| *get_included_files* | 返回被 include 和 require 文件名的 array |
| *get_include_path* | 获取当前的 include_path 配置选项 |
| *get_loaded_extensions* | 返回所有编译并加载模块名的 array |

| | |
|---|---|
| *get_magic_quotes_gpc* | 获取当前 magic_quotes_gpc 的配置选项设置 |
| *get_magic_quotes_runtime* | 获取当前 magic_quotes_runtime 配置选项的激活状态 |
| *get_meta_tags* | 从一个文件中提取所有的 meta 标签 content 属性，返回一个数组 |
| *get_object_vars* | 返回由对象属性组成的关联数组 |
| *get_parent_class* | 返回对象或类的父类名 |
| *get_required_files* | 别名 get_included_files |
| *get_resources* | Returns active resources |
| *get_resource_type* | 返回资源（resource）类型 |
| *glob* | 寻找与模式匹配的文件路径 |
| *gmdate* | 格式化一个 GMT/UTC 日期／时间 |
| *gmmktime* | 取得 GMT 日期的 UNIX 时间戳 |
| *gmp_abs* | Absolute value |
| *gmp_add* | Add numbers |
| *gmp_and* | Bitwise AND |
| *gmp_clrbit* | Clear bit |
| *gmp_cmp* | Compare numbers |
| *gmp_com* | Calculates one's complement |
| *gmp_div* | 别名 gmp_div_q |
| *gmp_divexact* | Exact division of numbers |
| *gmp_div_q* | Divide numbers |
| *gmp_div_qr* | Divide numbers and get quotient and remainder |
| *gmp_div_r* | Remainder of the division of numbers |
| *gmp_export* | Export to a binary string |
| *gmp_fact* | Factorial |
| *gmp_gcd* | Calculate GCD |
| *gmp_gcdext* | Calculate GCD and multipliers |
| *gmp_hamdist* | Hamming distance |
| *gmp_import* | Import from a binary string |
| *gmp_init* | Create GMP number |
| *gmp_intval* | Convert GMP number to integer |
| *gmp_invert* | Inverse by modulo |
| *gmp_jacobi* | Jacobi symbol |
| *gmp_legendre* | Legendre symbol |
| *gmp_mod* | Modulo operation |
| *gmp_mul* | Multiply numbers |
| *gmp_neg* | Negate number |
| *gmp_nextprime* | Find next prime number |
| *gmp_or* | Bitwise OR |
| *gmp_perfect_square* | Perfect square check |
| *gmp_popcount* | Population count |
| *gmp_pow* | Raise number into power |
| *gmp_powm* | Raise number into power with modulo |
| *gmp_prob_prime* | Check if number is "probably prime" |
| *gmp_random* | Random number |

| | |
|---|---|
| *gmp_random_bits* | Random number |
| *gmp_random_range* | Random number |
| *gmp_random_seed* | Sets the RNG seed |
| *gmp_root* | Take the integer part of nth root |
| *gmp_rootrem* | Take the integer part and remainder of nth root |
| *gmp_scan0* | Scan for 0 |
| *gmp_scan1* | Scan for 1 |
| *gmp_setbit* | Set bit |
| *gmp_sign* | Sign of number |
| *gmp_sqrt* | Calculate square root |
| *gmp_sqrtrem* | Square root with remainder |
| *gmp_strval* | Convert GMP number to string |
| *gmp_sub* | Subtract numbers |
| *gmp_testbit* | Tests if a bit is set |
| *gmp_xor* | Bitwise XOR |
| *gmstrftime* | 根据区域设置格式化 GMT/UTC 时间／日期 |
| *gnupg_adddecryptkey* | Add a key for decryption |
| *gnupg_addencryptkey* | Add a key for encryption |
| *gnupg_addsignkey* | Add a key for signing |
| *gnupg_cleardecryptkeys* | Removes all keys which were set for decryption before |
| *gnupg_clearencryptkeys* | Removes all keys which were set for encryption before |
| *gnupg_clearsignkeys* | Removes all keys which were set for signing before |
| *gnupg_decrypt* | Decrypts a given text |
| *gnupg_decryptverify* | Decrypts and verifies a given text |
| *gnupg_encrypt* | Encrypts a given text |
| *gnupg_encryptsign* | Encrypts and signs a given text |
| *gnupg_export* | Exports a key |
| *gnupg_geterror* | Returns the errortext, if a function fails |
| *gnupg_getprotocol* | Returns the currently active protocol for all operations |
| *gnupg_import* | Imports a key |
| *gnupg_init* | Initialize a connection |
| *gnupg_keyinfo* | Returns an array with information about all keys that matches the given pattern |
| *gnupg_setarmor* | Toggle armored output |
| *gnupg_seterrormode* | Sets the mode for error_reporting |
| *gnupg_setsignmode* | Sets the mode for signing |
| *gnupg_sign* | Signs a given text |
| *gnupg_verify* | Verifies a signed text |
| *gopher_parsedir* | Translate a gopher formatted directory entry into an associative array. |
| *grapheme_extract* | Function to extract a sequence of default grapheme clusters from a text buffer, which must be encoded in UTF-8. |
| *grapheme_stripos* | Find position (in grapheme units) of first occurrence of a case-insensitive string |
| *grapheme_stristr* | Returns part of haystack string from the first occurrence of |

| | case-insensitive needle to the end of haystack. |
|---|---|
| *grapheme_strlen* | Get string length in grapheme units |
| *grapheme_strpos* | Find position (in grapheme units) of first occurrence of a string |
| *grapheme_strripos* | Find position (in grapheme units) of last occurrence of a case-insensitive string |
| *grapheme_strrpos* | Find position (in grapheme units) of last occurrence of a string |
| *grapheme_strstr* | Returns part of haystack string from the first occurrence of needle to the end of haystack. |
| *grapheme_substr* | Return part of a string |
| *GregorianToJD* | 转变一个 Gregorian 历法日期到 Julian Day 计数 |
| *gupnp_context_get_host_ip* | Get the IP address |
| *gupnp_context_get_port* | Get the port |
| *gupnp_context_get_subscription_timeout* | Get the event subscription timeout |
| *gupnp_context_host_path* | Start hosting |
| *gupnp_context_new* | Create a new context |
| *gupnp_context_set_subscription_timeout* | Sets the event subscription timeout |
| *gupnp_context_timeout_add* | Sets a function to be called at regular intervals |
| *gupnp_context_unhost_path* | Stop hosting |
| *gupnp_control_point_browse_start* | Start browsing |
| *gupnp_control_point_browse_stop* | Stop browsing |
| *gupnp_control_point_callback_set* | Set control point callback |
| *gupnp_control_point_new* | Create a new control point |
| *gupnp_device_action_callback_set* | Set device callback function |
| *gupnp_device_info_get* | Get info of root device |
| *gupnp_device_info_get_service* | Get the service with type |
| *gupnp_root_device_get_available* | Check whether root device is available |
| *gupnp_root_device_get_relative_location* | Get the relative location of root device. |
| *gupnp_root_device_new* | Create a new root device |
| *gupnp_root_device_set_available* | Set whether or not root_device is available |
| *gupnp_root_device_start* | Start main loop |
| *gupnp_root_device_stop* | Stop main loop |
| *gupnp_service_action_get* | Retrieves the specified action arguments |
| *gupnp_service_action_return* | Return successfully |
| *gupnp_service_action_return_error* | Return error code |
| *gupnp_service_action_set* | Sets the specified action return values |
| *gupnp_service_freeze_notify* | Freeze new notifications |
| *gupnp_service_info_get* | Get full info of service |
| *gupnp_service_info_get_introspection* | Get resource introspection of service |
| *gupnp_service_introspection_get_state_variable* | Returns the state variable data |
| *gupnp_service_notify* | Notifies listening clients |
| *gupnp_service_proxy_action_get* | Send action to the service and get value |
| *gupnp_service_proxy_action_set* | Send action to the service and set value |
| *gupnp_service_proxy_add_notify* | Sets up callback for variable change notification |

| | |
|---|---|
| *gupnp_service_proxy_callback_set* | Set service proxy callback for signal |
| *gupnp_service_proxy_get_subscribed* | Check whether subscription is valid to the service |
| *gupnp_service_proxy_remove_notify* | Cancels the variable change notification |
| *gupnp_service_proxy_send_action* | Send action with multiple parameters synchronously |
| *gupnp_service_proxy_set_subscribed* | (Un)subscribes to the service. |
| *gupnp_service_thaw_notify* | Sends out any pending notifications and stops queuing of new ones. |
| *gzclose* | Close an open gz-file pointer |
| *gzcompress* | Compress a string |
| *gzdecode* | Decodes a gzip compressed string |
| *gzdeflate* | Deflate a string |
| *gzencode* | Create a gzip compressed string |
| *gzeof* | Test for EOF on a gz-file pointer |
| *gzfile* | Read entire gz-file into an array |
| *gzgetc* | Get character from gz-file pointer |
| *gzgets* | Get line from file pointer |
| *gzgetss* | Get line from gz-file pointer and strip HTML tags |
| *gzinflate* | Inflate a deflated string |
| *gzopen* | Open gz-file |
| *gzpassthru* | Output all remaining data on a gz-file pointer |
| *gzputs* | 别名 gzwrite |
| *gzread* | Binary-safe gz-file read |
| *gzrewind* | Rewind the position of a gz-file pointer |
| *gzseek* | Seek on a gz-file pointer |
| *gztell* | Tell gz-file pointer read/write position |
| *gzuncompress* | Uncompress a compressed string |
| *gzwrite* | Binary-safe gz-file write |

# h

| *函数* | 说明 |
|---|---|
| *hash* | 生成哈希值 （消息摘要） |
| *hash_algos* | 返回已注册的哈希算法列表 |
| *hash_copy* | 拷贝哈希运算上下文 |
| *hash_equals* | 可防止时序攻击的字符串比较 |
| *hash_file* | 使用给定文件的内容生成哈希值 |
| *hash_final* | 结束增量哈希，并且返回摘要结果 |
| *hash_hmac* | 使用 HMAC 方法生成带有密钥的哈希值 |
| *hash_hmac_file* | 使用 HMAC 方法和给定文件的内容生成带密钥的哈希值 |
| *hash_init* | 初始化增量哈希运算上下文 |
| *hash_pbkdf2* | 生成所提供密码的 PBKDF2 密钥导出 |
| *hash_update* | 向活跃的哈希运算上下文中填充数据 |
| *hash_update_file* | 从文件向活跃的哈希运算上下文中填充数据 |
| *hash_update_stream* | 从打开的流向活跃的哈希运算上下文中填充数据 |
| *header* | 发送原生 HTTP 头 |

| | |
|---|---|
| *headers_list* | Returns a list of response headers sent (or ready to send) |
| *headers_sent* | Checks if or where headers have been sent |
| *header_register_callback* | Call a header function |
| *header_remove* | Remove previously set headers |
| *hebrev* | 将逻辑顺序希伯来文（logical-Hebrew）转换为视觉顺序希伯来文（visual-Hebrew） |
| *hebrevc* | 将逻辑顺序希伯来文（logical-Hebrew）转换为视觉顺序希伯来文（visual-Hebrew），并且转换换行符 |
| *hex2bin* | 转换十六进制字符串为二进制字符串 |
| *hexdec* | 十六进制转换为十进制 |
| *highlight_file* | 语法高亮一个文件 |
| *highlight_string* | 字符串的语法高亮 |
| *htmlentities* | Convert all applicable characters to HTML entities |
| *htmlspecialchars* | Convert special characters to HTML entities |
| *htmlspecialchars_decode* | 将特殊的 HTML 实体转换回普通字符 |
| *html_entity_decode* | Convert all HTML entities to their applicable characters |
| *HTTP context 选项* | HTTP context 的选项列表 |
| *http_build_cookie* | Build cookie string |
| *http_build_query* | 生成 URL-encode 之后的请求字符串 |
| *http_build_str* | 产生一个查询字符串 |
| *http_build_url* | 产生一个 URL |
| *http_cache_etag* | Caching by ETag |
| *http_cache_last_modified* | Caching by last modification |
| *http_chunked_decode* | Decode chunked-encoded data |
| *http_date* | Compose HTTP RFC compliant date |
| *http_deflate* | Deflate data |
| *http_get* | Perform GET request |
| *http_get_request_body* | Get request body as string |
| *http_get_request_body_stream* | Get request body as stream |
| *http_get_request_headers* | Get request headers as array |
| *http_head* | Perform HEAD request |
| *http_inflate* | Inflate data |
| *http_match_etag* | Match ETag |
| *http_match_modified* | Match last modification |
| *http_match_request_header* | Match any header |
| *http_negotiate_charset* | Negotiate client's preferred character set |
| *http_negotiate_content_type* | Negotiate client's preferred content type |
| *http_negotiate_language* | Negotiate client's preferred language |
| *http_parse_cookie* | Parse HTTP cookie |
| *http_parse_headers* | Parse HTTP headers |
| *http_parse_message* | Parse HTTP messages |
| *http_parse_params* | Parse parameter list |
| *http_persistent_handles_clean* | Clean up persistent handles |
| *http_persistent_handles_count* | Stat persistent handles |

| | |
|---|---|
| *http_persistent_handles_ident* | Get/set ident of persistent handles |
| *http_post_data* | Perform POST request with pre-encoded data |
| *http_post_fields* | Perform POST request with data to be encoded |
| *http_put_data* | Perform PUT request with data |
| *http_put_file* | Perform PUT request with file |
| *http_put_stream* | Perform PUT request with stream |
| *http_redirect* | Issue HTTP redirect |
| *http_request* | Perform custom request |
| *http_request_body_encode* | Encode request body |
| *http_request_method_exists* | Check whether request method exists |
| *http_request_method_name* | Get request method name |
| *http_request_method_register* | Register request method |
| *http_request_method_unregister* | Unregister request method |
| *http_response_code* | Get or Set the HTTP response code |
| *http_send_content_disposition* | Send Content-Disposition |
| *http_send_content_type* | Send Content-Type |
| *http_send_data* | Send arbitrary data |
| *http_send_file* | Send file |
| *http_send_last_modified* | Send Last-Modified |
| *http_send_status* | Send HTTP response status |
| *http_send_stream* | Send stream |
| *http_support* | Check built-in HTTP support |
| *http_throttle* | HTTP throttling |
| *hwapi_attribute_new* | Creates instance of class hw_api_attribute |
| *hwapi_content_new* | Create new instance of class hw_api_content |
| *hwapi_hgcsp* | Returns object of class hw_api |
| *hwapi_object_new* | Creates a new instance of class hwapi_object_new |
| *hypot* | 计算一直角三角形的斜边长度 |

# i

| *函数* | 说明 |
|---|---|
| *ibase_add_user* | Add a user to a security database |
| *ibase_affected_rows* | Return the number of rows that were affected by the previous query |
| *ibase_backup* | Initiates a backup task in the service manager and returns immediately |
| *ibase_blob_add* | Add data into a newly created blob |
| *ibase_blob_cancel* | Cancel creating blob |
| *ibase_blob_close* | Close blob |
| *ibase_blob_create* | Create a new blob for adding data |
| *ibase_blob_echo* | Output blob contents to browser |
| *ibase_blob_get* | Get len bytes data from open blob |
| *ibase_blob_import* | Create blob, copy file in it, and close it |
| *ibase_blob_info* | Return blob length and other useful info |

| | |
|---|---|
| *ibase_blob_open* | Open blob for retrieving data parts |
| *ibase_close* | Close a connection to an InterBase database |
| *ibase_commit* | Commit a transaction |
| *ibase_commit_ret* | Commit a transaction without closing it |
| *ibase_connect* | Open a connection to a database |
| *ibase_db_info* | Request statistics about a database |
| *ibase_delete_user* | Delete a user from a security database |
| *ibase_drop_db* | Drops a database |
| *ibase_errcode* | Return an error code |
| *ibase_errmsg* | Return error messages |
| *ibase_execute* | Execute a previously prepared query |
| *ibase_fetch_assoc* | Fetch a result row from a query as an associative array |
| *ibase_fetch_object* | Get an object from a InterBase database |
| *ibase_fetch_row* | Fetch a row from an InterBase database |
| *ibase_field_info* | Get information about a field |
| *ibase_free_event_handler* | Cancels a registered event handler |
| *ibase_free_query* | Free memory allocated by a prepared query |
| *ibase_free_result* | Free a result set |
| *ibase_gen_id* | Increments the named generator and returns its new value |
| *ibase_maintain_db* | Execute a maintenance command on the database server |
| *ibase_modify_user* | Modify a user to a security database |
| *ibase_name_result* | Assigns a name to a result set |
| *ibase_num_fields* | Get the number of fields in a result set |
| *ibase_num_params* | Return the number of parameters in a prepared query |
| *ibase_param_info* | Return information about a parameter in a prepared query |
| *ibase_pconnect* | Open a persistent connection to an InterBase database |
| *ibase_prepare* | Prepare a query for later binding of parameter placeholders and execution |
| *ibase_query* | Execute a query on an InterBase database |
| *ibase_restore* | Initiates a restore task in the service manager and returns immediately |
| *ibase_rollback* | Roll back a transaction |
| *ibase_rollback_ret* | Roll back a transaction without closing it |
| *ibase_server_info* | Request information about a database server |
| *ibase_service_attach* | Connect to the service manager |
| *ibase_service_detach* | Disconnect from the service manager |
| *ibase_set_event_handler* | Register a callback function to be called when events are posted |
| *ibase_trans* | Begin a transaction |
| *ibase_wait_event* | Wait for an event to be posted by the database |
| *iconv* | 字符串按要求的字符编码来转换 |
| *iconv_get_encoding* | 获取 iconv 扩展的内部配置变量 |
| *iconv_mime_decode* | Decodes a MIME header field |
| *iconv_mime_decode_headers* | 一次性解码多个 MIME 头字段 |
| *iconv_mime_encode* | Composes a MIME header field |

| | |
|---|---|
| *iconv_set_encoding* | 为字符编码转换设定当前设置 |
| *iconv_strlen* | 返回字符串的字符数统计 |
| *iconv_strpos* | Finds position of first occurrence of a needle within a haystack |
| *iconv_strrpos* | Finds the last occurrence of a needle within a haystack |
| *iconv_substr* | 截取字符串的部分 |
| *id3_get_frame_long_name* | Get the long name of an ID3v2 frame |
| *id3_get_frame_short_name* | Get the short name of an ID3v2 frame |
| *id3_get_genre_id* | Get the id for a genre |
| *id3_get_genre_list* | Get all possible genre values |
| *id3_get_genre_name* | Get the name for a genre id |
| *id3_get_tag* | Get all information stored in an ID3 tag |
| *id3_get_version* | Get version of an ID3 tag |
| *id3_remove_tag* | Remove an existing ID3 tag |
| *id3_set_tag* | Update information stored in an ID3 tag |
| *idate* | 将本地时间日期格式化为整数 |
| *idn_to_ascii* | Convert domain name to IDNA ASCII form. |
| *idn_to_unicode* | 别名 idn_to_utf8 |
| *idn_to_utf8* | Convert domain name from IDNA ASCII to Unicode. |
| *ifxus_close_slob* | Deletes the slob object |
| *ifxus_create_slob* | Creates an slob object and opens it |
| *ifxus_free_slob* | Deletes the slob object |
| *ifxus_open_slob* | Opens an slob object |
| *ifxus_read_slob* | Reads nbytes of the slob object |
| *ifxus_seek_slob* | Sets the current file or seek position |
| *ifxus_tell_slob* | Returns the current file or seek position |
| *ifxus_write_slob* | Writes a string into the slob object |
| *ifx_affected_rows* | Get number of rows affected by a query |
| *ifx_blobinfile_mode* | Set the default blob mode for all select queries |
| *ifx_byteasvarchar* | Set the default byte mode |
| *ifx_close* | Close Informix connection |
| *ifx_connect* | Open Informix server connection |
| *ifx_copy_blob* | Duplicates the given blob object |
| *ifx_create_blob* | Creates an blob object |
| *ifx_create_char* | Creates an char object |
| *ifx_do* | Execute a previously prepared SQL-statement |
| *ifx_error* | Returns error code of last Informix call |
| *ifx_errormsg* | Returns error message of last Informix call |
| *ifx_fetch_row* | Get row as an associative array |
| *ifx_fieldproperties* | List of SQL fieldproperties |
| *ifx_fieldtypes* | List of Informix SQL fields |
| *ifx_free_blob* | Deletes the blob object |
| *ifx_free_char* | Deletes the char object |
| *ifx_free_result* | Releases resources for the query |
| *ifx_getsqlca* | Get the contents of sqlca.sqlerrd[0..5] after a query |

| | |
|---|---|
| *ifx_get_blob* | Return the content of a blob object |
| *ifx_get_char* | Return the content of the char object |
| *ifx_htmltbl_result* | Formats all rows of a query into a HTML table |
| *ifx_nullformat* | Sets the default return value on a fetch row |
| *ifx_num_fields* | Returns the number of columns in the query |
| *ifx_num_rows* | Count the rows already fetched from a query |
| *ifx_pconnect* | Open persistent Informix connection |
| *ifx_prepare* | Prepare an SQL-statement for execution |
| *ifx_query* | Send Informix query |
| *ifx_textasvarchar* | Set the default text mode |
| *ifx_update_blob* | Updates the content of the blob object |
| *ifx_update_char* | Updates the content of the char object |
| *ignore_user_abort* | 设置客户端断开连接时是否中断脚本的执行 |
| *iis_add_server* | Creates a new virtual web server |
| *iis_get_dir_security* | Gets Directory Security |
| *iis_get_script_map* | Gets script mapping on a virtual directory for a specific extension |
| *iis_get_server_by_comment* | Return the instance number associated with the Comment |
| *iis_get_server_by_path* | Return the instance number associated with the Path |
| *iis_get_server_rights* | Gets server rights |
| *iis_get_service_state* | Returns the state for the service defined by ServiceId |
| *iis_remove_server* | Removes the virtual web server indicated by ServerInstance |
| *iis_set_app_settings* | Creates application scope for a virtual directory |
| *iis_set_dir_security* | Sets Directory Security |
| *iis_set_script_map* | Sets script mapping on a virtual directory |
| *iis_set_server_rights* | Sets server rights |
| *iis_start_server* | Starts the virtual web server |
| *iis_start_service* | Starts the service defined by ServiceId |
| *iis_stop_server* | Stops the virtual web server |
| *iis_stop_service* | Stops the service defined by ServiceId |
| *image2wbmp* | 以 WBMP 格式将图像输出到浏览器或文件 |
| *imageaffine* | 返回经过仿射变换后的图像，剪切区域可选 |
| *imageaffinematrixconcat* | Concat two matrices (as in doing many ops in one go) |
| *imageaffinematrixget* | Return an image containing the affine tramsformed src image, using an optional clipping area |
| *imagealphablending* | 设定图像的混色模式 |
| *imageantialias* | 是否使用抗锯齿（antialias）功能 |
| *imagearc* | 画椭圆弧 |
| *imagechar* | 水平地画一个字符 |
| *imagecharup* | 垂直地画一个字符 |
| *imagecolorallocate* | 为一幅图像分配颜色 |
| *imagecolorallocatealpha* | 为一幅图像分配颜色 + alpha |
| *imagecolorat* | 取得某像素的颜色索引值 |
| *imagecolorclosest* | 取得与指定的颜色最接近的颜色的索引值 |
| *imagecolorclosestalpha* | 取得与指定的颜色加透明度最接近的颜色 |

| | |
|---|---|
| *imagecolorclosesthwb* | 取得与给定颜色最接近的色度的黑白色的索引 |
| *imagecolordeallocate* | 取消图像颜色的分配 |
| *imagecolorexact* | 取得指定颜色的索引值 |
| *imagecolorexactalpha* | 取得指定的颜色加透明度的索引值 |
| *imagecolormatch* | 使一个图像中调色板版本的颜色与真彩色版本更能匹配 |
| *imagecolorresolve* | 取得指定颜色的索引值或有可能得到的最接近的替代值 |
| *imagecolorresolvealpha* | 取得指定颜色 + alpha 的索引值或有可能得到的最接近的替代值 |
| *imagecolorset* | 给指定调色板索引设定颜色 |
| *imagecolorsforindex* | 取得某索引的颜色 |
| *imagecolorstotal* | 取得一幅图像的调色板中颜色的数目 |
| *imagecolortransparent* | 将某个颜色定义为透明色 |
| *imageconvolution* | 用系数 div 和 offset 申请一个 3x3 的卷积矩阵 |
| *imagecopy* | 拷贝图像的一部分 |
| *imagecopymerge* | 拷贝并合并图像的一部分 |
| *imagecopymergegray* | 用灰度拷贝并合并图像的一部分 |
| *imagecopyresampled* | 重采样拷贝部分图像并调整大小 |
| *imagecopyresized* | 拷贝部分图像并调整大小 |
| *imagecreate* | 新建一个基于调色板的图像 |
| *imagecreatefromgd* | 从 GD 文件或 URL 新建一图像 |
| *imagecreatefromgd2* | 从 GD2 文件或 URL 新建一图像 |
| *imagecreatefromgd2part* | 从给定的 GD2 文件或 URL 中的部分新建一图像 |
| *imagecreatefromgif* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefromjpeg* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefrompng* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefromstring* | 从字符串中的图像流新建一图像 |
| *imagecreatefromwbmp* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefromwebp* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefromxbm* | 由文件或 URL 创建一个新图象。 |
| *imagecreatefromxpm* | 由文件或 URL 创建一个新图象。 |
| *imagecreatetruecolor* | 新建一个真彩色图像 |
| *imagecrop* | Crop an image using the given coordinates and size, x, y, width and height |
| *imagecropauto* | Crop an image automatically using one of the available modes |
| *imagedashedline* | 画一虚线 |
| *imagedestroy* | 销毁一图像 |
| *imageellipse* | 画一个椭圆 |
| *imagefill* | 区域填充 |
| *imagefilledarc* | 画一椭圆弧且填充 |
| *imagefilledellipse* | 画一椭圆并填充 |
| *imagefilledpolygon* | 画一多边形并填充 |
| *imagefilledrectangle* | 画一矩形并填充 |
| *imagefilltoborder* | 区域填充到指定颜色的边界为止 |
| *imagefilter* | 对图像使用过滤器 |
| *imageflip* | Flips an image using a given mode |

| | |
|---|---|
| *imagefontheight* | 取得字体高度 |
| *imagefontwidth* | 取得字体宽度 |
| *imageftbbox* | 给出一个使用 FreeType 2 字体的文本框 |
| *imagefttext* | 使用 FreeType 2 字体将文本写入图像 |
| *imagegammacorrect* | 对 GD 图像应用 gamma 修正 |
| *imagegd* | 将 GD 图像输出到浏览器或文件 |
| *imagegd2* | 将 GD2 图像输出到浏览器或文件 |
| *imagegif* | 输出图象到浏览器或文件。 |
| *imagegrabscreen* | Captures the whole screen |
| *imagegrabwindow* | Captures a window |
| *imageinterlace* | 激活或禁止隔行扫描 |
| *imageistruecolor* | 检查图像是否为真彩色图像 |
| *imagejpeg* | 输出图象到浏览器或文件。 |
| *imagelayereffect* | 设定 alpha 混色标志以使用绑定的 libgd 分层效果 |
| *imageline* | 画一条线段 |
| *imageloadfont* | 载入一新字体 |
| *imagepalettecopy* | 将调色板从一幅图像拷贝到另一幅 |
| *imagepalettetotruecolor* | Converts a palette based image to true color |
| *imagepng* | 以 PNG 格式将图像输出到浏览器或文件 |
| *imagepolygon* | 画一个多边形 |
| *imagepsbbox* | 给出一个使用 PostScript Type1 字体的文本方框 |
| *imagepsencodefont* | 改变字体中的字符编码矢量 |
| *imagepsextendfont* | 扩充或精简字体 |
| *imagepsfreefont* | 释放一个 PostScript Type 1 字体所占用的内存 |
| *imagepsloadfont* | 从文件中加载一个 PostScript Type 1 字体 |
| *imagepsslantfont* | 倾斜某字体 |
| *imagepstext* | 用 PostScript Type1 字体把文本字符串画在图像上 |
| *imagerectangle* | 画一个矩形 |
| *imagerotate* | 用给定角度旋转图像 |
| *imagesavealpha* | 设置标记以在保存 PNG 图像时保存完整的 alpha 通道信息（与单一透明色相反） |
| *imagescale* | Scale an image using the given new width and height |
| *imagesetbrush* | 设定画线用的画笔图像 |
| *imagesetinterpolation* | Set the interpolation method |
| *imagesetpixel* | 画一个单一像素 |
| *imagesetstyle* | 设定画线的风格 |
| *imagesetthickness* | 设定画线的宽度 |
| *imagesettile* | 设定用于填充的贴图 |
| *imagestring* | 水平地画一行字符串 |
| *imagestringup* | 垂直地画一行字符串 |
| *imagesx* | 取得图像宽度 |
| *imagesy* | 取得图像高度 |
| *imagetruecolortopalette* | 将真彩色图像转换为调色板图像 |
| *imagettfbbox* | 取得使用 TrueType 字体的文本的范围 |

| | |
|---|---|
| *imagettftext* | 用 TrueType 字体向图像写入文本 |
| *imagetypes* | 返回当前 PHP 版本所支持的图像类型 |
| *imagewbmp* | 以 WBMP 格式将图像输出到浏览器或文件 |
| *imagewebp* | 将 WebP 格式的图像输出到浏览器或文件 |
| *imagexbm* | 将 XBM 图像输出到浏览器或文件 |
| *image_type_to_extension* | 取得图像类型的文件后缀 |
| *image_type_to_mime_type* | 取得 getimagesize，exif_read_data，exif_thumbnail，exif_imagetype 所返回的图像类型的 MIME 类型 |
| *imap_8bit* | Convert an 8bit string to a quoted-printable string |
| *imap_alerts* | Returns all IMAP alert messages that have occurred |
| *imap_append* | Append a string message to a specified mailbox |
| *imap_base64* | Decode BASE64 encoded text |
| *imap_binary* | Convert an 8bit string to a base64 string |
| *imap_body* | Read the message body |
| *imap_bodystruct* | Read the structure of a specified body section of a specific message |
| *imap_check* | Check current mailbox |
| *imap_clearflag_full* | Clears flags on messages |
| *imap_close* | Close an IMAP stream |
| *imap_create* | 别名 imap_createmailbox |
| *imap_createmailbox* | Create a new mailbox |
| *imap_delete* | Mark a message for deletion from current mailbox |
| *imap_deletemailbox* | Delete a mailbox |
| *imap_errors* | Returns all of the IMAP errors that have occurred |
| *imap_expunge* | Delete all messages marked for deletion |
| *imap_fetchbody* | Fetch a particular section of the body of the message |
| *imap_fetchheader* | Returns header for a message |
| *imap_fetchmime* | Fetch MIME headers for a particular section of the message |
| *imap_fetchstructure* | Read the structure of a particular message |
| *imap_fetchtext* | 别名 imap_body |
| *imap_fetch_overview* | Read an overview of the information in the headers of the given message |
| *imap_gc* | Clears IMAP cache |
| *imap_getacl* | Gets the ACL for a given mailbox |
| *imap_getmailboxes* | Read the list of mailboxes, returning detailed information on each one |
| *imap_getsubscribed* | List all the subscribed mailboxes |
| *imap_get_quota* | Retrieve the quota level settings, and usage statics per mailbox |
| *imap_get_quotaroot* | Retrieve the quota settings per user |
| *imap_header* | 别名 imap_headerinfo |
| *imap_headerinfo* | Read the header of the message |
| *imap_headers* | Returns headers for all messages in a mailbox |
| *imap_last_error* | Gets the last IMAP error that occurred during this page request |
| *imap_list* | Read the list of mailboxes |
| *imap_listmailbox* | 别名 imap_list |

| | |
|---|---|
| *imap_listscan* | Returns the list of mailboxes that matches the given text |
| *imap_listsubscribed* | 别名 imap_lsub |
| *imap_lsub* | List all the subscribed mailboxes |
| *imap_mail* | Send an email message |
| *imap_mailboxmsginfo* | Get information about the current mailbox |
| *imap_mail_compose* | Create a MIME message based on given envelope and body sections |
| *imap_mail_copy* | Copy specified messages to a mailbox |
| *imap_mail_move* | Move specified messages to a mailbox |
| *imap_mime_header_decode* | Decode MIME header elements |
| *imap_msgno* | Gets the message sequence number for the given UID |
| *imap_num_msg* | Gets the number of messages in the current mailbox |
| *imap_num_recent* | Gets the number of recent messages in current mailbox |
| *imap_open* | Open an IMAP stream to a mailbox |
| *imap_ping* | Check if the IMAP stream is still active |
| *imap_qprint* | Convert a quoted-printable string to an 8 bit string |
| *imap_rename* | 别名 imap_renamemailbox |
| *imap_renamemailbox* | Rename an old mailbox to new mailbox |
| *imap_reopen* | Reopen IMAP stream to new mailbox |
| *imap_rfc822_parse_adrlist* | Parses an address string |
| *imap_rfc822_parse_headers* | Parse mail headers from a string |
| *imap_rfc822_write_address* | Returns a properly formatted email address given the mailbox, host, and personal info |
| *imap_savebody* | Save a specific body section to a file |
| *imap_scan* | 别名 imap_listscan |
| *imap_scanmailbox* | 别名 imap_listscan |
| *imap_search* | This function returns an array of messages matching the given search criteria |
| *imap_setacl* | Sets the ACL for a given mailbox |
| *imap_setflag_full* | Sets flags on messages |
| *imap_set_quota* | Sets a quota for a given mailbox |
| *imap_sort* | Gets and sort messages |
| *imap_status* | Returns status information on a mailbox |
| *imap_subscribe* | Subscribe to a mailbox |
| *imap_thread* | Returns a tree of threaded message |
| *imap_timeout* | Set or fetch imap timeout |
| *imap_uid* | This function returns the UID for the given message sequence number |
| *imap_undelete* | Unmark the message which is marked deleted |
| *imap_unsubscribe* | Unsubscribe from a mailbox |
| *imap_utf7_decode* | Decodes a modified UTF-7 encoded string |
| *imap_utf7_encode* | Converts ISO-8859-1 string to modified UTF-7 text |
| *imap_utf8* | Converts MIME-encoded text to UTF-8 |
| *implode* | 将一个一维数组的值转化为字符串 |
| *import_request_variables* | 将 GET / POST / Cookie 变量导入到全局作用域中 |
| *inclued_get_data* | Get the inclued data |

| | |
|---|---|
| *inet_ntop* | Converts a packed internet address to a human readable representation |
| *inet_pton* | Converts a human readable IP address to its packed in_addr representation |
| *ingres_autocommit* | Switch autocommit on or off |
| *ingres_autocommit_state* | Test if the connection is using autocommit |
| *ingres_charset* | Returns the installation character set |
| *ingres_close* | Close an Ingres database connection |
| *ingres_commit* | Commit a transaction |
| *ingres_connect* | Open a connection to an Ingres database |
| *ingres_cursor* | Get a cursor name for a given result resource |
| *ingres_errno* | Get the last Ingres error number generated |
| *ingres_error* | Get a meaningful error message for the last error generated |
| *ingres_errsqlstate* | Get the last SQLSTATE error code generated |
| *ingres_escape_string* | Escape special characters for use in a query |
| *ingres_execute* | Execute a prepared query |
| *ingres_fetch_array* | Fetch a row of result into an array |
| *ingres_fetch_assoc* | Fetch a row of result into an associative array |
| *ingres_fetch_object* | Fetch a row of result into an object |
| *ingres_fetch_proc_return* | Get the return value from a procedure call |
| *ingres_fetch_row* | Fetch a row of result into an enumerated array |
| *ingres_field_length* | Get the length of a field |
| *ingres_field_name* | Get the name of a field in a query result |
| *ingres_field_nullable* | Test if a field is nullable |
| *ingres_field_precision* | Get the precision of a field |
| *ingres_field_scale* | Get the scale of a field |
| *ingres_field_type* | Get the type of a field in a query result |
| *ingres_free_result* | Free the resources associated with a result identifier |
| *ingres_next_error* | Get the next Ingres error |
| *ingres_num_fields* | Get the number of fields returned by the last query |
| *ingres_num_rows* | Get the number of rows affected or returned by a query |
| *ingres_pconnect* | Open a persistent connection to an Ingres database |
| *ingres_prepare* | Prepare a query for later execution |
| *ingres_query* | Send an SQL query to Ingres |
| *ingres_result_seek* | Set the row position before fetching data |
| *ingres_rollback* | Roll back a transaction |
| *ingres_set_environment* | Set environment features controlling output options |
| *ingres_unbuffered_query* | Send an unbuffered SQL query to Ingres |
| *ini_alter* | 别名 ini_set |
| *ini_get* | 获取一个配置选项的值 |
| *ini_get_all* | 获取所有配置选项 |
| *ini_restore* | 恢复配置选项的值 |
| *ini_set* | 为一个配置选项设置值 |
| *inotify_add_watch* | Add a watch to an initialized inotify instance |

| | |
|---|---|
| *inotify_init* | Initialize an inotify instance |
| *inotify_queue_len* | Return a number upper than zero if there are pending events |
| *inotify_read* | Read events from an inotify instance |
| *inotify_rm_watch* | Remove an existing watch from an inotify instance |
| *intdiv* | Integer division |
| *interface_exists* | 检查接口是否已被定义 |
| *intl_error_name* | Get symbolic name for a given error code |
| *intl_get_error_code* | Get the last error code |
| *intl_get_error_message* | Get description of the last error |
| *intl_is_failure* | Check whether the given error code indicates failure |
| *intval* | 获取变量的整数值 |
| *in_array* | 检查数组中是否存在某个值 |
| *ip2long* | 将一个 IPV4 的字符串互联网协议转换成数字格式 |
| *iptcembed* | 将二进制 IPTC 数据嵌入到一幅 JPEG 图像中 |
| *iptcparse* | 将二进制 IPTC 块解析为单个标记 |
| *isset* | 检测变量是否设置 |
| *is_a* | 如果对象属于该类或该类是此对象的父类则返回 TRUE |
| *is_array* | 检测变量是否是数组 |
| *is_bool* | 检测变量是否是布尔型 |
| *is_callable* | 检测参数是否为合法的可调用结构 |
| *is_dir* | 判断给定文件名是否是一个目录 |
| *is_double* | is_float 的别名 |
| *is_executable* | 判断给定文件名是否可执行 |
| *is_file* | 判断给定文件名是否为一个正常的文件 |
| *is_finite* | 判断是否为有限值 |
| *is_float* | 检测变量是否是浮点型 |
| *is_infinite* | 判断是否为无限值 |
| *is_int* | 检测变量是否是整数 |
| *is_integer* | is_int 的别名 |
| *is_link* | 判断给定文件名是否为一个符号连接 |
| *is_long* | is_int 的别名 |
| *is_nan* | 判断是否为合法数值 |
| *is_null* | 检测变量是否为 NULL |
| *is_numeric* | 检测变量是否为数字或数字字符串 |
| *is_object* | 检测变量是否是一个对象 |
| *is_readable* | 判断给定文件名是否可读 |
| *is_real* | is_float 的别名 |
| *is_resource* | 检测变量是否为资源类型 |
| *is_scalar* | 检测变量是否是一个标量 |
| *is_soap_fault* | Checks if a SOAP call has failed |
| *is_string* | 检测变量是否是字符串 |
| *is_subclass_of* | 如果此对象是该类的子类，则返回 TRUE |
| *is_tainted* | Checks whether a string is tainted |
| *is_uploaded_file* | 判断文件是否是通过 HTTP POST 上传的 |

| | |
|---|---|
| *is_writable* | 判断给定的文件名是否可写 |
| *is_writeable* | is_writable 的别名 |
| *iterator_apply* | 为迭代器中每个元素调用一个用户自定义函数 |
| *iterator_count* | 计算迭代器中元素的个数 |
| | |
| *iterator_to_array* | 将迭代器中的元素拷贝到数组 |

# j

| *函数* | 说明 |
|---|---|
| *JDDayOfWeek* | 返回星期的日期 |
| *JDMonthName* | 返回月份的名称 |
| *JDToFrench* | 转变一个 Julian Day 计数到 French Republican 历法的日期 |
| *JDToGregorian* | 转变一个 Julian Day 计数为 Gregorian 历法日期 |
| *jdtojewish* | 转换一个 julian 天数为 Jewish 历法的日期 |
| *JDToJulian* | 转变一个 Julian Day 计数到 Julian 历法的日期 |
| *jdtounix* | 转变 Julian Day 计数为一个 Unix 时间戳 |
| *JewishToJD* | 转变一个 Jewish 历法的日期为一个 Julian Day 计数 |
| *join* | 别名 implode |
| *jpeg2wbmp* | 将 JPEG 图像文件转换为 WBMP 图像文件 |
| *json_decode* | 对 JSON 格式的字符串进行编码 |
| *json_encode* | 对变量进行 JSON 编码 |
| *json_last_error* | 返回最后发生的错误 |
| *json_last_error_msg* | Returns the error string of the last json_encode() or json_decode() call |
| *judy_type* | Return the type of a Judy array |
| *judy_version* | Return or print the current PHP Judy version |
| *JulianToJD* | 转变一个 Julian 历法的日期为 Julian Day 计数 |

# k

| *函数* | 说明 |
|---|---|
| *kadm5_chpass_principal* | Changes the principal's password |
| *kadm5_create_principal* | Creates a kerberos principal with the given parameters |
| *kadm5_delete_principal* | Deletes a kerberos principal |
| *kadm5_destroy* | Closes the connection to the admin server and releases all related resources |
| *kadm5_flush* | Flush all changes to the Kerberos database |
| *kadm5_get_policies* | Gets all policies from the Kerberos database |
| *kadm5_get_principal* | Gets the principal's entries from the Kerberos database |
| *kadm5_get_principals* | Gets all principals from the Kerberos database |
| *kadm5_init_with_password* | Opens a connection to the KADM5 library |
| *kadm5_modify_principal* | Modifies a kerberos principal with the given parameters |
| *key* | 从关联数组中取得键名 |

| | |
|---|---|
| *key_exists* | 别名 array_key_exists |
| *krsort* | 对数组按照键名逆向排序 |
| | |
| *ksort* | 对数组按照键名排序 |

## L

| *函数* | 说明 |
|---|---|
| *lcfirst* | 使一个字符串的第一个字符小写 |
| *lcg_value* | 组合线性同余发生器 |
| *lchgrp* | Changes group ownership of symlink |
| *lchown* | Changes user ownership of symlink |
| *ldap_8859_to_t61* | Translate 8859 characters to t61 characters |
| *ldap_add* | Add entries to LDAP directory |
| *ldap_bind* | 绑定 LDAP 目录 |
| *ldap_close* | 别名 ldap_unbind |
| *ldap_compare* | Compare value of attribute found in entry specified with DN |
| *ldap_connect* | Connect to an LDAP server |
| *ldap_control_paged_result* | Send LDAP pagination control |
| *ldap_control_paged_result_response* | Retrieve the LDAP pagination cookie |
| *ldap_count_entries* | Count the number of entries in a search |
| *ldap_delete* | Delete an entry from a directory |
| *ldap_dn2ufn* | Convert DN to User Friendly Naming format |
| *ldap_err2str* | Convert LDAP error number into string error message |
| *ldap_errno* | Return the LDAP error number of the last LDAP command |
| *ldap_error* | Return the LDAP error message of the last LDAP command |
| *ldap_escape* | Escape a string for use in an LDAP filter or DN |
| *ldap_explode_dn* | Splits DN into its component parts |
| *ldap_first_attribute* | Return first attribute |
| *ldap_first_entry* | Return first result id |
| *ldap_first_reference* | Return first reference |
| *ldap_free_result* | Free result memory |
| *ldap_get_attributes* | Get attributes from a search result entry |
| *ldap_get_dn* | Get the DN of a result entry |
| *ldap_get_entries* | Get all result entries |
| *ldap_get_option* | Get the current value for given option |
| *ldap_get_values* | Get all values from a result entry |
| *ldap_get_values_len* | Get all binary values from a result entry |
| *ldap_list* | Single-level search |
| *ldap_modify* | Modify an LDAP entry |
| *ldap_modify_batch* | Batch and execute modifications on an LDAP entry |
| *ldap_mod_add* | Add attribute values to current attributes |
| *ldap_mod_del* | Delete attribute values from current attributes |
| *ldap_mod_replace* | Replace attribute values with new ones |

| | |
|---|---|
| *ldap_next_attribute* | Get the next attribute in result |
| *ldap_next_entry* | Get next result entry |
| *ldap_next_reference* | Get next reference |
| *ldap_parse_reference* | Extract information from reference entry |
| *ldap_parse_result* | Extract information from result |
| *ldap_read* | Read an entry |
| *ldap_rename* | Modify the name of an entry |
| *ldap_sasl_bind* | Bind to LDAP directory using SASL |
| *ldap_search* | Search LDAP tree |
| *ldap_set_option* | Set the value of the given option |
| *ldap_set_rebind_proc* | Set a callback function to do re-binds on referral chasing |
| *ldap_sort* | Sort LDAP result entries |
| *ldap_start_tls* | Start TLS |
| *ldap_t61_to_8859* | Translate t61 characters to 8859 characters |
| *ldap_unbind* | Unbind from LDAP directory |
| *levenshtein* | 计算两个字符串之间的编辑距离 |
| *libxml_clear_errors* | Clear libxml error buffer |
| *libxml_disable_entity_loader* | Disable the ability to load external entities |
| *libxml_get_errors* | Retrieve array of errors |
| *libxml_get_last_error* | Retrieve last error from libxml |
| *libxml_set_external_entity_loader* | Changes the default external entity loader |
| *libxml_set_streams_context* | Set the streams context for the next libxml document load or write |
| *libxml_use_internal_errors* | Disable libxml errors and allow user to fetch error information as needed |
| *link* | 建立一个硬连接 |
| *linkinfo* | 获取一个连接的信息 |
| *list* | 把数组中的值赋给一些变量 |
| *localeconv* | Get numeric formatting information |
| *localtime* | 取得本地时间 |
| *log* | 自然对数 |
| *log1p* | 返回 log(1 + number)，甚至当 number 的值接近零也能计算出准确结果 |
| *log10* | 以 10 为底的对数 |
| *long2ip* | Converts an long integer address into a string in (IPv4) Internet standard dotted format |
| *lstat* | 给出一个文件或符号连接的信息 |
| *ltrim* | 删除字符串开头的空白字符（或其他字符） |
| *lzf_compress* | LZF compression |
| *lzf_decompress* | LZF decompression |
| *lzf_optimized_for* | Determines what LZF extension was optimized for |

# m

| *函数* | 说明 |
|---|---|
| *magic_quotes_runtime* | 别名 set_magic_quotes_runtime |

| | |
|---|---|
| *mail* | 发送邮件 |
| *mailparse_determine_best_xfer_encoding* | Gets the best way of encoding |
| *mailparse_msg_create* | Create a mime mail resource |
| *mailparse_msg_extract_part* | Extracts/decodes a message section |
| *mailparse_msg_extract_part_file* | Extracts/decodes a message section |
| *mailparse_msg_extract_whole_part_file* | Extracts a message section including headers without decoding the transfer encoding |
| *mailparse_msg_free* | Frees a MIME resource |
| *mailparse_msg_get_part* | Returns a handle on a given section in a mimemessage |
| *mailparse_msg_get_part_data* | Returns an associative array of info about the message |
| *mailparse_msg_get_structure* | Returns an array of mime section names in the supplied message |
| *mailparse_msg_parse* | Incrementally parse data into buffer |
| *mailparse_msg_parse_file* | Parses a file |
| *mailparse_rfc822_parse_addresses* | Parse RFC 822 compliant addresses |
| *mailparse_stream_encode* | Streams data from source file pointer, apply encoding and write to destfp |
| *mailparse_uudecode_all* | Scans the data from fp and extract each embedded uuencoded file |
| *main* | 虚拟的 main |
| *max* | 找出最大值 |
| *maxdb_affected_rows* | Gets the number of affected rows in a previous MaxDB operation |
| *maxdb_autocommit* | Turns on or off auto-commiting database modifications |
| *maxdb_bind_param* | 别名 maxdb_stmt_bind_param |
| *maxdb_bind_result* | 别名 maxdb_stmt_bind_result |
| *maxdb_change_user* | Changes the user of the specified database connection |
| *maxdb_character_set_name* | Returns the default character set for the database connection |
| *maxdb_client_encoding* | 别名 maxdb_character_set_name |
| *maxdb_close* | Closes a previously opened database connection |
| *maxdb_close_long_data* | 别名 maxdb_stmt_close_long_data |
| *maxdb_commit* | Commits the current transaction |
| *maxdb_connect* | Open a new connection to the MaxDB server |
| *maxdb_connect_errno* | Returns the error code from last connect call |
| *maxdb_connect_error* | Returns a string description of the last connect error |
| *maxdb_data_seek* | Adjusts the result pointer to an arbitary row in the result |
| *maxdb_debug* | Performs debugging operations |
| *maxdb_disable_reads_from_master* | Disable reads from master |
| *maxdb_disable_rpl_parse* | Disable RPL parse |
| *maxdb_dump_debug_info* | Dump debugging information into the log |
| *maxdb_embedded_connect* | Open a connection to an embedded MaxDB server |
| *maxdb_enable_reads_from_master* | Enable reads from master |
| *maxdb_enable_rpl_parse* | Enable RPL parse |
| *maxdb_errno* | Returns the error code for the most recent function call |
| *maxdb_error* | Returns a string description of the last error |
| *maxdb_escape_string* | 别名 maxdb_real_escape_string |
| *maxdb_execute* | 别名 maxdb_stmt_execute |

| | |
|---|---|
| *maxdb_fetch* | 别名 maxdb_stmt_fetch |
| *maxdb_fetch_array* | Fetch a result row as an associative, a numeric array, or both |
| *maxdb_fetch_assoc* | Fetch a result row as an associative array |
| *maxdb_fetch_field* | Returns the next field in the result set |
| *maxdb_fetch_fields* | Returns an array of resources representing the fields in a result set |
| *maxdb_fetch_field_direct* | Fetch meta-data for a single field |
| *maxdb_fetch_lengths* | Returns the lengths of the columns of the current row in the result set |
| *maxdb_fetch_object* | Returns the current row of a result set as an object |
| *maxdb_fetch_row* | Get a result row as an enumerated array |
| *maxdb_field_count* | Returns the number of columns for the most recent query |
| *maxdb_field_seek* | Set result pointer to a specified field offset |
| *maxdb_field_tell* | Get current field offset of a result pointer |
| *maxdb_free_result* | Frees the memory associated with a result |
| *maxdb_get_client_info* | Returns the MaxDB client version as a string |
| *maxdb_get_client_version* | Get MaxDB client info |
| *maxdb_get_host_info* | Returns a string representing the type of connection used |
| *maxdb_get_metadata* | 别名 maxdb_stmt_result_metadata |
| *maxdb_get_proto_info* | Returns the version of the MaxDB protocol used |
| *maxdb_get_server_info* | Returns the version of the MaxDB server |
| *maxdb_get_server_version* | Returns the version of the MaxDB server as an integer |
| *maxdb_info* | Retrieves information about the most recently executed query |
| *maxdb_init* | Initializes MaxDB and returns an resource for use with maxdb_real_connect |
| *maxdb_insert_id* | Returns the auto generated id used in the last query |
| *maxdb_kill* | Disconnects from a MaxDB server |
| *maxdb_master_query* | Enforce execution of a query on the master in a master/slave setup |
| *maxdb_more_results* | Check if there any more query results from a multi query |
| *maxdb_multi_query* | Performs a query on the database |
| *maxdb_next_result* | Prepare next result from multi_query |
| *maxdb_num_fields* | Get the number of fields in a result |
| *maxdb_num_rows* | Gets the number of rows in a result |
| *maxdb_options* | Set options |
| *maxdb_param_count* | 别名 maxdb_stmt_param_count |
| *maxdb_ping* | Pings a server connection, or tries to reconnect if the connection has gone down |
| *maxdb_prepare* | Prepare an SQL statement for execution |
| *maxdb_query* | Performs a query on the database |
| *maxdb_real_connect* | Opens a connection to a MaxDB server |
| *maxdb_real_escape_string* | Escapes special characters in a string for use in an SQL statement, taking into account the current charset of the connection |
| *maxdb_real_query* | Execute an SQL query |
| *maxdb_report* | Enables or disables internal report functions |
| *maxdb_rollback* | Rolls back current transaction |

| | |
|---|---|
| *maxdb_rpl_parse_enabled* | Check if RPL parse is enabled |
| *maxdb_rpl_probe* | RPL probe |
| *maxdb_rpl_query_type* | Returns RPL query type |
| *maxdb_select_db* | Selects the default database for database queries |
| *maxdb_send_long_data* | 别名 maxdb_stmt_send_long_data |
| *maxdb_send_query* | Send the query and return |
| *maxdb_server_end* | Shut down the embedded server |
| *maxdb_server_init* | Initialize embedded server |
| *maxdb_set_opt* | 别名 maxdb_options |
| *maxdb_sqlstate* | Returns the SQLSTATE error from previous MaxDB operation |
| *maxdb_ssl_set* | Used for establishing secure connections using SSL |
| *maxdb_stat* | Gets the current system status |
| *maxdb_stmt_affected_rows* | Returns the total number of rows changed, deleted, or inserted by the last executed statement |
| *maxdb_stmt_bind_param* | Binds variables to a prepared statement as parameters |
| *maxdb_stmt_bind_result* | Binds variables to a prepared statement for result storage |
| *maxdb_stmt_close* | Closes a prepared statement |
| *maxdb_stmt_close_long_data* | Ends a sequence of maxdb_stmt_send_long_data |
| *maxdb_stmt_data_seek* | Seeks to an arbitray row in statement result set |
| *maxdb_stmt_errno* | Returns the error code for the most recent statement call |
| *maxdb_stmt_error* | Returns a string description for last statement error |
| *maxdb_stmt_execute* | Executes a prepared Query |
| *maxdb_stmt_fetch* | Fetch results from a prepared statement into the bound variables |
| *maxdb_stmt_free_result* | Frees stored result memory for the given statement handle |
| *maxdb_stmt_init* | Initializes a statement and returns an resource for use with maxdb_stmt_prepare |
| *maxdb_stmt_num_rows* | Return the number of rows in statements result set |
| *maxdb_stmt_param_count* | Returns the number of parameter for the given statement |
| *maxdb_stmt_prepare* | Prepare an SQL statement for execution |
| *maxdb_stmt_reset* | Resets a prepared statement |
| *maxdb_stmt_result_metadata* | Returns result set metadata from a prepared statement |
| *maxdb_stmt_send_long_data* | Send data in blocks |
| *maxdb_stmt_sqlstate* | Returns SQLSTATE error from previous statement operation |
| *maxdb_stmt_store_result* | Transfers a result set from a prepared statement |
| *maxdb_store_result* | Transfers a result set from the last query |
| *maxdb_thread_id* | Returns the thread ID for the current connection |
| *maxdb_thread_safe* | Returns whether thread safety is given or not |
| *maxdb_use_result* | Initiate a result set retrieval |
| *maxdb_warning_count* | Returns the number of warnings from the last query for the given link |
| *mb_check_encoding* | 检查字符串在指定的编码里是否有效 |
| *mb_convert_case* | 对字符串进行大小写转换 |
| *mb_convert_encoding* | 转换字符的编码 |
| *mb_convert_kana* | Convert "kana" one from another ("zen-kaku", "han-kaku" and more) |

| | |
|---|---|
| *mb_convert_variables* | 转换一个或多个变量的字符编码 |
| *mb_decode_mimeheader* | 解码 MIME 头字段中的字符串 |
| *mb_decode_numericentity* | 根据 HTML 数字字符串解码成字符 |
| *mb_detect_encoding* | 检测字符的编码 |
| *mb_detect_order* | 设置/获取 字符编码的检测顺序 |
| *mb_encode_mimeheader* | 为 MIME 头编码字符串 |
| *mb_encode_numericentity* | Encode character to HTML numeric string reference |
| *mb_encoding_aliases* | Get aliases of a known encoding type |
| *mb_ereg* | Regular expression match with multibyte support |
| *mb_eregi* | Regular expression match ignoring case with multibyte support |
| *mb_eregi_replace* | Replace regular expression with multibyte support ignoring case |
| *mb_ereg_match* | Regular expression match for multibyte string |
| *mb_ereg_replace* | Replace regular expression with multibyte support |
| *mb_ereg_replace_callback* | Perform a regular expresssion seach and replace with multibyte support using a callback |
| *mb_ereg_search* | Multibyte regular expression match for predefined multibyte string |
| *mb_ereg_search_getpos* | Returns start point for next regular expression match |
| *mb_ereg_search_getregs* | Retrieve the result from the last multibyte regular expression match |
| *mb_ereg_search_init* | Setup string and regular expression for a multibyte regular expression match |
| *mb_ereg_search_pos* | Returns position and length of a matched part of the multibyte regular expression for a predefined multibyte string |
| *mb_ereg_search_regs* | Returns the matched part of a multibyte regular expression |
| *mb_ereg_search_setpos* | Set start point of next regular expression match |
| *mb_get_info* | 获取 mbstring 的内部设置 |
| *mb_http_input* | 检测 HTTP 输入字符编码 |
| *mb_http_output* | 设置/获取 HTTP 输出字符编码 |
| *mb_internal_encoding* | 设置/获取内部字符编码 |
| *mb_language* | 设置/获取当前的语言 |
| *mb_list_encodings* | 返回所有支持编码的数组 |
| *mb_output_handler* | 在输出缓冲中转换字符编码的回调函数 |
| *mb_parse_str* | 解析 GET/POST/COOKIE 数据并设置全局变量 |
| *mb_preferred_mime_name* | 获取 MIME 字符串 |
| *mb_regex_encoding* | Set/Get character encoding for multibyte regex |
| *mb_regex_set_options* | Set/Get the default options for mbregex functions |
| *mb_send_mail* | 发送编码过的邮件 |
| *mb_split* | 使用正则表达式分割多字节字符串 |
| *mb_strcut* | 获取字符的一部分 |
| *mb_strimwidth* | 获取按指定宽度截断的字符串 |
| *mb_stripos* | 大小写不敏感地查找字符串在另一个字符串中首次出现的位置 |
| *mb_stristr* | 大小写不敏感地查找字符串在另一个字符串里的首次出现 |
| *mb_strlen* | 获取字符串的长度 |
| *mb_strpos* | 查找字符串在另一个字符串中首次出现的位置 |

| | |
|---|---|
| *mb_strrchr* | 查找指定字符在另一个字符串中最后一次的出现 |
| *mb_strrichr* | 大小写不敏感地查找指定字符在另一个字符串中最后一次的出现 |
| *mb_strripos* | 大小写不敏感地在字符串中查找一个字符串最后出现的位置 |
| *mb_strrpos* | 查找字符串在一个字符串中最后出现的位置 |
| *mb_strstr* | 查找字符串在另一个字符串里的首次出现 |
| *mb_strtolower* | 使字符串小写 |
| *mb_strtoupper* | 使字符串大写 |
| *mb_strwidth* | 返回字符串的宽度 |
| *mb_substitute_character* | 设置/获取替代字符 |
| *mb_substr* | 获取字符串的部分 |
| *mb_substr_count* | 统计字符串出现的次数 |
| *mcrypt_cbc* | 以 CBC 模式加解密数据 |
| *mcrypt_cfb* | 以 CFB 模式加解密数据 |
| *mcrypt_create_iv* | 从随机源创建初始向量 |
| *mcrypt_decrypt* | 使用给定参数解密密文 |
| *mcrypt_ecb* | 已废弃：使用 ECB 模式加解密数据 |
| *mcrypt_encrypt* | 使用给定参数加密明文 |
| *mcrypt_enc_get_algorithms_name* | 返回打开的算法名称 |
| *mcrypt_enc_get_block_size* | 返回打开的算法的分组大小 |
| *mcrypt_enc_get_iv_size* | 返回打开的算法的初始向量大小 |
| *mcrypt_enc_get_key_size* | 返回打开的模式所能支持的最长密钥 |
| *mcrypt_enc_get_modes_name* | 返回打开的模式的名称 |
| *mcrypt_enc_get_supported_key_sizes* | 以数组方式返回打开的算法所支持的密钥长度 |
| *mcrypt_enc_is_block_algorithm* | 检测打开模式的算法是否为分组算法 |
| *mcrypt_enc_is_block_algorithm_mode* | 检测打开的模式是否支持分组加密 |
| *mcrypt_enc_is_block_mode* | 检测打开的模式是否以分组方式输出 |
| *mcrypt_enc_self_test* | 在打开的模块上进行自检 |
| *mcrypt_generic* | 加密数据 |
| *mcrypt_generic_deinit* | 对加密模块进行清理工作 |
| *mcrypt_generic_end* | 终止加密 |
| *mcrypt_generic_init* | 初始化加密所需的缓冲区 |
| *mcrypt_get_block_size* | 获得加密算法的分组大小 |
| *mcrypt_get_cipher_name* | 获取加密算法名称 |
| *mcrypt_get_iv_size* | 返回指定算法/模式组合的初始向量大小 |
| *mcrypt_get_key_size* | 获取指定加密算法的密钥大小 |
| *mcrypt_list_algorithms* | 获取支持的加密算法 |
| *mcrypt_list_modes* | 获取所支持的模式 |
| *mcrypt_module_close* | 关闭加密模块 |
| *mcrypt_module_get_algo_block_size* | 返回指定算法的分组大小 |
| *mcrypt_module_get_algo_key_size* | 获取打开模式所支持的最大密钥大小 |
| *mcrypt_module_get_supported_key_sizes* | 以数组形式返回打开的算法所支持的密钥大小 |
| *mcrypt_module_is_block_algorithm* | 检测指定算法是否为分组加密算法 |
| *mcrypt_module_is_block_algorithm_mode* | 返回指定模块是否是分组加密模式 |
| *mcrypt_module_is_block_mode* | 检测指定模式是否以分组方式输出 |

| | |
|---|---|
| *mcrypt_module_open* | 打开算法和模式对应的模块 |
| *mcrypt_module_self_test* | 在指定模块上执行自检 |
| *mcrypt_ofb* | 使用 OFB 模式加密/解密数据 |
| *md5* | 计算字符串的 MD5 散列值 |
| *md5_file* | 计算指定文件的 MD5 散列值 |
| *mdecrypt_generic* | 解密数据 |
| *memcache_debug* | 转换调试输出的开/关 |
| *memory_get_peak_usage* | 返回分配给 PHP 内存的峰值 |
| *memory_get_usage* | 返回分配给 PHP 的内存量 |
| *metaphone* | Calculate the metaphone key of a string |
| *method_exists* | 检查类的方法是否存在 |
| *mhash* | Computes hash |
| *mhash_count* | Gets the highest available hash ID |
| *mhash_get_block_size* | Gets the block size of the specified hash |
| *mhash_get_hash_name* | Gets the name of the specified hash |
| *mhash_keygen_s2k* | Generates a key |
| *microtime* | 返回当前 Unix 时间戳和微秒数 |
| *mime_content_type* | 检测文件的 MIME 类型（已废弃） |
| *min* | 找出最小值 |
| *ming_keypress* | Returns the action flag for keyPress(char) |
| *ming_setcubicthreshold* | Set cubic threshold |
| *ming_setscale* | Set the global scaling factor. |
| *ming_setswfcompression* | Sets the SWF output compression |
| *ming_useconstants* | Use constant pool |
| *ming_useswfversion* | Sets the SWF version |
| *mkdir* | 新建目录 |
| *mktime* | 取得一个日期的 Unix 时间戳 |
| *money_format* | Formats a number as a currency string |
| *MongoDB context options* | MongoDB context option listing |
| *MongoDB\BSON\fromJSON* | Returns the BSON representation of a JSON value |
| *MongoDB\BSON\fromPHP* | Returns the BSON representation of a PHP value |
| *MongoDB\BSON\toJSON* | Returns the JSON representation of a BSON value |
| *MongoDB\BSON\toPHP* | Returns the PHP representation of a BSON value |
| *move_uploaded_file* | 将上传的文件移动到新位置 |
| *mqseries_back* | MQSeries MQBACK |
| *mqseries_begin* | MQseries MQBEGIN |
| *mqseries_close* | MQSeries MQCLOSE |
| *mqseries_cmit* | MQSeries MQCMIT |
| *mqseries_conn* | MQSeries MQCONN |
| *mqseries_connx* | MQSeries MQCONNX |
| *mqseries_disc* | MQSeries MQDISC |
| *mqseries_get* | MQSeries MQGET |
| *mqseries_inq* | MQSeries MQINQ |
| *mqseries_open* | MQSeries MQOPEN |

| | |
|---|---|
| *mqseries_put* | MQSeries MQPUT |
| *mqseries_put1* | MQSeries MQPUT1 |
| *mqseries_set* | MQSeries MQSET |
| *mqseries_strerror* | Returns the error message corresponding to a result code (MQRC). |
| *msession_connect* | Connect to msession server |
| *msession_count* | Get session count |
| *msession_create* | Create a session |
| *msession_destroy* | Destroy a session |
| *msession_disconnect* | Close connection to msession server |
| *msession_find* | Find all sessions with name and value |
| *msession_get* | Get value from session |
| *msession_get_array* | Get array of msession variables |
| *msession_get_data* | Get data session unstructured data |
| *msession_inc* | Increment value in session |
| *msession_list* | List all sessions |
| *msession_listvar* | List sessions with variable |
| *msession_lock* | Lock a session |
| *msession_plugin* | Call an escape function within the msession personality plugin |
| *msession_randstr* | Get random string |
| *msession_set* | Set value in session |
| *msession_set_array* | Set msession variables from an array |
| *msession_set_data* | Set data session unstructured data |
| *msession_timeout* | Set/get session timeout |
| *msession_uniq* | Get unique id |
| *msession_unlock* | Unlock a session |
| *msg_get_queue* | Create or attach to a message queue |
| *msg_queue_exists* | Check whether a message queue exists |
| *msg_receive* | Receive a message from a message queue |
| *msg_remove_queue* | Destroy a message queue |
| *msg_send* | Send a message to a message queue |
| *msg_set_queue* | Set information in the message queue data structure |
| *msg_stat_queue* | Returns information from the message queue data structure |
| *msql* | Alias of msql_db_query |
| *msql_affected_rows* | Returns number of affected rows |
| *msql_close* | Close mSQL connection |
| *msql_connect* | Open mSQL connection |
| *msql_createdb* | 别名 msql_create_db |
| *msql_create_db* | Create mSQL database |
| *msql_data_seek* | Move internal row pointer |
| *msql_dbname* | 别名 msql_result |
| *msql_db_query* | Send mSQL query |
| *msql_drop_db* | Drop (delete) mSQL database |
| *msql_error* | Returns error message of last msql call |
| *msql_fetch_array* | Fetch row as array |

| | |
|---|---|
| *msql_fetch_field* | Get field information |
| *msql_fetch_object* | Fetch row as object |
| *msql_fetch_row* | Get row as enumerated array |
| *msql_fieldflags* | Alias of msql_field_flags |
| *msql_fieldlen* | Alias of msql_field_len |
| *msql_fieldname* | Alias of msql_field_name |
| *msql_fieldtable* | Alias of msql_field_table |
| *msql_fieldtype* | Alias of msql_field_type |
| *msql_field_flags* | Get field flags |
| *msql_field_len* | Get field length |
| *msql_field_name* | Get the name of the specified field in a result |
| *msql_field_seek* | Set field offset |
| *msql_field_table* | Get table name for field |
| *msql_field_type* | Get field type |
| *msql_free_result* | Free result memory |
| *msql_list_dbs* | List mSQL databases on server |
| *msql_list_fields* | List result fields |
| *msql_list_tables* | List tables in an mSQL database |
| *msql_numfields* | Alias of msql_num_fields |
| *msql_numrows* | Alias of msql_num_rows |
| *msql_num_fields* | Get number of fields in result |
| *msql_num_rows* | Get number of rows in result |
| *msql_pconnect* | Open persistent mSQL connection |
| *msql_query* | Send mSQL query |
| *msql_regcase* | Alias of sql_regcase |
| *msql_result* | Get result data |
| *msql_select_db* | Select mSQL database |
| *msql_tablename* | Alias of msql_result |
| *mssql_bind* | Adds a parameter to a stored procedure or a remote stored procedure |
| *mssql_close* | 关闭 MS SQL Server 链接 |
| *mssql_connect* | 打开 MS SQL server 链接 |
| *mssql_data_seek* | Moves internal row pointer |
| *mssql_execute* | Executes a stored procedure on a MS SQL server database |
| *mssql_fetch_array* | Fetch a result row as an associative array, a numeric array, or both |
| *mssql_fetch_assoc* | Returns an associative array of the current row in the result |
| *mssql_fetch_batch* | Returns the next batch of records |
| *mssql_fetch_field* | Get field information |
| *mssql_fetch_object* | Fetch row as object |
| *mssql_fetch_row* | Get row as enumerated array |
| *mssql_field_length* | Get the length of a field |
| *mssql_field_name* | Get the name of a field |
| *mssql_field_seek* | Seeks to the specified field offset |
| *mssql_field_type* | Gets the type of a field |

| | |
|---|---|
| *mssql_free_result* | Free result memory |
| *mssql_free_statement* | Free statement memory |
| *mssql_get_last_message* | Returns the last message from the server |
| *mssql_guid_string* | Converts a 16 byte binary GUID to a string |
| *mssql_init* | Initializes a stored procedure or a remote stored procedure |
| *mssql_min_error_severity* | Sets the minimum error severity |
| *mssql_min_message_severity* | Sets the minimum message severity |
| *mssql_next_result* | Move the internal result pointer to the next result |
| *mssql_num_fields* | Gets the number of fields in result |
| *mssql_num_rows* | Gets the number of rows in result |
| *mssql_pconnect* | Open persistent MS SQL connection |
| *mssql_query* | Send MS SQL query |
| *mssql_result* | Get result data |
| *mssql_rows_affected* | Returns the number of records affected by the query |
| *mssql_select_db* | Select MS SQL database |
| *mt_getrandmax* | 显示随机数的最大可能值 |
| *mt_rand* | 生成更好的随机数 |
| *mt_srand* | 播下一个更好的随机数发生器种子 |
| *mysqli_bind_param* | mysqli_stmt_bind_param 的别名 |
| *mysqli_bind_result* | mysqli_stmt_bind_result 的别名 |
| *mysqli_client_encoding* | mysqli_character_set_name 的别名 |
| *mysqli_disable_rpl_parse* | 禁用 RPL 解析 |
| *mysqli_enable_reads_from_master* | 开启从主机读取 |
| *mysqli_enable_rpl_parse* | 开启 RPL 解析 |
| *mysqli_escape_string* | 别名 mysqli_real_escape_string |
| *mysqli_execute* | mysqli_stmt_execute 的别名 |
| *mysqli_fetch* | mysqli_stmt_fetch 的别名。 |
| *mysqli_get_cache_stats* | 返回客户端 Zval 缓存统计信息 |
| *mysqli_get_client_stats* | Returns client per-process statistics |
| *mysqli_get_client_version* | Returns the MySQL client version as an integer |
| *mysqli_get_links_stats* | Return information about open and cached links |
| *mysqli_get_metadata* | mysqli_stmt_result_metadata 的别名 |
| *mysqli_master_query* | 在主/从机制中强制在主机中执行一个查询 |
| *mysqli_param_count* | mysqli_stmt_param_count 的别名 |
| *mysqli_report* | 开启或禁用（Mysql）内部（错误）报告函数 |
| *mysqli_rpl_parse_enabled* | 检查是否开启了 RPL 解析 |
| *mysqli_rpl_probe* | RPL 探测 |
| *mysqli_send_long_data* | mysqli_stmt_send_long_data 的别名 |
| *mysqli_set_opt* | mysqli_options 的别名 |
| *mysqli_slave_query* | 在主/从机制中强制在从机上执行一个查询 |
| *mysqlnd_memcache_get_config* | Returns information about the plugin configuration |
| *mysqlnd_memcache_set* | Associate a MySQL connection with a Memcache connection |
| *mysqlnd_ms_dump_servers* | Returns a list of currently configured servers |
| *mysqlnd_ms_fabric_select_global* | Switch to global sharding server for a given table |

| | |
|---|---|
| *mysqlnd_ms_fabric_select_shard* | Switch to shard |
| *mysqlnd_ms_get_last_gtid* | 返回最后的全局同步 ID (GTID) |
| *mysqlnd_ms_get_last_used_connection* | Returns an array which describes the last used connection |
| *mysqlnd_ms_get_stats* | Returns query distribution and connection statistics |
| *mysqlnd_ms_match_wild* | Finds whether a table name matches a wildcard pattern or not |
| *mysqlnd_ms_query_is_select* | 查询给定的 SQL 会发送给 master、slave 还是最后使用的 MySQL server 执行。 |
| *mysqlnd_ms_set_qos* | Sets the quality of service needed from the cluster |
| *mysqlnd_ms_set_user_pick_server* | Sets a callback for user-defined read/write splitting |
| *mysqlnd_ms_xa_begin* | Starts a distributed/XA transaction among MySQL servers |
| *mysqlnd_ms_xa_commit* | Commits a distributed/XA transaction among MySQL servers |
| *mysqlnd_ms_xa_gc* | Garbage collects unfinished XA transactions after severe errors |
| *mysqlnd_ms_xa_rollback* | Rolls back a distributed/XA transaction among MySQL servers |
| *mysqlnd_qc_clear_cache* | Flush all cache contents |
| *mysqlnd_qc_get_available_handlers* | Returns a list of available storage handler |
| *mysqlnd_qc_get_cache_info* | Returns information on the current handler, the number of cache entries and cache entries, if available |
| *mysqlnd_qc_get_core_stats* | Statistics collected by the core of the query cache |
| *mysqlnd_qc_get_normalized_query_trace_log* | Returns a normalized query trace log for each query inspected by the query cache |
| *mysqlnd_qc_get_query_trace_log* | Returns a backtrace for each query inspected by the query cache |
| *mysqlnd_qc_set_cache_condition* | Set conditions for automatic caching |
| *mysqlnd_qc_set_is_select* | Installs a callback which decides whether a statement is cached |
| *mysqlnd_qc_set_storage_handler* | Change current storage handler |
| *mysqlnd_qc_set_user_handlers* | Sets the callback functions for a user-defined procedural storage handler |
| *mysqlnd_uh_convert_to_mysqlnd* | Converts a MySQL connection handle into a mysqlnd connection handle |
| *mysqlnd_uh_set_connection_proxy* | Installs a proxy for mysqlnd connections |
| *mysqlnd_uh_set_statement_proxy* | Installs a proxy for mysqlnd statements |
| *mysql_affected_rows* | 取得前一次 MySQL 操作所影响的记录行数 |
| *mysql_client_encoding* | 返回字符集的名称 |
| *mysql_close* | 关闭 MySQL 连接 |
| *mysql_connect* | 打开一个到 MySQL 服务器的连接 |
| *mysql_create_db* | 新建一个 MySQL 数据库 |
| *mysql_data_seek* | 移动内部结果的指针 |
| *mysql_db_name* | 取得结果数据 |
| *mysql_db_query* | 发送一条 MySQL 查询 |
| *mysql_drop_db* | 丢弃（删除）一个 MySQL 数据库 |
| *mysql_errno* | 返回上一个 MySQL 操作中的错误信息的数字编码 |
| *mysql_error* | 返回上一个 MySQL 操作产生的文本错误信息 |
| *mysql_escape_string* | 转义一个字符串用于 mysql_query |
| *mysql_fetch_array* | 从结果集中取得一行作为关联数组，或数字数组，或二者兼有 |
| *mysql_fetch_assoc* | 从结果集中取得一行作为关联数组 |
| *mysql_fetch_field* | 从结果集中取得列信息并作为对象返回 |

| mysql_fetch_lengths | 取得结果集中每个输出的长度 |
|---|---|
| mysql_fetch_object | 从结果集中取得一行作为对象 |
| mysql_fetch_row | 从结果集中取得一行作为枚举数组 |
| mysql_field_flags | 从结果中取得和指定字段关联的标志 |
| mysql_field_len | 返回指定字段的长度 |
| mysql_field_name | 取得结果中指定字段的字段名 |
| mysql_field_seek | 将结果集中的指针设定为制定的字段偏移量 |
| mysql_field_table | 取得指定字段所在的表名 |
| mysql_field_type | 取得结果集中指定字段的类型 |
| mysql_free_result | 释放结果内存 |
| mysql_get_client_info | 取得 MySQL 客户端信息 |
| mysql_get_host_info | 取得 MySQL 主机信息 |
| mysql_get_proto_info | 取得 MySQL 协议信息 |
| mysql_get_server_info | 取得 MySQL 服务器信息 |
| mysql_info | 取得最近一条查询的信息 |
| mysql_insert_id | 取得上一步 INSERT 操作产生的 ID |
| mysql_list_dbs | 列出 MySQL 服务器中所有的数据库 |
| mysql_list_fields | 列出 MySQL 结果中的字段 |
| mysql_list_processes | 列出 MySQL 进程 |
| mysql_list_tables | 列出 MySQL 数据库中的表 |
| mysql_num_fields | 取得结果集中字段的数目 |
| mysql_num_rows | 取得结果集中行的数目 |
| mysql_pconnect | 打开一个到 MySQL 服务器的持久连接 |
| mysql_ping | Ping 一个服务器连接，如果没有连接则重新连接 |
| mysql_query | 发送一条 MySQL 查询 |
| mysql_real_escape_string | 转义 SQL 语句中使用的字符串中的特殊字符,并考虑到连接的当前字符集 |
| mysql_result | 取得结果数据 |
| mysql_select_db | 选择 MySQL 数据库 |
| mysql_set_charset | 设置客户端的字符集 |
| mysql_stat | 取得当前系统状态 |
| mysql_tablename | 取得表名 |
| mysql_thread_id | 返回当前线程的 ID |
| mysql_unbuffered_query | 向 MySQL 发送一条 SQL 查询，并不获取和缓存结果的行 |
| m_checkstatus | Check to see if a transaction has completed |
| m_completeauthorizations | Number of complete authorizations in queue, returning an array of their identifiers |
| m_connect | Establish the connection to MCVE |
| m_connectionerror | Get a textual representation of why a connection failed |
| m_deletetrans | Delete specified transaction from MCVE_CONN structure |
| m_destroyconn | Destroy the connection and MCVE_CONN structure |
| m_destroyengine | Free memory associated with IP/SSL connectivity |
| m_getcell | Get a specific cell from a comma delimited response by column name |
| m_getcellbynum | Get a specific cell from a comma delimited response by column |

| | number |
|---|---|
| *m_getcommadelimited* | Get the RAW comma delimited data returned from MCVE |
| *m_getheader* | Get the name of the column in a comma-delimited response |
| *m_initconn* | Create and initialize an MCVE_CONN structure |
| *m_initengine* | Ready the client for IP/SSL Communication |
| *m_iscommadelimited* | Checks to see if response is comma delimited |
| *m_maxconntimeout* | The maximum amount of time the API will attempt a connection to MCVE |
| *m_monitor* | Perform communication with MCVE (send/receive data) Non-blocking |
| *m_numcolumns* | Number of columns returned in a comma delimited response |
| *m_numrows* | Number of rows returned in a comma delimited response |
| *m_parsecommadelimited* | Parse the comma delimited response so m_getcell, etc will work |
| *m_responsekeys* | Returns array of strings which represents the keys that can be used for response parameters on this transaction |
| *m_responseparam* | Get a custom response parameter |
| *m_returnstatus* | Check to see if the transaction was successful |
| *m_setblocking* | Set blocking/non-blocking mode for connection |
| *m_setdropfile* | Set the connection method to Drop-File |
| *m_setip* | Set the connection method to IP |
| *m_setssl* | Set the connection method to SSL |
| *m_setssl_cafile* | Set SSL CA (Certificate Authority) file for verification of server certificate |
| *m_setssl_files* | Set certificate key files and certificates if server requires client certificate verification |
| *m_settimeout* | Set maximum transaction time (per trans) |
| *m_sslcert_gen_hash* | Generate hash for SSL client certificate verification |
| *m_transactionssent* | Check to see if outgoing buffer is clear |
| *m_transinqueue* | Number of transactions in client-queue |
| *m_transkeyval* | Add key/value pair to a transaction. Replaces deprecated transparam() |
| *m_transnew* | Start a new transaction |
| *m_transsend* | Finalize and send the transaction |
| *m_uwait* | Wait x microsecs |
| *m_validateidentifier* | Whether or not to validate the passed identifier on any transaction it is passed to |
| *m_verifyconnection* | Set whether or not to PING upon connect to verify connection |
| | |
| *m_verifysslcert* | Set whether or not to verify the server ssl certificate |

# n

| *函数* | 说明 |
|---|---|
| *natcasesort* | 用"自然排序"算法对数组进行不区分大小写字母的排序 |
| *natsort* | 用"自然排序"算法对数组排序 |

| | |
|---|---|
| *ncurses_addch* | Add character at current position and advance cursor |
| *ncurses_addchnstr* | Add attributed string with specified length at current position |
| *ncurses_addchstr* | Add attributed string at current position |
| *ncurses_addnstr* | Add string with specified length at current position |
| *ncurses_addstr* | Output text at current position |
| *ncurses_assume_default_colors* | Define default colors for color 0 |
| *ncurses_attroff* | Turn off the given attributes |
| *ncurses_attron* | Turn on the given attributes |
| *ncurses_attrset* | Set given attributes |
| *ncurses_baudrate* | Returns baudrate of terminal |
| *ncurses_beep* | Let the terminal beep |
| *ncurses_bkgd* | Set background property for terminal screen |
| *ncurses_bkgdset* | Control screen background |
| *ncurses_border* | Draw a border around the screen using attributed characters |
| *ncurses_bottom_panel* | Moves a visible panel to the bottom of the stack |
| *ncurses_can_change_color* | Checks if terminal color definitions can be changed |
| *ncurses_cbreak* | Switch off input buffering |
| *ncurses_clear* | Clear screen |
| *ncurses_clrtobot* | Clear screen from current position to bottom |
| *ncurses_clrtoeol* | Clear screen from current position to end of line |
| *ncurses_color_content* | Retrieves RGB components of a color |
| *ncurses_color_set* | Set active foreground and background colors |
| *ncurses_curs_set* | Set cursor state |
| *ncurses_define_key* | Define a keycode |
| *ncurses_def_prog_mode* | Saves terminals (program) mode |
| *ncurses_def_shell_mode* | Saves terminals (shell) mode |
| *ncurses_delay_output* | Delay output on terminal using padding characters |
| *ncurses_delch* | Delete character at current position, move rest of line left |
| *ncurses_deleteln* | Delete line at current position, move rest of screen up |
| *ncurses_delwin* | Delete a ncurses window |
| *ncurses_del_panel* | Remove panel from the stack and delete it (but not the associated window) |
| *ncurses_doupdate* | Write all prepared refreshes to terminal |
| *ncurses_echo* | Activate keyboard input echo |
| *ncurses_echochar* | Single character output including refresh |
| *ncurses_end* | Stop using ncurses, clean up the screen |
| *ncurses_erase* | Erase terminal screen |
| *ncurses_erasechar* | Returns current erase character |
| *ncurses_filter* | Set LINES for iniscr() and newterm() to 1 |
| *ncurses_flash* | Flash terminal screen (visual bell) |
| *ncurses_flushinp* | Flush keyboard input buffer |
| *ncurses_getch* | Read a character from keyboard |
| *ncurses_getmaxyx* | Returns the size of a window |
| *ncurses_getmouse* | Reads mouse event |

| | |
|---|---|
| *ncurses_getyx* | Returns the current cursor position for a window |
| *ncurses_halfdelay* | Put terminal into halfdelay mode |
| *ncurses_has_colors* | Checks if terminal has color capabilities |
| *ncurses_has_ic* | Check for insert- and delete-capabilities |
| *ncurses_has_il* | Check for line insert- and delete-capabilities |
| *ncurses_has_key* | Check for presence of a function key on terminal keyboard |
| *ncurses_hide_panel* | Remove panel from the stack, making it invisible |
| *ncurses_hline* | Draw a horizontal line at current position using an attributed character and max. n characters long |
| *ncurses_inch* | Get character and attribute at current position |
| *ncurses_init* | Initialize ncurses |
| *ncurses_init_color* | Define a terminal color |
| *ncurses_init_pair* | Define a color pair |
| *ncurses_insch* | Insert character moving rest of line including character at current position |
| *ncurses_insdelln* | Insert lines before current line scrolling down (negative numbers delete and scroll up) |
| *ncurses_insertln* | Insert a line, move rest of screen down |
| *ncurses_insstr* | Insert string at current position, moving rest of line right |
| *ncurses_instr* | Reads string from terminal screen |
| *ncurses_isendwin* | Ncurses is in endwin mode, normal screen output may be performed |
| *ncurses_keyok* | Enable or disable a keycode |
| *ncurses_keypad* | Turns keypad on or off |
| *ncurses_killchar* | Returns current line kill character |
| *ncurses_longname* | Returns terminals description |
| *ncurses_meta* | Enables/Disable 8-bit meta key information |
| *ncurses_mouseinterval* | Set timeout for mouse button clicks |
| *ncurses_mousemask* | Sets mouse options |
| *ncurses_mouse_trafo* | Transforms coordinates |
| *ncurses_move* | Move output position |
| *ncurses_move_panel* | Moves a panel so that its upper-left corner is at [startx, starty] |
| *ncurses_mvaddch* | Move current position and add character |
| *ncurses_mvaddchnstr* | Move position and add attributed string with specified length |
| *ncurses_mvaddchstr* | Move position and add attributed string |
| *ncurses_mvaddnstr* | Move position and add string with specified length |
| *ncurses_mvaddstr* | Move position and add string |
| *ncurses_mvcur* | Move cursor immediately |
| *ncurses_mvdelch* | Move position and delete character, shift rest of line left |
| *ncurses_mvgetch* | Move position and get character at new position |
| *ncurses_mvhline* | Set new position and draw a horizontal line using an attributed character and max. n characters long |
| *ncurses_mvinch* | Move position and get attributed character at new position |
| *ncurses_mvvline* | Set new position and draw a vertical line using an attributed character and max. n characters long |
| *ncurses_mvwaddstr* | Add string at new position in window |

| | |
|---|---|
| *ncurses_napms* | Sleep |
| *ncurses_newpad* | Creates a new pad (window) |
| *ncurses_newwin* | Create a new window |
| *ncurses_new_panel* | Create a new panel and associate it with window |
| *ncurses_nl* | Translate newline and carriage return / line feed |
| *ncurses_nocbreak* | Switch terminal to cooked mode |
| *ncurses_noecho* | Switch off keyboard input echo |
| *ncurses_nonl* | Do not translate newline and carriage return / line feed |
| *ncurses_noqiflush* | Do not flush on signal characters |
| *ncurses_noraw* | Switch terminal out of raw mode |
| *ncurses_pair_content* | Retrieves foreground and background colors of a color pair |
| *ncurses_panel_above* | Returns the panel above panel |
| *ncurses_panel_below* | Returns the panel below panel |
| *ncurses_panel_window* | Returns the window associated with panel |
| *ncurses_pnoutrefresh* | Copies a region from a pad into the virtual screen |
| *ncurses_prefresh* | Copies a region from a pad into the virtual screen |
| *ncurses_putp* | Apply padding information to the string and output it |
| *ncurses_qiflush* | Flush on signal characters |
| *ncurses_raw* | Switch terminal into raw mode |
| *ncurses_refresh* | Refresh screen |
| *ncurses_replace_panel* | Replaces the window associated with panel |
| *ncurses_resetty* | Restores saved terminal state |
| *ncurses_reset_prog_mode* | Resets the prog mode saved by def_prog_mode |
| *ncurses_reset_shell_mode* | Resets the shell mode saved by def_shell_mode |
| *ncurses_savetty* | Saves terminal state |
| *ncurses_scrl* | Scroll window content up or down without changing current position |
| *ncurses_scr_dump* | Dump screen content to file |
| *ncurses_scr_init* | Initialize screen from file dump |
| *ncurses_scr_restore* | Restore screen from file dump |
| *ncurses_scr_set* | Inherit screen from file dump |
| *ncurses_show_panel* | Places an invisible panel on top of the stack, making it visible |
| *ncurses_slk_attr* | Returns current soft label key attribute |
| *ncurses_slk_attroff* | Turn off the given attributes for soft function-key labels |
| *ncurses_slk_attron* | Turn on the given attributes for soft function-key labels |
| *ncurses_slk_attrset* | Set given attributes for soft function-key labels |
| *ncurses_slk_clear* | Clears soft labels from screen |
| *ncurses_slk_color* | Sets color for soft label keys |
| *ncurses_slk_init* | Initializes soft label key functions |
| *ncurses_slk_noutrefresh* | Copies soft label keys to virtual screen |
| *ncurses_slk_refresh* | Copies soft label keys to screen |
| *ncurses_slk_restore* | Restores soft label keys |
| *ncurses_slk_set* | Sets function key labels |
| *ncurses_slk_touch* | Forces output when ncurses_slk_noutrefresh is performed |
| *ncurses_standend* | Stop using 'standout' attribute |

| | |
|---|---|
| *ncurses_standout* | Start using 'standout' attribute |
| *ncurses_start_color* | Initializes color functionality |
| *ncurses_termattrs* | Returns a logical OR of all attribute flags supported by terminal |
| *ncurses_termname* | Returns terminals (short)-name |
| *ncurses_timeout* | Set timeout for special key sequences |
| *ncurses_top_panel* | Moves a visible panel to the top of the stack |
| *ncurses_typeahead* | Specify different filedescriptor for typeahead checking |
| *ncurses_ungetch* | Put a character back into the input stream |
| *ncurses_ungetmouse* | Pushes mouse event to queue |
| *ncurses_update_panels* | Refreshes the virtual screen to reflect the relations between panels in the stack |
| *ncurses_use_default_colors* | Assign terminal default colors to color id -1 |
| *ncurses_use_env* | Control use of environment information about terminal size |
| *ncurses_use_extended_names* | Control use of extended names in terminfo descriptions |
| *ncurses_vidattr* | Display the string on the terminal in the video attribute mode |
| *ncurses_vline* | Draw a vertical line at current position using an attributed character and max. n characters long |
| *ncurses_waddch* | Adds character at current position in a window and advance cursor |
| *ncurses_waddstr* | Outputs text at current postion in window |
| *ncurses_wattroff* | Turns off attributes for a window |
| *ncurses_wattron* | Turns on attributes for a window |
| *ncurses_wattrset* | Set the attributes for a window |
| *ncurses_wborder* | Draws a border around the window using attributed characters |
| *ncurses_wclear* | Clears window |
| *ncurses_wcolor_set* | Sets windows color pairings |
| *ncurses_werase* | Erase window contents |
| *ncurses_wgetch* | Reads a character from keyboard (window) |
| *ncurses_whline* | Draws a horizontal line in a window at current position using an attributed character and max. n characters long |
| *ncurses_wmouse_trafo* | Transforms window/stdscr coordinates |
| *ncurses_wmove* | Moves windows output position |
| *ncurses_wnoutrefresh* | Copies window to virtual screen |
| *ncurses_wrefresh* | Refresh window on terminal screen |
| *ncurses_wstandend* | End standout mode for a window |
| *ncurses_wstandout* | Enter standout mode for a window |
| *ncurses_wvline* | Draws a vertical line in a window at current position using an attributed character and max. n characters long |
| *newt_bell* | Send a beep to the terminal |
| *newt_button* | Create a new button |
| *newt_button_bar* | This function returns a grid containing the buttons created. |
| *newt_centered_window* | Open a centered window of the specified size |
| *newt_checkbox* | 说明 |
| *newt_checkbox_get_value* | Retreives value of checkox resource |
| *newt_checkbox_set_flags* | Configures checkbox resource |
| *newt_checkbox_set_value* | Sets the value of the checkbox |

| | |
|---|---|
| *newt_checkbox_tree* | 说明 |
| *newt_checkbox_tree_add_item* | Adds new item to the checkbox tree |
| *newt_checkbox_tree_find_item* | Finds an item in the checkbox tree |
| *newt_checkbox_tree_get_current* | Returns checkbox tree selected item |
| *newt_checkbox_tree_get_entry_value* | 说明 |
| *newt_checkbox_tree_get_multi_selection* | 说明 |
| *newt_checkbox_tree_get_selection* | 说明 |
| *newt_checkbox_tree_multi* | 说明 |
| *newt_checkbox_tree_set_current* | 说明 |
| *newt_checkbox_tree_set_entry* | 说明 |
| *newt_checkbox_tree_set_entry_value* | 说明 |
| *newt_checkbox_tree_set_width* | 说明 |
| *newt_clear_key_buffer* | Discards the contents of the terminal's input buffer without waiting for additional input |
| *newt_cls* | 说明 |
| *newt_compact_button* | 说明 |
| *newt_component_add_callback* | 说明 |
| *newt_component_takes_focus* | 说明 |
| *newt_create_grid* | 说明 |
| *newt_cursor_off* | 说明 |
| *newt_cursor_on* | 说明 |
| *newt_delay* | 说明 |
| *newt_draw_form* | 说明 |
| *newt_draw_root_text* | Displays the string text at the position indicated |
| *newt_entry* | 说明 |
| *newt_entry_get_value* | 说明 |
| *newt_entry_set* | 说明 |
| *newt_entry_set_filter* | 说明 |
| *newt_entry_set_flags* | 说明 |
| *newt_finished* | Uninitializes newt interface |
| *newt_form* | Create a form |
| *newt_form_add_component* | Adds a single component to the form |
| *newt_form_add_components* | Add several components to the form |
| *newt_form_add_hot_key* | 说明 |
| *newt_form_destroy* | Destroys a form |
| *newt_form_get_current* | 说明 |
| *newt_form_run* | Runs a form |
| *newt_form_set_background* | 说明 |
| *newt_form_set_height* | 说明 |
| *newt_form_set_size* | 说明 |
| *newt_form_set_timer* | 说明 |
| *newt_form_set_width* | 说明 |
| *newt_form_watch_fd* | 说明 |
| *newt_get_screen_size* | Fills in the passed references with the current size of the |

| | terminal |
|---|---|
| *newt_grid_add_components_to_form* | 说明 |
| *newt_grid_basic_window* | 说明 |
| *newt_grid_free* | 说明 |
| *newt_grid_get_size* | 说明 |
| *newt_grid_h_close_stacked* | 说明 |
| *newt_grid_h_stacked* | 说明 |
| *newt_grid_place* | 说明 |
| *newt_grid_set_field* | 说明 |
| *newt_grid_simple_window* | 说明 |
| *newt_grid_v_close_stacked* | 说明 |
| *newt_grid_v_stacked* | 说明 |
| *newt_grid_wrapped_window* | 说明 |
| *newt_grid_wrapped_window_at* | 说明 |
| *newt_init* | Initialize newt |
| *newt_label* | 说明 |
| *newt_label_set_text* | 说明 |
| *newt_listbox* | 说明 |
| *newt_listbox_append_entry* | 说明 |
| *newt_listbox_clear* | 说明 |
| *newt_listbox_clear_selection* | 说明 |
| *newt_listbox_delete_entry* | 说明 |
| *newt_listbox_get_current* | 说明 |
| *newt_listbox_get_selection* | 说明 |
| *newt_listbox_insert_entry* | 说明 |
| *newt_listbox_item_count* | 说明 |
| *newt_listbox_select_item* | 说明 |
| *newt_listbox_set_current* | 说明 |
| *newt_listbox_set_current_by_key* | 说明 |
| *newt_listbox_set_data* | 说明 |
| *newt_listbox_set_entry* | 说明 |
| *newt_listbox_set_width* | 说明 |
| *newt_listitem* | 说明 |
| *newt_listitem_get_data* | 说明 |
| *newt_listitem_set* | 说明 |
| *newt_open_window* | Open a window of the specified size and position |
| *newt_pop_help_line* | Replaces the current help line with the one from the stack |
| *newt_pop_window* | Removes the top window from the display |
| *newt_push_help_line* | Saves the current help line on a stack, and displays the new line |
| *newt_radiobutton* | 说明 |
| *newt_radio_get_current* | 说明 |
| *newt_redraw_help_line* | 说明 |
| *newt_reflow_text* | 说明 |
| *newt_refresh* | Updates modified portions of the screen |

| | |
|---|---|
| *newt_resize_screen* | 说明 |
| *newt_resume* | Resume using the newt interface after calling newt_suspend |
| *newt_run_form* | Runs a form |
| *newt_scale* | 说明 |
| *newt_scale_set* | 说明 |
| *newt_scrollbar_set* | 说明 |
| *newt_set_help_callback* | 说明 |
| *newt_set_suspend_callback* | Set a callback function which gets invoked when user presses the suspend key |
| *newt_suspend* | Tells newt to return the terminal to its initial state |
| *newt_textbox* | 说明 |
| *newt_textbox_get_num_lines* | 说明 |
| *newt_textbox_reflowed* | 说明 |
| *newt_textbox_set_height* | 说明 |
| *newt_textbox_set_text* | 说明 |
| *newt_vertical_scrollbar* | 说明 |
| *newt_wait_for_key* | Doesn't return until a key has been pressed |
| *newt_win_choice* | 说明 |
| *newt_win_entries* | 说明 |
| *newt_win_menu* | 说明 |
| *newt_win_message* | 说明 |
| *newt_win_messagev* | 说明 |
| *newt_win_ternary* | 说明 |
| *next* | 将数组中的内部指针向前移动一位 |
| *ngettext* | Plural version of gettext |
| *nl2br* | 在字符串所有新行之前插入 HTML 换行标记 |
| *nl_langinfo* | Query language and locale information |
| *nsapi_request_headers* | Fetch all HTTP request headers |
| *nsapi_response_headers* | Fetch all HTTP response headers |
| *nsapi_virtual* | Perform an NSAPI sub-request |
| *nthmac* | Obtain a nthmac key (needs 2 arguments) |
| | |
| *number_format* | 以千位分隔符方式格式化一个数字 |

# O

| *函数* | 说明 |
|---|---|
| *oauth_get_sbs* | 生成一个签名字符基串 |
| *oauth_urlencode* | 将 URI 编码为 RFC 3986 规范 |
| *ob_clean* | 清空（擦掉）输出缓冲区 |
| *ob_deflatehandler* | Deflate output handler |
| *ob_end_clean* | 清空（擦除）缓冲区并关闭输出缓冲 |
| *ob_end_flush* | 冲刷出（送出）输出缓冲区内容并关闭缓冲 |

| | |
|---|---|
| *ob_etaghandler* | ETag output handler |
| *ob_flush* | 冲刷出（送出）输出缓冲区中的内容 |
| *ob_get_clean* | 得到当前缓冲区的内容并删除当前输出缓。 |
| *ob_get_contents* | 返回输出缓冲区的内容 |
| *ob_get_flush* | 刷出（送出）缓冲区内容，以字符串形式返回内容，并关闭输出缓冲区。 |
| *ob_get_length* | 返回输出缓冲区内容的长度 |
| *ob_get_level* | 返回输出缓冲机制的嵌套级别 |
| *ob_get_status* | 得到所有输出缓冲区的状态 |
| *ob_gzhandler* | 在 ob_start 中使用的用来压缩输出缓冲区中内容的回调函数。ob_start callback function to gzip output buffer |
| *ob_iconv_handler* | 以输出缓冲处理程序转换字符编码 |
| *ob_implicit_flush* | 打开/关闭绝对刷送 |
| *ob_inflatehandler* | Inflate output handler |
| *ob_list_handlers* | 列出所有使用中的输出处理程序。 |
| *ob_start* | 打开输出控制缓冲 |
| *ob_tidyhandler* | ob_start callback function to repair the buffer |
| *ocibindbyname* | 别名 oci_bind_by_name |
| *ocicancel* | 别名 oci_cancel |
| *ocicolumnisnull* | 别名 oci_field_is_null |
| *ocicolumnname* | 别名 oci_field_name |
| *ocicolumnprecision* | 别名 oci_field_precision |
| *ocicolumnscale* | 别名 oci_field_scale |
| *ocicolumnsize* | 别名 oci_field_size |
| *ocicolumntype* | 别名 oci_field_type |
| *ocicolumntyperaw* | 别名 oci_field_type_raw |
| *ocicommit* | 别名 oci_commit |
| *ocidefinebyname* | 别名 oci_define_by_name |
| *ocierror* | 别名 oci_error |
| *ociexecute* | 别名 oci_execute |
| *ocifetch* | 别名 oci_fetch |
| *ocifetchinto* | Obsolete variant of oci_fetch_array, oci_fetch_object, oci_fetch_assoc and oci_fetch_row |
| *ocifetchstatement* | 别名 oci_fetch_all |
| *ocifreecursor* | 别名 oci_free_statement |
| *ocifreestatement* | 别名 oci_free_statement |
| *ociinternaldebug* | 别名 oci_internal_debug |
| *ocilogoff* | 别名 oci_close |
| *ocilogon* | 别名 oci_connect |
| *ocinewcollection* | 别名 oci_new_collection |
| *ocinewcursor* | 别名 oci_new_cursor |
| *ocinewdescriptor* | 别名 oci_new_descriptor |
| *ocinlogon* | 别名 oci_new_connect |
| *ocinumcols* | 别名 oci_num_fields |
| *ociparse* | 别名 oci_parse |

| | |
|---|---|
| *ociplogon* | 别名 oci_pconnect |
| *ociresult* | 别名 oci_result |
| *ocirollback* | 别名 oci_rollback |
| *ocirowcount* | 别名 oci_num_rows |
| *ociserverversion* | 别名 oci_server_version |
| *ocisetprefetch* | 别名 oci_set_prefetch |
| *ocistatementtype* | 别名 oci_statement_type |
| *oci_bind_array_by_name* | Binds a PHP array to an Oracle PL/SQL array parameter |
| *oci_bind_by_name* | 绑定一个 PHP 变量到一个 Oracle 位置标志符 |
| *oci_cancel* | 中断游标读取数据 |
| *oci_client_version* | Returns the Oracle client library version |
| *oci_close* | 关闭 Oracle 连接 |
| *oci_commit* | 提交未执行的事务处理 |
| *oci_connect* | 建立一个到 Oracle 服务器的连接 |
| *oci_define_by_name* | 在 SELECT 中使用 PHP 变量作为定义的步骤 |
| *oci_error* | 返回上一个错误 |
| *oci_execute* | 执行一条语句 |
| *oci_fetch* | Fetches the next row into result-buffer |
| *oci_fetch_all* | 获取结果数据的所有行到一个数组 |
| *oci_fetch_array* | Returns the next row from a query as an associative or numeric array |
| *oci_fetch_assoc* | Returns the next row from a query as an associative array |
| *oci_fetch_object* | Returns the next row from a query as an object |
| *oci_fetch_row* | Returns the next row from a query as a numeric array |
| *oci_field_is_null* | 检查字段是否为 NULL |
| *oci_field_name* | 返回字段名 |
| *oci_field_precision* | 返回字段精度 |
| *oci_field_scale* | 返回字段范围 |
| *oci_field_size* | 返回字段大小 |
| *oci_field_type* | 返回字段的数据类型 |
| *oci_field_type_raw* | 返回字段的原始 Oracle 数据类型 |
| *oci_free_descriptor* | Frees a descriptor |
| *oci_free_statement* | 释放关联于语句或游标的所有资源 |
| *oci_get_implicit_resultset* | Returns the next child statement resource from a parent statement resource that has Oracle Database 12c Implicit Result Sets |
| *oci_internal_debug* | 打开或关闭内部调试输出 |
| *oci_lob_copy* | Copies large object |
| *oci_lob_is_equal* | Compares two LOB/FILE locators for equality |
| *oci_new_collection* | 分配新的 collection 对象 |
| *oci_new_connect* | 建定一个到 Oracle 服务器的新连接 |
| *oci_new_cursor* | 分配并返回一个新的游标（语句句柄） |
| *oci_new_descriptor* | 初始化一个新的空 LOB 或 FILE 描述符 |
| *oci_num_fields* | 返回结果列的数目 |
| *oci_num_rows* | 返回语句执行后受影响的行数 |
| *oci_parse* | 配置 Oracle 语句预备执行 |

| | |
|---|---|
| *oci_password_change* | 修改 Oracle 用户的密码 |
| *oci_pconnect* | 使用一个持久连接连到 Oracle 数据库 |
| *oci_result* | 返回所取得行中字段的值 |
| *oci_rollback* | 回滚未提交的事务 |
| *oci_server_version* | 返回服务器版本信息 |
| *oci_set_action* | Sets the action name |
| *oci_set_client_identifier* | Sets the client identifier |
| *oci_set_client_info* | Sets the client information |
| *oci_set_edition* | Sets the database edition |
| *oci_set_module_name* | Sets the module name |
| *oci_set_prefetch* | 设置预提取行数 |
| *oci_statement_type* | 返回 OCI 语句的类型 |
| *octdec* | 八进制转换为十进制 |
| *odbc_autocommit* | Toggle autocommit behaviour |
| *odbc_binmode* | Handling of binary column data |
| *odbc_close* | Close an ODBC connection |
| *odbc_close_all* | Close all ODBC connections |
| *odbc_columnprivileges* | Lists columns and associated privileges for the given table |
| *odbc_columns* | Lists the column names in specified tables |
| *odbc_commit* | Commit an ODBC transaction |
| *odbc_connect* | Connect to a datasource |
| *odbc_cursor* | Get cursorname |
| *odbc_data_source* | Returns information about a current connection |
| *odbc_do* | 别名 odbc_exec |
| *odbc_error* | Get the last error code |
| *odbc_errormsg* | Get the last error message |
| *odbc_exec* | Prepare and execute an SQL statement |
| *odbc_execute* | Execute a prepared statement |
| *odbc_fetch_array* | Fetch a result row as an associative array |
| *odbc_fetch_into* | Fetch one result row into array |
| *odbc_fetch_object* | Fetch a result row as an object |
| *odbc_fetch_row* | Fetch a row |
| *odbc_field_len* | Get the length (precision) of a field |
| *odbc_field_name* | Get the columnname |
| *odbc_field_num* | Return column number |
| *odbc_field_precision* | 别名 odbc_field_len |
| *odbc_field_scale* | Get the scale of a field |
| *odbc_field_type* | Datatype of a field |
| *odbc_foreignkeys* | Retrieves a list of foreign keys |
| *odbc_free_result* | Free resources associated with a result |
| *odbc_gettypeinfo* | Retrieves information about data types supported by the data source |
| *odbc_longreadlen* | Handling of LONG columns |
| *odbc_next_result* | Checks if multiple results are available |

| | |
|---|---|
| *odbc_num_fields* | Number of columns in a result |
| *odbc_num_rows* | Number of rows in a result |
| *odbc_pconnect* | Open a persistent database connection |
| *odbc_prepare* | Prepares a statement for execution |
| *odbc_primarykeys* | Gets the primary keys for a table |
| *odbc_procedurecolumns* | Retrieve information about parameters to procedures |
| *odbc_procedures* | Get the list of procedures stored in a specific data source |
| *odbc_result* | Get result data |
| *odbc_result_all* | Print result as HTML table |
| *odbc_rollback* | Rollback a transaction |
| *odbc_setoption* | Adjust ODBC settings |
| *odbc_specialcolumns* | Retrieves special columns |
| *odbc_statistics* | Retrieve statistics about a table |
| *odbc_tableprivileges* | Lists tables and the privileges associated with each table |
| *odbc_tables* | Get the list of table names stored in a specific data source |
| *opcache_compile_file* | 无需运行，即可编译并缓存 PHP 脚本 |
| *opcache_get_configuration* | 获取缓存的配置信息 |
| *opcache_get_status* | 获取缓存的状态信息 |
| *opcache_invalidate* | 废除脚本缓存 |
| *opcache_is_script_cached* | Tells whether a script is cached in OPCache |
| *opcache_reset* | 重置字节码缓存的内容 |
| *openal_buffer_create* | Generate OpenAL buffer |
| *openal_buffer_data* | Load a buffer with data |
| *openal_buffer_destroy* | Destroys an OpenAL buffer |
| *openal_buffer_get* | Retrieve an OpenAL buffer property |
| *openal_buffer_loadwav* | Load a .wav file into a buffer |
| *openal_context_create* | Create an audio processing context |
| *openal_context_current* | Make the specified context current |
| *openal_context_destroy* | Destroys a context |
| *openal_context_process* | Process the specified context |
| *openal_context_suspend* | Suspend the specified context |
| *openal_device_close* | Close an OpenAL device |
| *openal_device_open* | Initialize the OpenAL audio layer |
| *openal_listener_get* | Retrieve a listener property |
| *openal_listener_set* | Set a listener property |
| *openal_source_create* | Generate a source resource |
| *openal_source_destroy* | Destroy a source resource |
| *openal_source_get* | Retrieve an OpenAL source property |
| *openal_source_pause* | Pause the source |
| *openal_source_play* | Start playing the source |
| *openal_source_rewind* | Rewind the source |
| *openal_source_set* | Set source property |
| *openal_source_stop* | Stop playing the source |
| *openal_stream* | Begin streaming on a source |

| | |
|---|---|
| *opendir* | 打开目录句柄 |
| *openlog* | Open connection to system logger |
| *openssl_cipher_iv_length* | Gets the cipher iv length |
| *openssl_csr_export* | Exports a CSR as a string |
| *openssl_csr_export_to_file* | Exports a CSR to a file |
| *openssl_csr_get_public_key* | Returns the public key of a CERT |
| *openssl_csr_get_subject* | Returns the subject of a CERT |
| *openssl_csr_new* | Generates a CSR |
| *openssl_csr_sign* | Sign a CSR with another certificate (or itself) and generate a certificate |
| *openssl_decrypt* | Decrypts data |
| *openssl_dh_compute_key* | Computes shared secret for public value of remote DH key and local DH key |
| *openssl_digest* | Computes a digest |
| *openssl_encrypt* | Encrypts data |
| *openssl_error_string* | Return openSSL error message |
| *openssl_free_key* | Free key resource |
| *openssl_get_cert_locations* | Retrieve the available certificate locations |
| *openssl_get_cipher_methods* | Gets available cipher methods |
| *openssl_get_md_methods* | Gets available digest methods |
| *openssl_get_privatekey* | 别名 openssl_pkey_get_private |
| *openssl_get_publickey* | 别名 openssl_pkey_get_public |
| *openssl_open* | Open sealed data |
| *openssl_pbkdf2* | Generates a PKCS5 v2 PBKDF2 string, defaults to SHA-1 |
| *openssl_pkcs7_decrypt* | Decrypts an S/MIME encrypted message |
| *openssl_pkcs7_encrypt* | Encrypt an S/MIME message |
| *openssl_pkcs7_sign* | Sign an S/MIME message |
| *openssl_pkcs7_verify* | Verifies the signature of an S/MIME signed message |
| *openssl_pkcs12_export* | Exports a PKCS#12 Compatible Certificate Store File to variable. |
| *openssl_pkcs12_export_to_file* | Exports a PKCS#12 Compatible Certificate Store File |
| *openssl_pkcs12_read* | Parse a PKCS#12 Certificate Store into an array |
| *openssl_pkey_export* | Gets an exportable representation of a key into a string |
| *openssl_pkey_export_to_file* | Gets an exportable representation of a key into a file |
| *openssl_pkey_free* | Frees a private key |
| *openssl_pkey_get_details* | Returns an array with the key details |
| *openssl_pkey_get_private* | Get a private key |
| *openssl_pkey_get_public* | Extract public key from certificate and prepare it for use |
| *openssl_pkey_new* | Generates a new private key |
| *openssl_private_decrypt* | Decrypts data with private key |
| *openssl_private_encrypt* | Encrypts data with private key |
| *openssl_public_decrypt* | Decrypts data with public key |
| *openssl_public_encrypt* | Encrypts data with public key |
| *openssl_random_pseudo_bytes* | Generate a pseudo-random string of bytes |
| *openssl_seal* | Seal (encrypt) data |

| | |
|---|---|
| *openssl_sign* | Generate signature |
| *openssl_spki_export* | Exports a valid PEM formatted public key signed public key and challenge |
| *openssl_spki_export_challenge* | Exports the challenge assoicated with a signed public key and challenge |
| *openssl_spki_new* | Generate a new signed public key and challenge |
| *openssl_spki_verify* | Verifies a signed public key and challenge |
| *openssl_verify* | Verify signature |
| *openssl_x509_checkpurpose* | Verifies if a certificate can be used for a particular purpose |
| *openssl_x509_check_private_key* | Checks if a private key corresponds to a certificate |
| *openssl_x509_export* | Exports a certificate as a string |
| *openssl_x509_export_to_file* | Exports a certificate to file |
| *openssl_x509_fingerprint* | Calculates the fingerprint, or digest, of a given X.509 certificate |
| *openssl_x509_free* | Free certificate resource |
| *openssl_x509_parse* | Parse an X509 certificate and return the information as an array |
| *openssl_x509_read* | Parse an X.509 certificate and return a resource identifier for it |
| *ord* | 返回字符的 ASCII 码值 |
| *output_add_rewrite_var* | 添加 URL 重写器的值（Add URL rewriter values） |
| *output_reset_rewrite_vars* | 重设 URL 重写器的值（Reset URL rewriter values） |
| *override_function* | Overrides built-in functions |

# p

| 函数 | 说明 |
|---|---|
| *pack* | Pack data into binary string |
| *parsekit_compile_file* | Compile a PHP file and return the resulting op array |
| *parsekit_compile_string* | Compile a string of PHP code and return the resulting op array |
| *parsekit_func_arginfo* | Return information regarding function argument(s) |
| *parse_ini_file* | 解析一个配置文件 |
| *parse_ini_string* | Parse a configuration string |
| *parse_str* | 将字符串解析成多个变量 |
| *parse_url* | 解析 URL，返回其组成部分 |
| *passthru* | 执行外部程序并且显示原始输出 |
| *password_get_info* | Returns information about the given hash |
| *password_hash* | Creates a password hash |
| *password_needs_rehash* | Checks if the given hash matches the given options |
| *password_verify* | Verifies that a password matches a hash |
| *pathinfo* | 返回文件路径的信息 |
| *pclose* | 关闭进程文件指针 |
| *pcntl_alarm* | 为进程设置一个 alarm 闹钟信号 |
| *pcntl_errno* | 别名 pcntl_strerror |
| *pcntl_exec* | 在当前进程空间执行指定程序 |

| | |
|---|---|
| *pcntl_fork* | 在当前进程当前位置产生分支（子进程）。译注：fork 是创建了一个子进程，父进程和子进程 都从 fork 的位置开始向下继续执行，不同的是父进程执行过程中，得到的 fork 返回值为子进程 号，而子进程得到的是 0。 |
| *pcntl_getpriority* | 获取任意进程的优先级 |
| *pcntl_get_last_error* | Retrieve the error number set by the last pcntl function which failed |
| *pcntl_setpriority* | 修改任意进程的优先级 |
| *pcntl_signal* | 安装一个信号处理器 |
| *pcntl_signal_dispatch* | 调用等待信号的处理器 |
| *pcntl_sigprocmask* | 设置或检索阻塞信号 |
| *pcntl_sigtimedwait* | 带超时机制的信号等待 |
| *pcntl_sigwaitinfo* | 等待信号 |
| *pcntl_strerror* | Retrieve the system error message associated with the given errno |
| *pcntl_wait* | 等待或返回 fork 的子进程状态 |
| *pcntl_waitpid* | 等待或返回 fork 的子进程状态 |
| *pcntl_wexitstatus* | 返回一个中断的子进程的返回代码 |
| *pcntl_wifexited* | 检查状态代码是否代表一个正常的退出。 |
| *pcntl_wifsignaled* | 检查子进程状态码是否代表由于某个信号而中断 |
| *pcntl_wifstopped* | 检查子进程当前是否已经停止 |
| *pcntl_wstopsig* | 返回导致子进程停止的信号 |
| *pcntl_wtermsig* | 返回导致子进程中断的信号 |
| *PDF_activate_item* | Activate structure element or other content item |
| *PDF_add_annotation* | Add annotation [deprecated] |
| *PDF_add_bookmark* | Add bookmark for current page [deprecated] |
| *PDF_add_launchlink* | Add launch annotation for current page [deprecated] |
| *PDF_add_locallink* | Add link annotation for current page [deprecated] |
| *PDF_add_nameddest* | Create named destination |
| *PDF_add_note* | Set annotation for current page [deprecated] |
| *PDF_add_outline* | Add bookmark for current page [deprecated] |
| *PDF_add_pdflink* | Add file link annotation for current page [deprecated] |
| *PDF_add_table_cell* | Add a cell to a new or existing table |
| *PDF_add_textflow* | Create Textflow or add text to existing Textflow |
| *PDF_add_thumbnail* | Add thumbnail for current page |
| *PDF_add_weblink* | Add weblink for current page [deprecated] |
| *PDF_arc* | Draw a counterclockwise circular arc segment |
| *PDF_arcn* | Draw a clockwise circular arc segment |
| *PDF_attach_file* | Add file attachment for current page [deprecated] |
| *PDF_begin_document* | Create new PDF file |
| *PDF_begin_font* | Start a Type 3 font definition |
| *PDF_begin_glyph* | Start glyph definition for Type 3 font |
| *PDF_begin_item* | Open structure element or other content item |
| *PDF_begin_layer* | Start layer |
| *PDF_begin_page* | Start new page [deprecated] |
| *PDF_begin_page_ext* | Start new page |

| | |
|---|---|
| *PDF_begin_pattern* | Start pattern definition |
| *PDF_begin_template* | Start template definition [deprecated] |
| *PDF_begin_template_ext* | Start template definition |
| *PDF_circle* | Draw a circle |
| *PDF_clip* | Clip to current path |
| *PDF_close* | Close pdf resource [deprecated] |
| *PDF_closepath* | Close current path |
| *PDF_closepath_fill_stroke* | Close, fill and stroke current path |
| *PDF_closepath_stroke* | Close and stroke path |
| *PDF_close_image* | Close image |
| *PDF_close_pdi* | Close the input PDF document [deprecated] |
| *PDF_close_pdi_page* | Close the page handle |
| *PDF_concat* | Concatenate a matrix to the CTM |
| *PDF_continue_text* | Output text in next line |
| *PDF_create_3dview* | Create 3D view |
| *PDF_create_action* | Create action for objects or events |
| *PDF_create_annotation* | Create rectangular annotation |
| *PDF_create_bookmark* | Create bookmark |
| *PDF_create_field* | Create form field |
| *PDF_create_fieldgroup* | Create form field group |
| *PDF_create_gstate* | Create graphics state object |
| *PDF_create_pvf* | Create PDFlib virtual file |
| *PDF_create_textflow* | Create textflow object |
| *PDF_curveto* | Draw Bezier curve |
| *PDF_define_layer* | Create layer definition |
| *PDF_delete* | Delete PDFlib object |
| *PDF_delete_pvf* | Delete PDFlib virtual file |
| *PDF_delete_table* | Delete table object |
| *PDF_delete_textflow* | Delete textflow object |
| *PDF_encoding_set_char* | Add glyph name and/or Unicode value |
| *PDF_endpath* | End current path |
| *PDF_end_document* | Close PDF file |
| *PDF_end_font* | Terminate Type 3 font definition |
| *PDF_end_glyph* | Terminate glyph definition for Type 3 font |
| *PDF_end_item* | Close structure element or other content item |
| *PDF_end_layer* | Deactivate all active layers |
| *PDF_end_page* | Finish page |
| *PDF_end_page_ext* | Finish page |
| *PDF_end_pattern* | Finish pattern |
| *PDF_end_template* | Finish template |
| *PDF_fill* | Fill current path |
| *PDF_fill_imageblock* | Fill image block with variable data |
| *PDF_fill_pdfblock* | Fill PDF block with variable data |
| *PDF_fill_stroke* | Fill and stroke path |

| | |
|---|---|
| *PDF_fill_textblock* | Fill text block with variable data |
| *PDF_findfont* | Prepare font for later use [deprecated] |
| *PDF_fit_image* | Place image or template |
| *PDF_fit_pdi_page* | Place imported PDF page |
| *PDF_fit_table* | Place table on page |
| *PDF_fit_textflow* | Format textflow in rectangular area |
| *PDF_fit_textline* | Place single line of text |
| *PDF_get_apiname* | Get name of unsuccessfull API function |
| *PDF_get_buffer* | Get PDF output buffer |
| *PDF_get_errmsg* | Get error text |
| *PDF_get_errnum* | Get error number |
| *PDF_get_font* | Get font [deprecated] |
| *PDF_get_fontname* | Get font name [deprecated] |
| *PDF_get_fontsize* | Font handling [deprecated] |
| *PDF_get_image_height* | Get image height [deprecated] |
| *PDF_get_image_width* | Get image width [deprecated] |
| *PDF_get_majorversion* | Get major version number [deprecated] |
| *PDF_get_minorversion* | Get minor version number [deprecated] |
| *PDF_get_parameter* | Get string parameter |
| *PDF_get_pdi_parameter* | Get PDI string parameter [deprecated] |
| *PDF_get_pdi_value* | Get PDI numerical parameter [deprecated] |
| *PDF_get_value* | Get numerical parameter |
| *PDF_info_font* | Query detailed information about a loaded font |
| *PDF_info_matchbox* | Query matchbox information |
| *PDF_info_table* | Retrieve table information |
| *PDF_info_textflow* | Query textflow state |
| *PDF_info_textline* | Perform textline formatting and query metrics |
| *PDF_initgraphics* | Reset graphic state |
| *PDF_lineto* | Draw a line |
| *PDF_load_3ddata* | Load 3D model |
| *PDF_load_font* | Search and prepare font |
| *PDF_load_iccprofile* | Search and prepare ICC profile |
| *PDF_load_image* | Open image file |
| *PDF_makespotcolor* | Make spot color |
| *PDF_moveto* | Set current point |
| *PDF_new* | Create PDFlib object |
| *PDF_open_ccitt* | Open raw CCITT image [deprecated] |
| *PDF_open_file* | Create PDF file [deprecated] |
| *PDF_open_gif* | Open GIF image [deprecated] |
| *PDF_open_image* | Use image data [deprecated] |
| *PDF_open_image_file* | Read image from file [deprecated] |
| *PDF_open_jpeg* | Open JPEG image [deprecated] |
| *PDF_open_memory_image* | Open image created with PHP's image functions [not supported] |
| *PDF_open_pdi* | Open PDF file [deprecated] |

| | |
|---|---|
| *PDF_open_pdi_document* | Prepare a pdi document |
| *PDF_open_pdi_page* | Prepare a page |
| *PDF_open_tiff* | Open TIFF image [deprecated] |
| *PDF_pcos_get_number* | Get value of pCOS path with type number or boolean |
| *PDF_pcos_get_stream* | Get contents of pCOS path with type stream, fstream, or string |
| *PDF_pcos_get_string* | Get value of pCOS path with type name, string, or boolean |
| *PDF_place_image* | Place image on the page [deprecated] |
| *PDF_place_pdi_page* | Place PDF page [deprecated] |
| *PDF_process_pdi* | Process imported PDF document |
| *PDF_rect* | Draw rectangle |
| *PDF_restore* | Restore graphics state |
| *PDF_resume_page* | Resume page |
| *PDF_rotate* | Rotate coordinate system |
| *PDF_save* | Save graphics state |
| *PDF_scale* | Scale coordinate system |
| *PDF_setcolor* | Set fill and stroke color |
| *PDF_setdash* | Set simple dash pattern |
| *PDF_setdashpattern* | Set dash pattern |
| *PDF_setflat* | Set flatness |
| *PDF_setfont* | Set font |
| *PDF_setgray* | Set color to gray [deprecated] |
| *PDF_setgray_fill* | Set fill color to gray [deprecated] |
| *PDF_setgray_stroke* | Set stroke color to gray [deprecated] |
| *PDF_setlinecap* | Set linecap parameter |
| *PDF_setlinejoin* | Set linejoin parameter |
| *PDF_setlinewidth* | Set line width |
| *PDF_setmatrix* | Set current transformation matrix |
| *PDF_setmiterlimit* | Set miter limit |
| *PDF_setpolydash* | Set complicated dash pattern [deprecated] |
| *PDF_setrgbcolor* | Set fill and stroke rgb color values [deprecated] |
| *PDF_setrgbcolor_fill* | Set fill rgb color values [deprecated] |
| *PDF_setrgbcolor_stroke* | Set stroke rgb color values [deprecated] |
| *PDF_set_border_color* | Set border color of annotations [deprecated] |
| *PDF_set_border_dash* | Set border dash style of annotations [deprecated] |
| *PDF_set_border_style* | Set border style of annotations [deprecated] |
| *PDF_set_char_spacing* | Set character spacing [deprecated] |
| *PDF_set_duration* | Set duration between pages [deprecated] |
| *PDF_set_gstate* | Activate graphics state object |
| *PDF_set_horiz_scaling* | Set horizontal text scaling [deprecated] |
| *PDF_set_info* | Fill document info field |
| *PDF_set_info_author* | Fill the author document info field [deprecated] |
| *PDF_set_info_creator* | Fill the creator document info field [deprecated] |
| *PDF_set_info_keywords* | Fill the keywords document info field [deprecated] |
| *PDF_set_info_subject* | Fill the subject document info field [deprecated] |

| | |
|---|---|
| *PDF_set_info_title* | Fill the title document info field [deprecated] |
| *PDF_set_layer_dependency* | Define relationships among layers |
| *PDF_set_leading* | Set distance between text lines [deprecated] |
| *PDF_set_parameter* | Set string parameter |
| *PDF_set_text_matrix* | Set text matrix [deprecated] |
| *PDF_set_text_pos* | Set text position |
| *PDF_set_text_rendering* | Determine text rendering [deprecated] |
| *PDF_set_text_rise* | Set text rise [deprecated] |
| *PDF_set_value* | Set numerical parameter |
| *PDF_set_word_spacing* | Set spacing between words [deprecated] |
| *PDF_shading* | Define blend |
| *PDF_shading_pattern* | Define shading pattern |
| *PDF_shfill* | Fill area with shading |
| *PDF_show* | Output text at current position |
| *PDF_show_boxed* | Output text in a box [deprecated] |
| *PDF_show_xy* | Output text at given position |
| *PDF_skew* | Skew the coordinate system |
| *PDF_stringwidth* | Return width of text |
| *PDF_stroke* | Stroke path |
| *PDF_suspend_page* | Suspend page |
| *PDF_translate* | Set origin of coordinate system |
| *PDF_utf8_to_utf16* | Convert string from UTF-8 to UTF-16 |
| *PDF_utf16_to_utf8* | Convert string from UTF-16 to UTF-8 |
| *PDF_utf32_to_utf16* | Convert string from UTF-32 to UTF-16 |
| *PDO_4D DSN* | Connecting to 4D SQL server |
| *PDO_CUBRID DSN* | Connecting to CUBRID databases |
| *PDO_DBLIB DSN* | Connecting to Microsoft SQL Server and Sybase databases |
| *PDO_FIREBIRD DSN* | Connecting to Firebird databases |
| *PDO_IBM DSN* | Connecting to IBM databases |
| *PDO_INFORMIX DSN* | Connecting to Informix databases |
| *PDO_MYSQL DSN* | Connecting to MySQL databases |
| *PDO_OCI DSN* | Connecting to Oracle databases |
| *PDO_ODBC DSN* | Connecting to ODBC or DB2 databases |
| *PDO_PGSQL DSN* | Connecting to PostgreSQL databases |
| *PDO_SQLITE DSN* | Connecting to SQLite databases |
| *PDO_SQLSRV DSN* | Connecting to MS SQL Server and SQL Azure databases |
| *pfsockopen* | 打开一个持久的网络连接或者 Unix 套接字连接。 |
| *pg_affected_rows* | 返回受影响的记录数目 |
| *pg_cancel_query* | 取消异步查询 |
| *pg_client_encoding* | 取得客户端编码方式 |
| *pg_close* | 关闭一个 PostgreSQL 连接 |
| *pg_connect* | 打开一个 PostgreSQL 连接 |
| *pg_connection_busy* | 获知连接是否为忙 |
| *pg_connection_reset* | 重置连接（再次连接） |

| | |
|---|---|
| *pg_connection_status* | 获得连接状态 |
| *pg_connect_poll* | Poll the status of an in-progress asynchronous PostgreSQL connection attempt. |
| *pg_consume_input* | Reads input on the connection |
| *pg_convert* | 将关联的数组值转换为适合 SQL 语句的格式。 |
| *pg_copy_from* | 根据数组将记录插入表中 |
| *pg_copy_to* | 将一个表拷贝到数组中 |
| *pg_dbname* | 获得数据库名 |
| *pg_delete* | 删除记录 |
| *pg_end_copy* | 与 PostgreSQL 后端同步 |
| *pg_escape_bytea* | 转义 bytea 类型的二进制数据 |
| *pg_escape_identifier* | Escape a identifier for insertion into a text field |
| *pg_escape_literal* | Escape a literal for insertion into a text field |
| *pg_escape_string* | 转义 text/char 类型的字符串 |
| *pg_execute* | Sends a request to execute a prepared statement with given parameters, and waits for the result. |
| *pg_fetch_all* | 从结果中提取所有行作为一个数组 |
| *pg_fetch_all_columns* | Fetches all rows in a particular result column as an array |
| *pg_fetch_array* | 提取一行作为数组 |
| *pg_fetch_assoc* | 提取一行作为关联数组 |
| *pg_fetch_object* | 提取一行作为对象 |
| *pg_fetch_result* | 从结果资源中返回值 |
| *pg_fetch_row* | 提取一行作为枚举数组 |
| *pg_field_is_null* | 测试字段是否为 NULL |
| *pg_field_name* | 返回字段的名字 |
| *pg_field_num* | 返回字段的编号 |
| *pg_field_prtlen* | 返回打印出来的长度 |
| *pg_field_size* | 返回指定字段占用内部存储空间的大小 |
| *pg_field_table* | Returns the name or oid of the tables field |
| *pg_field_type* | 返回相应字段的类型名称 |
| *pg_field_type_oid* | Returns the type ID (OID) for the corresponding field number |
| *pg_flush* | Flush outbound query data on the connection |
| *pg_free_result* | 释放查询结果占用的内存 |
| *pg_get_notify* | Ping 数据库连接 |
| *pg_get_pid* | Ping 数据库连接 |
| *pg_get_result* | 取得异步查询结果 |
| *pg_host* | 返回和某连接关联的主机名 |
| *pg_insert* | 将数组插入到表中 |
| *pg_last_error* | 得到某连接的最后一条错误信息 |
| *pg_last_notice* | 返回 PostgreSQL 服务器最新一条公告信息 |
| *pg_last_oid* | 返回上一个对象的 oid |
| *pg_lo_close* | 关闭一个大型对象 |
| *pg_lo_create* | 新建一个大型对象 |
| *pg_lo_export* | 将大型对象导出到文件 |

| | |
|---|---|
| *pg_lo_import* | 将文件导入为大型对象 |
| *pg_lo_open* | 打开一个大型对象 |
| *pg_lo_read* | 从大型对象中读入数据 |
| *pg_lo_read_all* | 读入整个大型对象并直接发送给浏览器 |
| *pg_lo_seek* | 移动大型对象中的指针 |
| *pg_lo_tell* | 返回大型对象的当前指针位置 |
| *pg_lo_truncate* | Truncates a large object |
| *pg_lo_unlink* | 删除一个大型对象 |
| *pg_lo_write* | 向大型对象写入数据 |
| *pg_meta_data* | 获得表的元数据 |
| *pg_num_fields* | 返回字段的数目 |
| *pg_num_rows* | 返回行的数目 |
| *pg_options* | 获得和连接有关的选项 |
| *pg_parameter_status* | Looks up a current parameter setting of the server. |
| *pg_pconnect* | 打开一个持久的 PostgreSQL 连接 |
| *pg_ping* | Ping 数据库连接 |
| *pg_port* | 返回该连接的端口号 |
| *pg_prepare* | Submits a request to create a prepared statement with the given parameters, and waits for completion. |
| *pg_put_line* | 向 PostgreSQL 后端发送以 NULL 结尾的字符串 |
| *pg_query* | 执行查询 |
| *pg_query_params* | Submits a command to the server and waits for the result, with the ability to pass parameters separately from the SQL command text. |
| *pg_result_error* | 获得查询结果的错误信息 |
| *pg_result_error_field* | Returns an individual field of an error report. |
| *pg_result_seek* | 在结果资源中设定内部行偏移量 |
| *pg_result_status* | 获得查询结果的状态 |
| *pg_select* | 选择记录 |
| *pg_send_execute* | Sends a request to execute a prepared statement with given parameters, without waiting for the result(s). |
| *pg_send_prepare* | Sends a request to create a prepared statement with the given parameters, without waiting for completion. |
| *pg_send_query* | 发送异步查询 |
| *pg_send_query_params* | Submits a command and separate parameters to the server without waiting for the result(s). |
| *pg_set_client_encoding* | 设定客户端编码 |
| *pg_set_error_verbosity* | Determines the verbosity of messages returned by pg_last_error and pg_result_error. |
| *pg_socket* | Get a read only handle to the socket underlying a PostgreSQL connection |
| *pg_trace* | 启动一个 PostgreSQL 连接的追踪功能 |
| *pg_transaction_status* | Returns the current in-transaction status of the server. |
| *pg_tty* | 返回该连接的 tty 号 |
| *pg_unescape_bytea* | 取消 bytea 类型中的字符串转义 |
| *pg_untrace* | 关闭 PostgreSQL 连接的追踪功能 |

| | |
|---|---|
| *pg_update* | 更新表 |
| *pg_version* | Returns an array with client, protocol and server version (when available) |
| *PharException* | The PharException class provides a phar-specific exception class for try/catch blocks. |
| *Phar 上下文（context）选项* | Phar 上下文（context）选项列表 |
| *phpcredits* | 打印 PHP 贡献者名单 |
| *phpinfo* | 输出关于 PHP 配置的信息 |
| *phpversion* | 获取当前的 PHP 版本 |
| *php_check_syntax* | 检查 PHP 的语法（并执行）指定的文件 |
| *php_ini_loaded_file* | 取得已加载的 php.ini 文件的路径 |
| *php_ini_scanned_files* | 返回从额外 ini 目录里解析的 .ini 文件列表 |
| *php_logo_guid* | 获取 logo 的 guid |
| *php_sapi_name* | 返回 web 服务器和 PHP 之间的接口类型 |
| *php_strip_whitespace* | 返回删除注释和空格后的 PHP 源码 |
| *php_uname* | 返回运行 PHP 的系统的有关信息 |
| *pi* | 得到圆周率值 |
| *png2wbmp* | 将 PNG 图像文件转换为 WBMP 图像文件 |
| *popen* | 打开进程文件指针 |
| *pos* | current 的别名 |
| *posix_access* | Determine accessibility of a file |
| *posix_ctermid* | Get path name of controlling terminal |
| *posix_errno* | 别名 posix_get_last_error |
| *posix_getcwd* | Pathname of current directory |
| *posix_getegid* | Return the effective group ID of the current process |
| *posix_geteuid* | Return the effective user ID of the current process |
| *posix_getgid* | Return the real group ID of the current process |
| *posix_getgrgid* | Return info about a group by group id |
| *posix_getgrnam* | Return info about a group by name |
| *posix_getgroups* | Return the group set of the current process |
| *posix_getlogin* | Return login name |
| *posix_getpgid* | Get process group id for job control |
| *posix_getpgrp* | Return the current process group identifier |
| *posix_getpid* | 返回当前进程 id |
| *posix_getppid* | Return the parent process identifier |
| *posix_getpwnam* | Return info about a user by username |
| *posix_getpwuid* | Return info about a user by user id |
| *posix_getrlimit* | Return info about system resource limits |
| *posix_getsid* | Get the current sid of the process |
| *posix_getuid* | Return the real user ID of the current process |
| *posix_get_last_error* | Retrieve the error number set by the last posix function that failed |
| *posix_initgroups* | Calculate the group access list |
| *posix_isatty* | Determine if a file descriptor is an interactive terminal |

| | |
|---|---|
| *posix_kill* | Send a signal to a process |
| *posix_mkfifo* | Create a fifo special file (a named pipe) |
| *posix_mknod* | Create a special or ordinary file (POSIX.1) |
| *posix_setegid* | Set the effective GID of the current process |
| *posix_seteuid* | Set the effective UID of the current process |
| *posix_setgid* | Set the GID of the current process |
| *posix_setpgid* | Set process group id for job control |
| *posix_setrlimit* | Set system resource limits |
| *posix_setsid* | Make the current process a session leader |
| *posix_setuid* | Set the UID of the current process |
| *posix_strerror* | Retrieve the system error message associated with the given errno |
| *posix_times* | Get process times |
| *posix_ttyname* | Determine terminal device name |
| *posix_uname* | Get system name |
| *pow* | 指数表达式 |
| *preg_filter* | 执行一个正则表达式搜索和替换 |
| *preg_grep* | 返回匹配模式的数组条目 |
| *preg_last_error* | 返回最后一个 PCRE 正则执行产生的错误代码 |
| *preg_match* | 执行一个正则表达式匹配 |
| *preg_match_all* | 执行一个全局正则表达式匹配 |
| *preg_quote* | 转义正则表达式字符 |
| *preg_replace* | 执行一个正则表达式的搜索和替换 |
| *preg_replace_callback* | 执行一个正则表达式搜索并且使用一个回调进行替换 |
| *preg_replace_callback_array* | Perform a regular expression search and replace using callbacks |
| *preg_split* | 通过一个正则表达式分隔字符串 |
| *prev* | 将数组的内部指针倒回一位 |
| *print* | 输出字符串 |
| *printf* | 输出格式化字符串 |
| *print_r* | 打印关于变量的易于理解的信息。 |
| *proc_close* | 关闭由 proc_open 打开的进程并且返回进程退出码 |
| *proc_get_status* | 获取由 proc_open 函数打开的进程的信息 |
| *proc_nice* | 修改当前进程的优先级 |
| *proc_open* | 执行一个命令，并且打开用来输入/输出的文件指针。 |
| *proc_terminate* | 杀除由 proc_open 打开的进程 |
| *property_exists* | 检查对象或类是否具有该属性 |
| *pspell_add_to_personal* | Add the word to a personal wordlist |
| *pspell_add_to_session* | Add the word to the wordlist in the current session |
| *pspell_check* | Check a word |
| *pspell_clear_session* | Clear the current session |
| *pspell_config_create* | Create a config used to open a dictionary |
| *pspell_config_data_dir* | location of language data files |
| *pspell_config_dict_dir* | Location of the main word list |
| *pspell_config_ignore* | Ignore words less than N characters long |
| *pspell_config_mode* | Change the mode number of suggestions returned |

| | |
|---|---|
| *pspell_config_personal* | Set a file that contains personal wordlist |
| *pspell_config_repl* | Set a file that contains replacement pairs |
| *pspell_config_runtogether* | Consider run-together words as valid compounds |
| *pspell_config_save_repl* | Determine whether to save a replacement pairs list along with the wordlist |
| *pspell_new* | Load a new dictionary |
| *pspell_new_config* | Load a new dictionary with settings based on a given config |
| *pspell_new_personal* | Load a new dictionary with personal wordlist |
| *pspell_save_wordlist* | Save the personal wordlist to a file |
| *pspell_store_replacement* | Store a replacement pair for a word |
| *pspell_suggest* | Suggest spellings of a word |
| *ps_add_bookmark* | Add bookmark to current page |
| *ps_add_launchlink* | Adds link which launches file |
| *ps_add_locallink* | Adds link to a page in the same document |
| *ps_add_note* | Adds note to current page |
| *ps_add_pdflink* | Adds link to a page in a second pdf document |
| *ps_add_weblink* | Adds link to a web location |
| *ps_arc* | Draws an arc counterclockwise |
| *ps_arcn* | Draws an arc clockwise |
| *ps_begin_page* | Start a new page |
| *ps_begin_pattern* | Start a new pattern |
| *ps_begin_template* | Start a new template |
| *ps_circle* | Draws a circle |
| *ps_clip* | Clips drawing to current path |
| *ps_close* | Closes a PostScript document |
| *ps_closepath* | Closes path |
| *ps_closepath_stroke* | Closes and strokes path |
| *ps_close_image* | Closes image and frees memory |
| *ps_continue_text* | Continue text in next line |
| *ps_curveto* | Draws a curve |
| *ps_delete* | Deletes all resources of a PostScript document |
| *ps_end_page* | End a page |
| *ps_end_pattern* | End a pattern |
| *ps_end_template* | End a template |
| *ps_fill* | Fills the current path |
| *ps_fill_stroke* | Fills and strokes the current path |
| *ps_findfont* | Loads a font |
| *ps_get_buffer* | Fetches the full buffer containig the generated PS data |
| *ps_get_parameter* | Gets certain parameters |
| *ps_get_value* | Gets certain values |
| *ps_hyphenate* | Hyphenates a word |
| *ps_include_file* | Reads an external file with raw PostScript code |
| *ps_lineto* | Draws a line |
| *ps_makespotcolor* | Create spot color |

| | |
|---|---|
| *ps_moveto* | Sets current point |
| *ps_new* | Creates a new PostScript document object |
| *ps_open_file* | Opens a file for output |
| *ps_open_image* | Reads an image for later placement |
| *ps_open_image_file* | Opens image from file |
| *ps_open_memory_image* | Takes an GD image and returns an image for placement in a PS document |
| *ps_place_image* | Places image on the page |
| *ps_rect* | Draws a rectangle |
| *ps_restore* | Restore previously save context |
| *ps_rotate* | Sets rotation factor |
| *ps_save* | Save current context |
| *ps_scale* | Sets scaling factor |
| *ps_setcolor* | Sets current color |
| *ps_setdash* | Sets appearance of a dashed line |
| *ps_setflat* | Sets flatness |
| *ps_setfont* | Sets font to use for following output |
| *ps_setgray* | Sets gray value |
| *ps_setlinecap* | Sets appearance of line ends |
| *ps_setlinejoin* | Sets how contected lines are joined |
| *ps_setlinewidth* | Sets width of a line |
| *ps_setmiterlimit* | Sets the miter limit |
| *ps_setoverprintmode* | Sets overprint mode |
| *ps_setpolydash* | Sets appearance of a dashed line |
| *ps_set_border_color* | Sets color of border for annotations |
| *ps_set_border_dash* | Sets length of dashes for border of annotations |
| *ps_set_border_style* | Sets border style of annotations |
| *ps_set_info* | Sets information fields of document |
| *ps_set_parameter* | Sets certain parameters |
| *ps_set_text_pos* | Sets position for text output |
| *ps_set_value* | Sets certain values |
| *ps_shading* | Creates a shading for later use |
| *ps_shading_pattern* | Creates a pattern based on a shading |
| *ps_shfill* | Fills an area with a shading |
| *ps_show* | Output text |
| *ps_show2* | Output a text at current position |
| *ps_show_boxed* | Output text in a box |
| *ps_show_xy* | Output text at given position |
| *ps_show_xy2* | Output text at position |
| *ps_stringwidth* | Gets width of a string |
| *ps_string_geometry* | Gets geometry of a string |
| *ps_stroke* | Draws the current path |
| *ps_symbol* | Output a glyph |
| *ps_symbol_name* | Gets name of a glyph |

| | |
|---|---|
| *ps_symbol_width* | Gets width of a glyph |
| *ps_translate* | Sets translation |
| *putenv* | 设置环境变量的值 |
| *px_close* | Closes a paradox database |
| *px_create_fp* | Create a new paradox database |
| *px_date2string* | Converts a date into a string. |
| *px_delete* | Deletes resource of paradox database |
| *px_delete_record* | Deletes record from paradox database |
| *px_get_field* | Returns the specification of a single field |
| *px_get_info* | Return lots of information about a paradox file |
| *px_get_parameter* | Gets a parameter |
| *px_get_record* | Returns record of paradox database |
| *px_get_schema* | Returns the database schema |
| *px_get_value* | Gets a value |
| *px_insert_record* | Inserts record into paradox database |
| *px_new* | Create a new paradox object |
| *px_numfields* | Returns number of fields in a database |
| *px_numrecords* | Returns number of records in a database |
| *px_open_fp* | Open paradox database |
| *px_put_record* | Stores record into paradox database |
| *px_retrieve_record* | Returns record of paradox database |
| *px_set_blob_file* | Sets the file where blobs are read from |
| *px_set_parameter* | Sets a parameter |
| *px_set_tablename* | Sets the name of a table (deprecated) |
| *px_set_targetencoding* | Sets the encoding for character fields (deprecated) |
| *px_set_value* | Sets a value |
| *px_timestamp2string* | Converts the timestamp into a string. |
| *px_update_record* | Updates record in paradox database |

# q

| *函数* | 说明 |
|---|---|
| *quoted_printable_decode* | 将 quoted-printable 字符串转换为 8-bit 字符串 |
| *quoted_printable_encode* | 将 8-bit 字符串转换成 quoted-printable 字符串 |
| *quotemeta* | 转义元字符集 |

# r

| *函数* | 说明 |
|---|---|
| *rad2deg* | 将弧度数转换为相应的角度数 |
| *radius_acct_open* | Creates a Radius handle for accounting |
| *radius_add_server* | Adds a server |
| *radius_auth_open* | Creates a Radius handle for authentication |

| radius_close | Frees all ressources |
|---|---|
| radius_config | Causes the library to read the given configuration file |
| radius_create_request | Create accounting or authentication request |
| radius_cvt_addr | Converts raw data to IP-Address |
| radius_cvt_int | Converts raw data to integer |
| radius_cvt_string | Converts raw data to string |
| radius_demangle | Demangles data |
| radius_demangle_mppe_key | Derives mppe-keys from mangled data |
| radius_get_attr | Extracts an attribute |
| radius_get_tagged_attr_data | Extracts the data from a tagged attribute |
| radius_get_tagged_attr_tag | Extracts the tag from a tagged attribute |
| radius_get_vendor_attr | Extracts a vendor specific attribute |
| radius_put_addr | Attaches an IP address attribute |
| radius_put_attr | Attaches a binary attribute |
| radius_put_int | Attaches an integer attribute |
| radius_put_string | Attaches a string attribute |
| radius_put_vendor_addr | Attaches a vendor specific IP address attribute |
| radius_put_vendor_attr | Attaches a vendor specific binary attribute |
| radius_put_vendor_int | Attaches a vendor specific integer attribute |
| radius_put_vendor_string | Attaches a vendor specific string attribute |
| radius_request_authenticator | Returns the request authenticator |
| radius_salt_encrypt_attr | Salt-encrypts a value |
| radius_send_request | Sends the request and waites for a reply |
| radius_server_secret | Returns the shared secret |
| radius_strerror | Returns an error message |
| rand | 产生一个随机整数 |
| random_bytes | Generates cryptographically secure pseudo-random bytes |
| random_int | Generates cryptographically secure pseudo-random integers |
| range | 建立一个包含指定范围单元的数组 |
| rar_wrapper_cache_stats | Cache hits and misses for the URL wrapper |
| rawurldecode | 对已编码的 URL 字符串进行解码 |
| rawurlencode | 按照 RFC 3986 对 URL 进行编码 |
| readdir | 从目录句柄中读取条目 |
| readfile | 输出一个文件 |
| readgzfile | Output a gz-file |
| readline | 读取一行 |
| readline_add_history | 添加一行命令行历史记录 |
| readline_callback_handler_install | 初始化一个 readline 回调接口，然后终端输出提示信息并立即返回 |
| readline_callback_handler_remove | 移除上一个安装的回调函数句柄并且恢复终端设置 |
| readline_callback_read_char | 当一个行被接收时读取一个字符并且通知 readline 调用回调函数 |
| readline_clear_history | 清除历史 |
| readline_completion_function | 注册一个完成函数 |
| readline_info | 获取/设置 readline 内部的各个变量 |
| readline_list_history | 获取命令历史列表 |

| | |
|---|---|
| *readline_on_new_line* | 通知 readline 将光标移动到新行 |
| *readline_read_history* | 读取命令历史 |
| *readline_redisplay* | 重绘显示区 |
| *readline_write_history* | 写入历史记录 |
| *readlink* | 返回符号连接指向的目标 |
| *read_exif_data* | 别名 exif_read_data |
| *realpath* | 返回规范化的绝对路径名 |
| *realpath_cache_get* | Get realpath cache entries |
| *realpath_cache_size* | Get realpath cache size |
| *recode* | 别名 recode_string |
| *recode_file* | Recode from file to file according to recode request |
| *recode_string* | Recode a string according to a recode request |
| *register_shutdown_function* | Register a function for execution on shutdown |
| *register_tick_function* | Register a function for execution on each tick |
| *rename* | 重命名一个文件或目录 |
| *rename_function* | Renames orig_name to new_name in the global function table |
| *reset* | 将数组的内部指针指向第一个单元 |
| *restore_error_handler* | 还原之前的错误处理函数 |
| *restore_exception_handler* | 恢复之前定义过的异常处理函数。 |
| *restore_include_path* | 还原 include_path 配置选项的值 |
| *rewind* | 倒回文件指针的位置 |
| *rewinddir* | 倒回目录句柄 |
| *rmdir* | 删除目录 |
| *round* | 对浮点数进行四舍五入 |
| *rpm_close* | Closes an RPM file |
| *rpm_get_tag* | Retrieves a header tag from an RPM file |
| *rpm_is_valid* | Tests a filename for validity as an RPM file |
| *rpm_open* | Opens an RPM file |
| *rpm_version* | Returns a string representing the current version of the rpmreader extension |
| *rrdc_disconnect* | Close any outstanding connection to rrd caching daemon |
| *rrd_create* | Creates rrd database file |
| *rrd_error* | Gets latest error message. |
| *rrd_fetch* | Fetch the data for graph as array. |
| *rrd_first* | Gets the timestamp of the first sample from rrd file. |
| *rrd_graph* | Creates image from a data. |
| *rrd_info* | Gets information about rrd file |
| *rrd_last* | Gets unix timestamp of the last sample. |
| *rrd_lastupdate* | Gets information about last updated data. |
| *rrd_restore* | Restores the RRD file from XML dump. |
| *rrd_tune* | Tunes some RRD database file header options. |
| *rrd_update* | Updates the RRD database. |
| *rrd_version* | Gets information about underlying rrdtool library |
| *rrd_xport* | Exports the information about RRD database. |

| | |
|---|---|
| *rsort* | 对数组逆向排序 |
| *rtrim* | 删除字符串末端的空白字符（或者其他字符） |
| *runkit_class_adopt* | Convert a base class to an inherited class, add ancestral methods when appropriate |
| *runkit_class_emancipate* | Convert an inherited class to a base class, removes any method whose scope is ancestral |
| *runkit_constant_add* | Similar to define(), but allows defining in class definitions as well |
| *runkit_constant_redefine* | Redefine an already defined constant |
| *runkit_constant_remove* | Remove/Delete an already defined constant |
| *runkit_function_add* | Add a new function, similar to create_function |
| *runkit_function_copy* | Copy a function to a new function name |
| *runkit_function_redefine* | Replace a function definition with a new implementation |
| *runkit_function_remove* | Remove a function definition |
| *runkit_function_rename* | Change a function's name |
| *runkit_import* | Process a PHP file importing function and class definitions, overwriting where appropriate |
| *runkit_lint* | Check the PHP syntax of the specified php code |
| *runkit_lint_file* | Check the PHP syntax of the specified file |
| *runkit_method_add* | Dynamically adds a new method to a given class |
| *runkit_method_copy* | Copies a method from class to another |
| *runkit_method_redefine* | Dynamically changes the code of the given method |
| *runkit_method_remove* | Dynamically removes the given method |
| *runkit_method_rename* | Dynamically changes the name of the given method |
| *runkit_return_value_used* | Determines if the current functions return value will be used |
| *Runkit_Sandbox* | Runkit Sandbox Class – PHP Virtual Machine |
| *runkit_sandbox_output_handler* | Specify a function to capture and/or process output from a runkit sandbox |
| *Runkit_Sandbox_Parent* | Runkit Anti-Sandbox Class |
| | |
| *runkit_superglobals* | Return numerically indexed array of registered superglobals |

**S**

| *函数* | 说明 |
|---|---|
| *scandir* | 列出指定路径中的文件和目录 |
| *sem_acquire* | Acquire a semaphore |
| *sem_get* | Get a semaphore id |
| *sem_release* | Release a semaphore |
| *sem_remove* | Remove a semaphore |
| *serialize* | 产生一个可存储的值的表示 |
| *session_abort* | Discard session array changes and finish session |
| *session_cache_expire* | 返回当前缓存的到期时间 |
| *session_cache_limiter* | 读取/设置缓存限制器 |

| | |
|---|---|
| *session_commit* | session_write_close 的别名 |
| *session_decode* | 解码会话数据 |
| *session_destroy* | 销毁一个会话中的全部数据 |
| *session_encode* | 将当前会话数据编码为一个字符串 |
| *session_get_cookie_params* | 获取会话 cookie 参数 |
| *session_id* | 获取/设置当前会话 ID |
| *session_is_registered* | 检查变量是否在会话中已经注册 |
| *session_module_name* | 获取/设置会话模块名称 |
| *session_name* | 读取/设置会话名称 |
| *session_pgsql_add_error* | Increments error counts and sets last error message |
| *session_pgsql_get_error* | Returns number of errors and last error message |
| *session_pgsql_get_field* | Get custom field value |
| *session_pgsql_reset* | Reset connection to session database servers |
| *session_pgsql_set_field* | Set custom field value |
| *session_pgsql_status* | Get current save handler status |
| *session_regenerate_id* | 使用新生成的会话 ID 更新现有会话 ID |
| *session_register* | Register one or more global variables with the current session |
| *session_register_shutdown* | 关闭会话 |
| *session_reset* | Re-initialize session array with original values |
| *session_save_path* | 读取/设置当前会话的保存路径 |
| *session_set_cookie_params* | 设置会话 cookie 参数 |
| *session_set_save_handler* | 设置用户自定义会话存储函数 |
| *session_start* | 启动新会话或者重用现有会话 |
| *session_status* | Returns the current session status |
| *session_unregister* | Unregister a global variable from the current session |
| *session_unset* | Free all session variables |
| *session_write_close* | Write session data and end session |
| *setcookie* | Send a cookie |
| *setlocale* | 设置地区信息 |
| *setproctitle* | Set the process title |
| *setrawcookie* | Send a cookie without urlencoding the cookie value |
| *setthreadtitle* | Set the thread title |
| *settype* | 设置变量的类型 |
| *set_error_handler* | 设置一个用户定义的错误处理函数 |
| *set_exception_handler* | 设置一个用户定义的异常处理函数。 |
| *set_file_buffer* | stream_set_write_buffer 的别名 |
| *set_include_path* | 设置 include_path 配置选项 |
| *set_magic_quotes_runtime* | 设置当前 magic_quotes_runtime 配置选项的激活状态 |
| *set_socket_blocking* | 别名 stream_set_blocking |
| *set_time_limit* | 设置脚本最大执行时间 |
| *sha1* | 计算字符串的 sha1 散列值 |
| *sha1_file* | 计算文件的 sha1 散列值 |
| *shell_exec* | 通过 shell 环境执行命令，并且将完整的输出以字符串的方式返回。 |
| *shmop_close* | Close shared memory block |

| | |
|---|---|
| *shmop_delete* | Delete shared memory block |
| *shmop_open* | Create or open shared memory block |
| *shmop_read* | Read data from shared memory block |
| *shmop_size* | Get size of shared memory block |
| *shmop_write* | Write data into shared memory block |
| *shm_attach* | Creates or open a shared memory segment |
| *shm_detach* | Disconnects from shared memory segment |
| *shm_get_var* | Returns a variable from shared memory |
| *shm_has_var* | Check whether a specific entry exists |
| *shm_put_var* | Inserts or updates a variable in shared memory |
| *shm_remove* | Removes shared memory from Unix systems |
| *shm_remove_var* | Removes a variable from shared memory |
| *show_source* | 别名 highlight_file |
| *shuffle* | 将数组打乱 |
| *signeurlpaiement* | Obtain the payment url (needs 2 arguments) |
| *similar_text* | 计算两个字符串的相似度 |
| *simplexml_import_dom* | Get a SimpleXMLElement object from a DOM node. |
| *simplexml_load_file* | Interprets an XML file into an object |
| *simplexml_load_string* | Interprets a string of XML into an object |
| *sin* | 正弦 |
| *sinh* | 双曲正弦 |
| *sizeof* | count 的别名 |
| *sleep* | 延缓执行 |
| *snmp2_get* | Fetch an SNMP object |
| *snmp2_getnext* | Fetch the SNMP object which follows the given object id |
| *snmp2_real_walk* | Return all objects including their respective object ID within the specified one |
| *snmp2_set* | Set the value of an SNMP object |
| *snmp2_walk* | Fetch all the SNMP objects from an agent |
| *snmp3_get* | Fetch an SNMP object |
| *snmp3_getnext* | Fetch the SNMP object which follows the given object id |
| *snmp3_real_walk* | Return all objects including their respective object ID within the specified one |
| *snmp3_set* | Set the value of an SNMP object |
| *snmp3_walk* | Fetch all the SNMP objects from an agent |
| *snmpget* | 获取一个 SNMP 对象 |
| *snmpgetnext* | Fetch the SNMP object which follows the given object id |
| *snmprealwalk* | 返回指定的所有对象，包括它们各自的对象 ID |
| *snmpset* | 设置一个 SNMP 对象 |
| *snmpwalk* | 从代理返回所有的 SNMP 对象 |
| *snmpwalkoid* | 查询关于网络实体的信息树 |
| *snmp_get_quick_print* | 返回 UCD 库中 quick_print 设置的当前值 |
| *snmp_get_valueretrieval* | Return the method how the SNMP values will be returned |
| *snmp_read_mib* | Reads and parses a MIB file into the active MIB tree |

| | |
|---|---|
| *snmp_set_enum_print* | Return all values that are enums with their enum value instead of the raw integer |
| *snmp_set_oid_numeric_print* | Set the OID output format |
| *snmp_set_oid_output_format* | Set the OID output format |
| *snmp_set_quick_print* | 设置 UCD SNMP 库中 quick_print 的值 |
| *snmp_set_valueretrieval* | Specify the method how the SNMP values will be returned |
| *socket_accept* | Accepts a connection on a socket |
| *socket_bind* | 给套接字绑定名字 |
| *socket_clear_error* | 清除套接字或者最后的错误代码上的错误 |
| *socket_close* | 关闭套接字资源 |
| *socket_cmsg_space* | Calculate message buffer size |
| *socket_connect* | 开启一个套接字连接 |
| *socket_create* | 创建一个套接字（通讯节点） |
| *socket_create_listen* | Opens a socket on port to accept connections |
| *socket_create_pair* | Creates a pair of indistinguishable sockets and stores them in an array |
| *socket_getpeername* | Queries the remote side of the given socket which may either result in host/port or in a Unix filesystem path, dependent on its type |
| *socket_getsockname* | Queries the local side of the given socket which may either result in host/port or in a Unix filesystem path, dependent on its type |
| *socket_get_option* | Gets socket options for the socket |
| *socket_get_status* | 别名 stream_get_meta_data |
| *socket_import_stream* | Import a stream |
| *socket_last_error* | Returns the last error on the socket |
| *socket_listen* | Listens for a connection on a socket |
| *socket_read* | Reads a maximum of length bytes from a socket |
| *socket_recv* | Receives data from a connected socket |
| *socket_recvfrom* | Receives data from a socket whether or not it is connection-oriented |
| *socket_recvmsg* | Read a message |
| *socket_select* | Runs the select() system call on the given arrays of sockets with a specified timeout |
| *socket_send* | Sends data to a connected socket |
| *socket_sendmsg* | Send a message |
| *socket_sendto* | Sends a message to a socket, whether it is connected or not |
| *socket_set_block* | Sets blocking mode on a socket resource |
| *socket_set_blocking* | 别名 stream_set_blocking |
| *socket_set_nonblock* | Sets nonblocking mode for file descriptor fd |
| *socket_set_option* | Sets socket options for the socket |
| *socket_set_timeout* | 别名 stream_set_timeout |
| *socket_shutdown* | Shuts down a socket for receiving, sending, or both |
| *socket_strerror* | Return a string describing a socket error |
| *socket_write* | Write to a socket |
| *solr_get_version* | 返回当前 Solr 扩展的版本 |
| *sort* | 对数组排序 |

| | |
|---|---|
| *soundex* | Calculate the soundex key of a string |
| *split* | 用正则表达式将字符串分割到数组中 |
| *spliti* | 用正则表达式不区分大小写将字符串分割到数组中 |
| *spl_autoload* | __autoload()函数的默认实现 |
| *spl_autoload_call* | 尝试调用所有已注册的__autoload()函数来装载请求类 |
| *spl_autoload_extensions* | 注册并返回 spl_autoload 函数使用的默认文件扩展名。 |
| *spl_autoload_functions* | 返回所有已注册的__autoload()函数。 |
| *spl_autoload_register* | 注册给定的函数作为 __autoload 的实现 |
| *spl_autoload_unregister* | 注销已注册的__autoload()函数 |
| *spl_classes* | 返回所有可用的 SPL 类 |
| *spl_object_hash* | 返回指定对象的 hash id |
| *sprintf* | Return a formatted string |
| *SQL acceptable by 4D* | PDO and SQL 4D |
| *sqlite_array_query* | Execute a query against a given database and returns an array |
| *sqlite_busy_timeout* | Set busy timeout duration, or disable busy handlers |
| *sqlite_changes* | Returns the number of rows that were changed by the most recent SQL statement |
| *sqlite_close* | Closes an open SQLite database |
| *sqlite_column* | Fetches a column from the current row of a result set |
| *sqlite_create_aggregate* | Register an aggregating UDF for use in SQL statements |
| *sqlite_create_function* | Registers a "regular" User Defined Function for use in SQL statements |
| *sqlite_current* | Fetches the current row from a result set as an array |
| *sqlite_error_string* | Returns the textual description of an error code |
| *sqlite_escape_string* | Escapes a string for use as a query parameter |
| *sqlite_exec* | Executes a result-less query against a given database |
| *sqlite_factory* | Opens an SQLite database and returns an SQLiteDatabase object |
| *sqlite_fetch_all* | Fetches all rows from a result set as an array of arrays |
| *sqlite_fetch_array* | Fetches the next row from a result set as an array |
| *sqlite_fetch_column_types* | Return an array of column types from a particular table |
| *sqlite_fetch_object* | Fetches the next row from a result set as an object |
| *sqlite_fetch_single* | Fetches the first column of a result set as a string |
| *sqlite_fetch_string* | 别名 sqlite_fetch_single |
| *sqlite_field_name* | Returns the name of a particular field |
| *sqlite_has_more* | Finds whether or not more rows are available |
| *sqlite_has_prev* | Returns whether or not a previous row is available |
| *sqlite_key* | Returns the current row index |
| *sqlite_last_error* | Returns the error code of the last error for a database |
| *sqlite_last_insert_rowid* | Returns the rowid of the most recently inserted row |
| *sqlite_libencoding* | Returns the encoding of the linked SQLite library |
| *sqlite_libversion* | Returns the version of the linked SQLite library |
| *sqlite_next* | Seek to the next row number |
| *sqlite_num_fields* | Returns the number of fields in a result set |
| *sqlite_num_rows* | Returns the number of rows in a buffered result set |
| *sqlite_open* | Opens an SQLite database and create the database if it does not |

| | exist |
|---|---|
| *sqlite_popen* | Opens a persistent handle to an SQLite database and create the database if it does not exist |
| *sqlite_prev* | Seek to the previous row number of a result set |
| *sqlite_query* | Executes a query against a given database and returns a result handle |
| *sqlite_rewind* | Seek to the first row number |
| *sqlite_seek* | Seek to a particular row number of a buffered result set |
| *sqlite_single_query* | Executes a query and returns either an array for one single column or the value of the first row |
| *sqlite_udf_decode_binary* | Decode binary data passed as parameters to an UDF |
| *sqlite_udf_encode_binary* | Encode binary data before returning it from an UDF |
| *sqlite_unbuffered_query* | Execute a query that does not prefetch and buffer all data |
| *sqlite_valid* | Returns whether more rows are available |
| *sqlsrv_begin_transaction* | Begins a database transaction |
| *sqlsrv_cancel* | Cancels a statement |
| *sqlsrv_client_info* | Returns information about the client and specified connection |
| *sqlsrv_close* | Closes an open connection and releases resources associated with the connection |
| *sqlsrv_commit* | Commits a transaction that was begun with sqlsrv_begin_transaction |
| *sqlsrv_configure* | Changes the driver error handling and logging configurations |
| *sqlsrv_connect* | Opens a connection to a Microsoft SQL Server database |
| *sqlsrv_errors* | Returns error and warning information about the last SQLSRV operation performed |
| *sqlsrv_execute* | Executes a statement prepared with sqlsrv_prepare |
| *sqlsrv_fetch* | Makes the next row in a result set available for reading |
| *sqlsrv_fetch_array* | Returns a row as an array |
| *sqlsrv_fetch_object* | Retrieves the next row of data in a result set as an object |
| *sqlsrv_field_metadata* | Retrieves metadata for the fields of a statement prepared by sqlsrv_prepare or sqlsrv_query |
| *sqlsrv_free_stmt* | Frees all resources for the specified statement |
| *sqlsrv_get_config* | Returns the value of the specified configuration setting |
| *sqlsrv_get_field* | Gets field data from the currently selected row |
| *sqlsrv_has_rows* | Indicates whether the specified statement has rows |
| *sqlsrv_next_result* | Makes the next result of the specified statement active |
| *sqlsrv_num_fields* | Retrieves the number of fields (columns) on a statement |
| *sqlsrv_num_rows* | Retrieves the number of rows in a result set |
| *sqlsrv_prepare* | Prepares a query for execution |
| *sqlsrv_query* | Prepares and executes a query. |
| *sqlsrv_rollback* | Rolls back a transaction that was begun with sqlsrv_begin_transaction |
| *sqlsrv_rows_affected* | Returns the number of rows modified by the last INSERT, UPDATE, or DELETE query executed |
| *sqlsrv_send_stream_data* | Sends data from parameter streams to the server |

| | |
|---|---|
| *sqlsrv_server_info* | Returns information about the server |
| *SQL types with PDO_4D and PHP* | SQL types with PDO_4D and PHP |
| *sql_regcase* | 产生用于不区分大小的匹配的正则表达式 |
| *sqrt* | 平方根 |
| *srand* | 播下随机数发生器种子 |
| *sscanf* | 根据指定格式解析输入的字符 |
| *ssdeep_fuzzy_compare* | Calculates the match score between two fuzzy hash signatures |
| *ssdeep_fuzzy_hash* | Create a fuzzy hash from a string |
| *ssdeep_fuzzy_hash_filename* | Create a fuzzy hash from a file |
| *ssh2_auth_agent* | Authenticate over SSH using the ssh agent |
| *ssh2_auth_hostbased_file* | Authenticate using a public hostkey |
| *ssh2_auth_none* | Authenticate as "none" |
| *ssh2_auth_password* | Authenticate over SSH using a plain password |
| *ssh2_auth_pubkey_file* | Authenticate using a public key |
| *ssh2_connect* | Connect to an SSH server |
| *ssh2_exec* | Execute a command on a remote server |
| *ssh2_fetch_stream* | Fetch an extended data stream |
| *ssh2_fingerprint* | Retrieve fingerprint of remote server |
| *ssh2_methods_negotiated* | Return list of negotiated methods |
| *ssh2_publickey_add* | Add an authorized publickey |
| *ssh2_publickey_init* | Initialize Publickey subsystem |
| *ssh2_publickey_list* | List currently authorized publickeys |
| *ssh2_publickey_remove* | Remove an authorized publickey |
| *ssh2_scp_recv* | Request a file via SCP |
| *ssh2_scp_send* | Send a file via SCP |
| *ssh2_sftp* | Initialize SFTP subsystem |
| *ssh2_sftp_chmod* | Changes file mode |
| *ssh2_sftp_lstat* | Stat a symbolic link |
| *ssh2_sftp_mkdir* | Create a directory |
| *ssh2_sftp_readlink* | Return the target of a symbolic link |
| *ssh2_sftp_realpath* | Resolve the realpath of a provided path string |
| *ssh2_sftp_rename* | Rename a remote file |
| *ssh2_sftp_rmdir* | Remove a directory |
| *ssh2_sftp_stat* | Stat a file on a remote filesystem |
| *ssh2_sftp_symlink* | Create a symlink |
| *ssh2_sftp_unlink* | Delete a file |
| *ssh2_shell* | Request an interactive shell |
| *ssh2_tunnel* | Open a tunnel through a remote server |
| *SSL 上下文选项* | SSL 上下文选项清单 |
| *stat* | 给出文件的信息 |
| *stats_absolute_deviation* | Returns the absolute deviation of an array of values |
| *stats_cdf_beta* | CDF function for BETA Distribution. Calculates any one parameter of the beta distribution given values for the others. |
| *stats_cdf_binomial* | Calculates any one parameter of the binomial distribution given |

| | |
|---|---|
| | values for the others. |
| *stats_cdf_cauchy* | Not documented |
| *stats_cdf_chisquare* | Calculates any one parameter of the chi-square distribution given values for the others. |
| *stats_cdf_exponential* | Not documented |
| *stats_cdf_f* | Calculates any one parameter of the F distribution given values for the others. |
| *stats_cdf_gamma* | Calculates any one parameter of the gamma distribution given values for the others. |
| *stats_cdf_laplace* | Not documented |
| *stats_cdf_logistic* | Not documented |
| *stats_cdf_negative_binomial* | Calculates any one parameter of the negative binomial distribution given values for the others. |
| *stats_cdf_noncentral_chisquare* | Calculates any one parameter of the non-central chi-square distribution given values for the others. |
| *stats_cdf_noncentral_f* | Calculates any one parameter of the Non-central F distribution given values for the others. |
| *stats_cdf_poisson* | Calculates any one parameter of the Poisson distribution given values for the others. |
| *stats_cdf_t* | Calculates any one parameter of the T distribution given values for the others. |
| *stats_cdf_uniform* | Not documented |
| *stats_cdf_weibull* | Not documented |
| *stats_covariance* | Computes the covariance of two data sets |
| *stats_dens_beta* | Not documented |
| *stats_dens_cauchy* | Not documented |
| *stats_dens_chisquare* | Not documented |
| *stats_dens_exponential* | Not documented |
| *stats_dens_f* | 说明 |
| *stats_dens_gamma* | Not documented |
| *stats_dens_laplace* | Not documented |
| *stats_dens_logistic* | Not documented |
| *stats_dens_negative_binomial* | Not documented |
| *stats_dens_normal* | Not documented |
| *stats_dens_pmf_binomial* | Not documented |
| *stats_dens_pmf_hypergeometric* | 说明 |
| *stats_dens_pmf_poisson* | Not documented |
| *stats_dens_t* | Not documented |
| *stats_dens_weibull* | Not documented |
| *stats_den_uniform* | Not documented |
| *stats_harmonic_mean* | Returns the harmonic mean of an array of values |
| *stats_kurtosis* | Computes the kurtosis of the data in the array |
| *stats_rand_gen_beta* | Generates beta random deviate |
| *stats_rand_gen_chisquare* | Generates random deviate from the distribution of a chisquare with "df" degrees of freedom random variable. |

| | |
|---|---|
| *stats_rand_gen_exponential* | Generates a single random deviate from an exponential distribution with mean "av" |
| *stats_rand_gen_f* | Generates a random deviate |
| *stats_rand_gen_funiform* | Generates uniform float between low (exclusive) and high (exclusive) |
| *stats_rand_gen_gamma* | Generates random deviates from a gamma distribution |
| *stats_rand_gen_ibinomial* | Generates a single random deviate from a binomial distribution whose number of trials is "n" (n >= 0) and whose probability of an event in each trial is "pp" ([0;1]). Method : algorithm BTPE |
| *stats_rand_gen_ibinomial_negative* | Generates a single random deviate from a negative binomial distribution. Arguments : n |
| *stats_rand_gen_int* | Generates random integer between 1 and 2147483562 |
| *stats_rand_gen_ipoisson* | Generates a single random deviate from a Poisson distribution with mean "mu" (mu >= 0.0). |
| *stats_rand_gen_iuniform* | Generates integer uniformly distributed between LOW (inclusive) and HIGH (inclusive) |
| *stats_rand_gen_noncenral_chisquare* | Generates random deviate from the distribution of a noncentral chisquare with "df" degrees of freedom and noncentrality parameter "xnonc". d must be >= 1.0, xnonc must >= 0.0 |
| *stats_rand_gen_noncentral_f* | Generates a random deviate from the noncentral F (variance ratio) distribution with "dfn" degrees of freedom in the numerator, and "dfd" degrees of freedom in the denominator, and noncentrality parameter "xnonc". Method : directly generates ratio of noncentral numerator chisquare variate to central denominator chisquare variate. |
| *stats_rand_gen_noncentral_t* | Generates a single random deviate from a noncentral T distribution |
| *stats_rand_gen_normal* | Generates a single random deviate from a normal distribution with mean, av, and standard deviation, sd (sd >= 0). Method : Renames SNORM from TOMS as slightly modified by BWB to use RANF instead of SUNIF. |
| *stats_rand_gen_t* | Generates a single random deviate from a T distribution |
| *stats_rand_get_seeds* | Not documented |
| *stats_rand_phrase_to_seeds* | generate two seeds for the RGN random number generator |
| *stats_rand_ranf* | Returns a random floating point number from a uniform distribution over 0 |
| *stats_rand_setall* | Not documented |
| *stats_skew* | Computes the skewness of the data in the array |
| *stats_standard_deviation* | Returns the standard deviation |
| *stats_stat_binomial_coef* | Not documented |
| *stats_stat_correlation* | Not documented |
| *stats_stat_gennch* | Not documented |
| *stats_stat_independent_t* | Not documented |
| *stats_stat_innerproduct* | 说明 |
| *stats_stat_noncentral_t* | Calculates any one parameter of the noncentral t distribution give values for the others. |
| *stats_stat_paired_t* | Not documented |
| *stats_stat_percentile* | Not documented |

| | |
|---|---|
| *stats_stat_powersum* | Not documented |
| *stats_variance* | Returns the population variance |
| *stomp_connect_error* | Returns a string description of the last connect error |
| *stomp_version* | Gets the current stomp extension version |
| *strcasecmp* | 二进制安全比较字符串（不区分大小写） |
| *strchr* | 别名 strstr |
| *strcmp* | 二进制安全字符串比较 |
| *strcoll* | 基于区域设置的字符串比较 |
| *strcspn* | 获取不匹配遮罩的起始子字符串的长度 |
| *stream_bucket_append* | Append bucket to brigade |
| *stream_bucket_make_writeable* | Return a bucket object from the brigade for operating on |
| *stream_bucket_new* | Create a new bucket for use on the current stream |
| *stream_bucket_prepend* | Prepend bucket to brigade |
| *stream_context_create* | 创建资源流上下文 |
| *stream_context_get_default* | Retrieve the default stream context |
| *stream_context_get_options* | 获取资源流/数据包/上下文的参数 |
| *stream_context_get_params* | Retrieves parameters from a context |
| *stream_context_set_default* | Set the default stream context |
| *stream_context_set_option* | 对资源流、数据包或者上下文设置参数 |
| *stream_context_set_params* | Set parameters for a stream/wrapper/context |
| *stream_copy_to_stream* | Copies data from one stream to another |
| *stream_encoding* | 设置数据流的字符集 |
| *stream_filter_append* | Attach a filter to a stream |
| *stream_filter_prepend* | Attach a filter to a stream |
| *stream_filter_register* | Register a user defined stream filter |
| *stream_filter_remove* | 从资源流里移除某个过滤器 |
| *stream_get_contents* | 读取资源流到一个字符串 |
| *stream_get_filters* | 获取已注册的数据流过滤器列表 |
| *stream_get_line* | 从资源流里读取一行直到给定的定界符 |
| *stream_get_meta_data* | 从封装协议文件指针中取得报头 / 元数据 |
| *stream_get_transports* | 获取已注册的套接字传输协议列表 |
| *stream_get_wrappers* | 获取已注册的流类型 |
| *stream_is_local* | Checks if a stream is a local stream |
| *stream_notification_callback* | A callback function for the notification context parameter |
| *stream_register_wrapper* | 注册一个用 PHP 类实现的 URL 封装协议 |
| *stream_resolve_include_path* | Resolve filename against the include path |
| *stream_select* | Runs the equivalent of the select() system call on the given arrays of streams with a timeout specified by tv_sec and tv_usec |
| *stream_set_blocking* | 为资源流设置阻塞或者阻塞模式 |
| *stream_set_chunk_size* | 设置资源流区块大小 |
| *stream_set_read_buffer* | Set read file buffering on the given stream |
| *stream_set_timeout* | Set timeout period on a stream |
| *stream_set_write_buffer* | Sets write file buffering on the given stream |
| *stream_socket_accept* | 接受由 stream_socket_server 创建的套接字连接 |

| stream_socket_client | Open Internet or Unix domain socket connection |
|---|---|
| stream_socket_enable_crypto | Turns encryption on/off on an already connected socket |
| stream_socket_get_name | 获取本地或者远程的套接字名称 |
| stream_socket_pair | 创建一对完全一样的网络套接字连接流 |
| stream_socket_recvfrom | Receives data from a socket, connected or not |
| stream_socket_sendto | Sends a message to a socket, whether it is connected or not |
| stream_socket_server | Create an Internet or Unix domain server socket |
| stream_socket_shutdown | Shutdown a full-duplex connection |
| stream_supports_lock | Tells whether the stream supports locking. |
| stream_wrapper_restore | Restores a previously unregistered built-in wrapper |
| stream_wrapper_unregister | Unregister a URL wrapper |
| strftime | 根据区域设置格式化本地时间／日期 |
| stripcslashes | 反引用一个使用 addcslashes 转义的字符串 |
| stripos | 查找字符串首次出现的位置（不区分大小写） |
| stripslashes | 反引用一个引用字符串 |
| strip_tags | 从字符串中去除 HTML 和 PHP 标记 |
| stristr | strstr 函数的忽略大小写版本 |
| strlen | 获取字符串长度 |
| strnatcasecmp | 使用"自然顺序"算法比较字符串（不区分大小写） |
| strnatcmp | 使用自然排序算法比较字符串 |
| strncasecmp | 二进制安全比较字符串开头的若干个字符（不区分大小写） |
| strncmp | 二进制安全比较字符串开头的若干个字符 |
| strpbrk | 在字符串中查找一组字符的任何一个字符 |
| strpos | 查找字符串首次出现的位置 |
| strptime | 解析由 strftime 生成的日期／时间 |
| strrchr | 查找指定字符在字符串中的最后一次出现 |
| strrev | 反转字符串 |
| strripos | 计算指定字符串在目标字符串中最后一次出现的位置（不区分大小写） |
| strrpos | 计算指定字符串在目标字符串中最后一次出现的位置 |
| strspn | 计算字符串中全部字符都存在于指定字符集合中的第一段子串的长度。 |
| strstr | 查找字符串的首次出现 |
| strtok | 标记分割字符串 |
| strtolower | 将字符串转化为小写 |
| strtotime | 将任何英文文本的日期时间描述解析为 Unix 时间戳 |
| strtoupper | 将字符串转化为大写 |
| strtr | 转换指定字符 |
| strval | 获取变量的字符串值 |
| str_getcsv | 解析 CSV 字符串为一个数组 |
| str_ireplace | str_replace 的忽略大小写版本 |
| str_pad | 使用另一个字符串填充字符串为指定长度 |
| str_repeat | 重复一个字符串 |
| str_replace | 子字符串替换 |
| str_rot13 | 对字符串执行 ROT13 转换 |
| str_shuffle | 随机打乱一个字符串 |

| | |
|---|---|
| *str_split* | 将字符串转换为数组 |
| *str_word_count* | 返回字符串中单词的使用情况 |
| *substr* | 返回字符串的子串 |
| *substr_compare* | 二进制安全比较字符串（从偏移位置比较指定长度） |
| *substr_count* | 计算字串出现的次数 |
| *substr_replace* | 替换字符串的子串 |
| *svn_add* | 计划在工作目录添加项 |
| *svn_auth_get_parameter* | Retrieves authentication parameter |
| *svn_auth_set_parameter* | Sets an authentication parameter |
| *svn_blame* | Get the SVN blame for a file |
| *svn_cat* | Returns the contents of a file in a repository |
| *svn_checkout* | Checks out a working copy from the repository |
| *svn_cleanup* | Recursively cleanup a working copy directory, finishing incomplete operations and removing locks |
| *svn_client_version* | Returns the version of the SVN client libraries |
| *svn_commit* | Sends changes from the local working copy to the repository |
| *svn_delete* | Delete items from a working copy or repository. |
| *svn_diff* | Recursively diffs two paths |
| *svn_export* | Export the contents of a SVN directory |
| *svn_fs_abort_txn* | Abort a transaction, returns true if everything is okay, false otherwise |
| *svn_fs_apply_text* | Creates and returns a stream that will be used to replace |
| *svn_fs_begin_txn2* | Create a new transaction |
| *svn_fs_change_node_prop* | Return true if everything is ok, false otherwise |
| *svn_fs_check_path* | Determines what kind of item lives at path in a given repository fsroot |
| *svn_fs_contents_changed* | Return true if content is different, false otherwise |
| *svn_fs_copy* | Copies a file or a directory, returns true if all is ok, false otherwise |
| *svn_fs_delete* | Deletes a file or a directory, return true if all is ok, false otherwise |
| *svn_fs_dir_entries* | Enumerates the directory entries under path; returns a hash of dir names to file type |
| *svn_fs_file_contents* | Returns a stream to access the contents of a file from a given version of the fs |
| *svn_fs_file_length* | Returns the length of a file from a given version of the fs |
| *svn_fs_is_dir* | Return true if the path points to a directory, false otherwise |
| *svn_fs_is_file* | Return true if the path points to a file, false otherwise |
| *svn_fs_make_dir* | Creates a new empty directory, returns true if all is ok, false otherwise |
| *svn_fs_make_file* | Creates a new empty file, returns true if all is ok, false otherwise |
| *svn_fs_node_created_rev* | Returns the revision in which path under fsroot was created |
| *svn_fs_node_prop* | Returns the value of a property for a node |
| *svn_fs_props_changed* | Return true if props are different, false otherwise |
| *svn_fs_revision_prop* | Fetches the value of a named property |

| | |
|---|---|
| *svn_fs_revision_root* | Get a handle on a specific version of the repository root |
| *svn_fs_txn_root* | Creates and returns a transaction root |
| *svn_fs_youngest_rev* | Returns the number of the youngest revision in the filesystem |
| *svn_import* | Imports an unversioned path into a repository |
| *svn_log* | Returns the commit log messages of a repository URL |
| *svn_ls* | Returns list of directory contents in repository URL, optionally at revision number |
| *svn_mkdir* | Creates a directory in a working copy or repository |
| *svn_repos_create* | Create a new subversion repository at path |
| *svn_repos_fs* | Gets a handle on the filesystem for a repository |
| *svn_repos_fs_begin_txn_for_commit* | Create a new transaction |
| *svn_repos_fs_commit_txn* | Commits a transaction and returns the new revision |
| *svn_repos_hotcopy* | Make a hot-copy of the repos at repospath; copy it to destpath |
| *svn_repos_open* | Open a shared lock on a repository. |
| *svn_repos_recover* | Run recovery procedures on the repository located at path. |
| *svn_revert* | Revert changes to the working copy |
| *svn_status* | Returns the status of working copy files and directories |
| *svn_update* | Update working copy |
| *sybase_affected_rows* | Gets number of affected rows in last query |
| *sybase_close* | Closes a Sybase connection |
| *sybase_connect* | Opens a Sybase server connection |
| *sybase_data_seek* | Moves internal row pointer |
| *sybase_deadlock_retry_count* | Sets the deadlock retry count |
| *sybase_fetch_array* | Fetch row as array |
| *sybase_fetch_assoc* | Fetch a result row as an associative array |
| *sybase_fetch_field* | Get field information from a result |
| *sybase_fetch_object* | Fetch a row as an object |
| *sybase_fetch_row* | Get a result row as an enumerated array |
| *sybase_field_seek* | Sets field offset |
| *sybase_free_result* | Frees result memory |
| *sybase_get_last_message* | Returns the last message from the server |
| *sybase_min_client_severity* | Sets minimum client severity |
| *sybase_min_error_severity* | Sets minimum error severity |
| *sybase_min_message_severity* | Sets minimum message severity |
| *sybase_min_server_severity* | Sets minimum server severity |
| *sybase_num_fields* | Gets the number of fields in a result set |
| *sybase_num_rows* | Get number of rows in a result set |
| *sybase_pconnect* | Open persistent Sybase connection |
| *sybase_query* | Sends a Sybase query |
| *sybase_result* | Get result data |
| *sybase_select_db* | Selects a Sybase database |
| *sybase_set_message_handler* | Sets the handler called when a server message is raised |
| *sybase_unbuffered_query* | Send a Sybase query and do not block |
| *symlink* | 建立符号连接 |

| | |
|---|---|
| *syslog* | Generate a system log message |
| *system* | 执行外部程序，并且显示输出 |
| *sys_getloadavg* | 获取系统的负载（load average） |
| *sys_get_temp_dir* | 返回用于临时文件的目录 |

# t

| *函数* | 说明 |
|---|---|
| *taint* | Taint a string |
| *tan* | 正切 |
| *tanh* | 双曲正切 |
| *tcpwrap_check* | Performs a tcpwrap check |
| *tempnam* | 建立一个具有唯一文件名的文件 |
| *textdomain* | Sets the default domain |
| *tidy_access_count* | Returns the Number of Tidy accessibility warnings encountered for specified document |
| *tidy_config_count* | Returns the Number of Tidy configuration errors encountered for specified document |
| *tidy_error_count* | Returns the Number of Tidy errors encountered for specified document |
| *tidy_get_output* | Return a string representing the parsed tidy markup |
| *tidy_load_config* | Load an ASCII Tidy configuration file with the specified encoding |
| *tidy_reset_config* | Restore Tidy configuration to default values |
| *tidy_save_config* | Save current settings to named file |
| *tidy_setopt* | Updates the configuration settings for the specified tidy document |
| *tidy_set_encoding* | Set the input/output character encoding for parsing markup |
| *tidy_warning_count* | Returns the Number of Tidy warnings encountered for specified document |
| *time* | 返回当前的 Unix 时间戳 |
| *timezone_name_from_abbr* | Returns the timezone name from abbreviation |
| *timezone_version_get* | Gets the version of the timezonedb |
| *time_nanosleep* | 延缓执行若干秒和纳秒 |
| *time_sleep_until* | 使脚本睡眠到指定的时间为止。 |
| *tmpfile* | 建立一个临时文件 |
| *token_get_all* | 将提供的源码按 PHP 标记进行分割 |
| *token_name* | 获取提供的 PHP 解析器代号的符号名称 |
| *touch* | 设定文件的访问和修改时间 |
| *trader_acos* | Vector Trigonometric ACos |
| *trader_ad* | Chaikin A/D Line |
| *trader_add* | Vector Arithmetic Add |
| *trader_adosc* | Chaikin A/D Oscillator |
| *trader_adx* | Average Directional Movement Index |
| *trader_adxr* | Average Directional Movement Index Rating |
| *trader_apo* | Absolute Price Oscillator |

| | |
|---|---|
| *trader_aroon* | Aroon |
| *trader_aroonosc* | Aroon Oscillator |
| *trader_asin* | Vector Trigonometric ASin |
| *trader_atan* | Vector Trigonometric ATan |
| *trader_atr* | Average True Range |
| *trader_avgprice* | Average Price |
| *trader_bbands* | Bollinger Bands |
| *trader_beta* | Beta |
| *trader_bop* | Balance Of Power |
| *trader_cci* | Commodity Channel Index |
| *trader_cdl2crows* | Two Crows |
| *trader_cdl3blackcrows* | Three Black Crows |
| *trader_cdl3inside* | Three Inside Up/Down |
| *trader_cdl3linestrike* | Three-Line Strike |
| *trader_cdl3outside* | Three Outside Up/Down |
| *trader_cdl3starsinsouth* | Three Stars In The South |
| *trader_cdl3whitesoldiers* | Three Advancing White Soldiers |
| *trader_cdlabandonedbaby* | Abandoned Baby |
| *trader_cdladvanceblock* | Advance Block |
| *trader_cdlbelthold* | Belt-hold |
| *trader_cdlbreakaway* | Breakaway |
| *trader_cdlclosingmarubozu* | Closing Marubozu |
| *trader_cdlconcealbabyswall* | Concealing Baby Swallow |
| *trader_cdlcounterattack* | Counterattack |
| *trader_cdldarkcloudcover* | Dark Cloud Cover |
| *trader_cdldoji* | Doji |
| *trader_cdldojistar* | Doji Star |
| *trader_cdldragonflydoji* | Dragonfly Doji |
| *trader_cdlengulfing* | Engulfing Pattern |
| *trader_cdleveningdojistar* | Evening Doji Star |
| *trader_cdleveningstar* | Evening Star |
| *trader_cdlgapsidesidewhite* | Up/Down-gap side-by-side white lines |
| *trader_cdlgravestonedoji* | Gravestone Doji |
| *trader_cdlhammer* | Hammer |
| *trader_cdlhangingman* | Hanging Man |
| *trader_cdlharami* | Harami Pattern |
| *trader_cdlharamicross* | Harami Cross Pattern |
| *trader_cdlhighwave* | High-Wave Candle |
| *trader_cdlhikkake* | Hikkake Pattern |
| *trader_cdlhikkakemod* | Modified Hikkake Pattern |
| *trader_cdlhomingpigeon* | Homing Pigeon |
| *trader_cdlidentical3crows* | Identical Three Crows |
| *trader_cdlinneck* | In-Neck Pattern |
| *trader_cdlinvertedhammer* | Inverted Hammer |

| | |
|---|---|
| *trader_cdlkicking* | Kicking |
| *trader_cdlkickingbylength* | Kicking |
| *trader_cdlladderbottom* | Ladder Bottom |
| *trader_cdllongleggeddoji* | Long Legged Doji |
| *trader_cdllongline* | Long Line Candle |
| *trader_cdlmarubozu* | Marubozu |
| *trader_cdlmatchinglow* | Matching Low |
| *trader_cdlmathold* | Mat Hold |
| *trader_cdlmorningdojistar* | Morning Doji Star |
| *trader_cdlmorningstar* | Morning Star |
| *trader_cdlonneck* | On-Neck Pattern |
| *trader_cdlpiercing* | Piercing Pattern |
| *trader_cdlrickshawman* | Rickshaw Man |
| *trader_cdlrisefall3methods* | Rising/Falling Three Methods |
| *trader_cdlseparatinglines* | Separating Lines |
| *trader_cdlshootingstar* | Shooting Star |
| *trader_cdlshortline* | Short Line Candle |
| *trader_cdlspinningtop* | Spinning Top |
| *trader_cdlstalledpattern* | Stalled Pattern |
| *trader_cdlsticksandwich* | Stick Sandwich |
| *trader_cdltakuri* | Takuri (Dragonfly Doji with very long lower shadow) |
| *trader_cdltasukigap* | Tasuki Gap |
| *trader_cdlthrusting* | Thrusting Pattern |
| *trader_cdltristar* | Tristar Pattern |
| *trader_cdlunique3river* | Unique 3 River |
| *trader_cdlupsidegap2crows* | Upside Gap Two Crows |
| *trader_cdlxsidegap3methods* | Upside/Downside Gap Three Methods |
| *trader_ceil* | Vector Ceil |
| *trader_cmo* | Chande Momentum Oscillator |
| *trader_correl* | Pearson's Correlation Coefficient (r) |
| *trader_cos* | Vector Trigonometric Cos |
| *trader_cosh* | Vector Trigonometric Cosh |
| *trader_dema* | Double Exponential Moving Average |
| *trader_div* | Vector Arithmetic Div |
| *trader_dx* | Directional Movement Index |
| *trader_ema* | Exponential Moving Average |
| *trader_errno* | Get error code |
| *trader_exp* | Vector Arithmetic Exp |
| *trader_floor* | Vector Floor |
| *trader_get_compat* | Get compatibility mode |
| *trader_get_unstable_period* | Get unstable period |
| *trader_ht_dcperiod* | Hilbert Transform |
| *trader_ht_dcphase* | Hilbert Transform |
| *trader_ht_phasor* | Hilbert Transform |

| | |
|---|---|
| *trader_ht_sine* | Hilbert Transform |
| *trader_ht_trendline* | Hilbert Transform |
| *trader_ht_trendmode* | Hilbert Transform |
| *trader_kama* | Kaufman Adaptive Moving Average |
| *trader_linearreg* | Linear Regression |
| *trader_linearreg_angle* | Linear Regression Angle |
| *trader_linearreg_intercept* | Linear Regression Intercept |
| *trader_linearreg_slope* | Linear Regression Slope |
| *trader_ln* | Vector Log Natural |
| *trader_log10* | Vector Log10 |
| *trader_ma* | Moving average |
| *trader_macd* | Moving Average Convergence/Divergence |
| *trader_macdext* | MACD with controllable MA type |
| *trader_macdfix* | Moving Average Convergence/Divergence Fix 12/26 |
| *trader_mama* | MESA Adaptive Moving Average |
| *trader_mavp* | Moving average with variable period |
| *trader_max* | Highest value over a specified period |
| *trader_maxindex* | Index of highest value over a specified period |
| *trader_medprice* | Median Price |
| *trader_mfi* | Money Flow Index |
| *trader_midpoint* | MidPoint over period |
| *trader_midprice* | Midpoint Price over period |
| *trader_min* | Lowest value over a specified period |
| *trader_minindex* | Index of lowest value over a specified period |
| *trader_minmax* | Lowest and highest values over a specified period |
| *trader_minmaxindex* | Indexes of lowest and highest values over a specified period |
| *trader_minus_di* | Minus Directional Indicator |
| *trader_minus_dm* | Minus Directional Movement |
| *trader_mom* | Momentum |
| *trader_mult* | Vector Arithmetic Mult |
| *trader_natr* | Normalized Average True Range |
| *trader_obv* | On Balance Volume |
| *trader_plus_di* | Plus Directional Indicator |
| *trader_plus_dm* | Plus Directional Movement |
| *trader_ppo* | Percentage Price Oscillator |
| *trader_roc* | Rate of change : ((price/prevPrice)-1)*100 |
| *trader_rocp* | Rate of change Percentage: (price-prevPrice)/prevPrice |
| *trader_rocr* | Rate of change ratio: (price/prevPrice) |
| *trader_rocr100* | Rate of change ratio 100 scale: (price/prevPrice)*100 |
| *trader_rsi* | Relative Strength Index |
| *trader_sar* | Parabolic SAR |
| *trader_sarext* | Parabolic SAR |
| *trader_set_compat* | Set compatibility mode |
| *trader_set_unstable_period* | Set unstable period |

| *trader_sin* | Vector Trigonometric Sin |
|---|---|
| *trader_sinh* | Vector Trigonometric Sinh |
| *trader_sma* | Simple Moving Average |
| *trader_sqrt* | Vector Square Root |
| *trader_stddev* | Standard Deviation |
| *trader_stoch* | Stochastic |
| *trader_stochf* | Stochastic Fast |
| *trader_stochrsi* | Stochastic Relative Strength Index |
| *trader_sub* | Vector Arithmetic Subtraction |
| *trader_sum* | Summation |
| *trader_t3* | Triple Exponential Moving Average (T3) |
| *trader_tan* | Vector Trigonometric Tan |
| *trader_tanh* | Vector Trigonometric Tanh |
| *trader_tema* | Triple Exponential Moving Average |
| *trader_trange* | True Range |
| *trader_trima* | Triangular Moving Average |
| *trader_trix* | 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA |
| *trader_tsf* | Time Series Forecast |
| *trader_typprice* | Typical Price |
| *trader_ultosc* | Ultimate Oscillator |
| *trader_var* | Variance |
| *trader_wclprice* | Weighted Close Price |
| *trader_willr* | Williams' %R |
| *trader_wma* | Weighted Moving Average |
| *trait_exists* | 检查指定的 trait 是否存在 |
| *trigger_error* | 产生一个用户级别的 error/warning/notice 信息 |
| *trim* | 去除字符串首尾处的空白字符（或者其他字符） |

# u

| *函数* | 说明 |
|---|---|
| *uasort* | 使用用户自定义的比较函数对数组中的值进行排序并保持索引关联 |
| *ucfirst* | 将字符串的首字母转换为大写 |
| *ucwords* | 将字符串中每个单词的首字母转换为大写 |
| *udm_add_search_limit* | Add various search limits |
| *udm_alloc_agent* | Allocate mnoGoSearch session |
| *udm_alloc_agent_array* | Allocate mnoGoSearch session |
| *udm_api_version* | Get mnoGoSearch API version |
| *udm_cat_list* | Get all the categories on the same level with the current one |
| *udm_cat_path* | Get the path to the current category |
| *udm_check_charset* | Check if the given charset is known to mnogosearch |
| *udm_clear_search_limits* | Clear all mnoGoSearch search restrictions |
| *udm_crc32* | Return CRC32 checksum of given string |

| | |
|---|---|
| *udm_errno* | Get mnoGoSearch error number |
| *udm_error* | Get mnoGoSearch error message |
| *udm_find* | Perform search |
| *udm_free_agent* | Free mnoGoSearch session |
| *udm_free_ispell_data* | Free memory allocated for ispell data |
| *udm_free_res* | Free mnoGoSearch result |
| *udm_get_doc_count* | Get total number of documents in database |
| *udm_get_res_field* | Fetch a result field |
| *udm_get_res_param* | Get mnoGoSearch result parameters |
| *udm_hash32* | Return Hash32 checksum of given string |
| *udm_load_ispell_data* | Load ispell data |
| *udm_set_agent_param* | Set mnoGoSearch agent session parameters |
| *uksort* | 使用用户自定义的比较函数对数组中的键名进行排序 |
| *umask* | 改变当前的 umask |
| *uniqid* | 生成一个唯一 ID |
| *unixtojd* | 转变 Unix 时间戳为 Julian Day 计数 |
| *unlink* | 删除文件 |
| *unpack* | Unpack data from binary string |
| *unregister_tick_function* | De-register a function for execution on each tick |
| *unserialize* | 从已存储的表示中创建 PHP 的值 |
| *unset* | 释放给定的变量 |
| *untaint* | Untaint strings |
| *uopz_backup* | Backup a function |
| *uopz_compose* | Compose a class |
| *uopz_copy* | Copy a function |
| *uopz_delete* | Delete a function |
| *uopz_extend* | Extend a class at runtime |
| *uopz_flags* | Get or set flags on function or class |
| *uopz_function* | Creates a function at runtime |
| *uopz_implement* | Implements an interface at runtime |
| *uopz_overload* | Overload a VM opcode |
| *uopz_redefine* | Redefine a constant |
| *uopz_rename* | Rename a function at runtime |
| *uopz_restore* | Restore a previously backed up function |
| *uopz_undefine* | Undefine a constant |
| *urldecode* | 解码已编码的 URL 字符串 |
| *urlencode* | 编码 URL 字符串 |
| *user_error* | trigger_error 的别名 |
| *use_soap_error_handler* | Set whether to use the SOAP error handler |
| *usleep* | 以指定的微秒数延迟执行 |
| *usort* | 使用用户自定义的比较函数对数组中的值进行排序 |
| *utf8_decode* | 将用 UTF-8 方式编码的 ISO-8859-1 字符串转换成单字节的 ISO-8859-1 字符串。 |
| *utf8_encode* | 将 ISO-8859-1 编码的字符串转换为 UTF-8 编码 |

# V

| 函数 | 说明 |
|---|---|
| *variant_abs* | Returns the absolute value of a variant |
| *variant_add* | "Adds" two variant values together and returns the result |
| *variant_and* | Performs a bitwise AND operation between two variants |
| *variant_cast* | Convert a variant into a new variant object of another type |
| *variant_cat* | concatenates two variant values together and returns the result |
| *variant_cmp* | Compares two variants |
| *variant_date_from_timestamp* | Returns a variant date representation of a Unix timestamp |
| *variant_date_to_timestamp* | Converts a variant date/time value to Unix timestamp |
| *variant_div* | Returns the result from dividing two variants |
| *variant_eqv* | Performs a bitwise equivalence on two variants |
| *variant_fix* | Returns the integer portion of a variant |
| *variant_get_type* | Returns the type of a variant object |
| *variant_idiv* | Converts variants to integers and then returns the result from dividing them |
| *variant_imp* | Performs a bitwise implication on two variants |
| *variant_int* | Returns the integer portion of a variant |
| *variant_mod* | Divides two variants and returns only the remainder |
| *variant_mul* | Multiplies the values of the two variants |
| *variant_neg* | Performs logical negation on a variant |
| *variant_not* | Performs bitwise not negation on a variant |
| *variant_or* | Performs a logical disjunction on two variants |
| *variant_pow* | Returns the result of performing the power function with two variants |
| *variant_round* | Rounds a variant to the specified number of decimal places |
| *variant_set* | Assigns a new value for a variant object |
| *variant_set_type* | Convert a variant into another type "in-place" |
| *variant_sub* | Subtracts the value of the right variant from the left variant value |
| *variant_xor* | Performs a logical exclusion on two variants |
| *var_dump* | 打印变量的相关信息 |
| *var_export* | 输出或返回一个变量的字符串表示 |
| *version_compare* | 对比两个「PHP 规范化」的版本数字字符串 |
| *vfprintf* | 将格式化字符串写入流 |
| *virtual* | 执行 Apache 子请求 |
| *vpopmail_add_alias_domain* | Add an alias for a virtual domain |
| *vpopmail_add_alias_domain_ex* | Add alias to an existing virtual domain |
| *vpopmail_add_domain* | Add a new virtual domain |
| *vpopmail_add_domain_ex* | Add a new virtual domain |
| *vpopmail_add_user* | Add a new user to the specified virtual domain |
| *vpopmail_alias_add* | Insert a virtual alias |

| vpopmail_alias_del | Deletes all virtual aliases of a user |
|---|---|
| vpopmail_alias_del_domain | Deletes all virtual aliases of a domain |
| vpopmail_alias_get | Get all lines of an alias for a domain |
| vpopmail_alias_get_all | Get all lines of an alias for a domain |
| vpopmail_auth_user | Attempt to validate a username/domain/password |
| vpopmail_del_domain | Delete a virtual domain |
| vpopmail_del_domain_ex | Delete a virtual domain |
| vpopmail_del_user | Delete a user from a virtual domain |
| vpopmail_error | Get text message for last vpopmail error |
| vpopmail_passwd | Change a virtual user's password |
| vpopmail_set_user_quota | Sets a virtual user's quota |
| vprintf | 输出格式化字符串 |
| vsprintf | 返回格式化字符串 |

## W

| 函数 | 说明 |
|---|---|
| wddx_add_vars | Add variables to a WDDX packet with the specified ID |
| wddx_deserialize | Unserializes a WDDX packet |
| wddx_packet_end | Ends a WDDX packet with the specified ID |
| wddx_packet_start | Starts a new WDDX packet with structure inside it |
| wddx_serialize_value | Serialize a single value into a WDDX packet |
| wddx_serialize_vars | Serialize variables into a WDDX packet |
| win32_continue_service | Resumes a paused service |
| win32_create_service | Creates a new service entry in the SCM database |
| win32_delete_service | Deletes a service entry from the SCM database |
| win32_get_last_control_message | Returns the last control message that was sent to this service |
| win32_pause_service | Pauses a service |
| win32_ps_list_procs | List running processes |
| win32_ps_stat_mem | Stat memory utilization |
| win32_ps_stat_proc | Stat process |
| win32_query_service_status | Queries the status of a service |
| win32_set_service_status | Update the service status |
| win32_start_service | Starts a service |
| win32_start_service_ctrl_dispatcher | Registers the script with the SCM, so that it can act as the service with the given name |
| win32_stop_service | Stops a service |
| wincache_fcache_fileinfo | Retrieves information about files cached in the file cache |
| wincache_fcache_meminfo | Retrieves information about file cache memory usage |
| wincache_lock | Acquires an exclusive lock on a given key |
| wincache_ocache_fileinfo | Retrieves information about files cached in the opcode cache |
| wincache_ocache_meminfo | Retrieves information about opcode cache memory usage |
| wincache_refresh_if_changed | Refreshes the cache entries for the cached files |
| wincache_rplist_fileinfo | Retrieves information about resolve file path cache |

| | |
|---|---|
| *wincache_rplist_meminfo* | Retrieves information about memory usage by the resolve file path cache |
| *wincache_scache_info* | Retrieves information about files cached in the session cache |
| *wincache_scache_meminfo* | Retrieves information about session cache memory usage |
| *wincache_ucache_add* | Adds a variable in user cache only if variable does not already exist in the cache |
| *wincache_ucache_cas* | Compares the variable with old value and assigns new value to it |
| *wincache_ucache_clear* | Deletes entire content of the user cache |
| *wincache_ucache_dec* | Decrements the value associated with the key |
| *wincache_ucache_delete* | Deletes variables from the user cache |
| *wincache_ucache_exists* | Checks if a variable exists in the user cache |
| *wincache_ucache_get* | Gets a variable stored in the user cache |
| *wincache_ucache_inc* | Increments the value associated with the key |
| *wincache_ucache_info* | Retrieves information about data stored in the user cache |
| *wincache_ucache_meminfo* | Retrieves information about user cache memory usage |
| *wincache_ucache_set* | Adds a variable in user cache and overwrites a variable if it already exists in the cache |
| *wincache_unlock* | Releases an exclusive lock on a given key |
| *wordwrap* | 打断字符串为指定数量的字串 |

# X

| *函数* | 说明 |
|---|---|
| *xattr_get* | Get an extended attribute |
| *xattr_list* | Get a list of extended attributes |
| *xattr_remove* | Remove an extended attribute |
| *xattr_set* | Set an extended attribute |
| *xattr_supported* | Check if filesystem supports extended attributes |
| *xdiff_file_bdiff* | Make binary diff of two files |
| *xdiff_file_bdiff_size* | Read a size of file created by applying a binary diff |
| *xdiff_file_bpatch* | Patch a file with a binary diff |
| *xdiff_file_diff* | Make unified diff of two files |
| *xdiff_file_diff_binary* | Alias of xdiff_file_bdiff |
| *xdiff_file_merge3* | Merge 3 files into one |
| *xdiff_file_patch* | Patch a file with an unified diff |
| *xdiff_file_patch_binary* | Alias of xdiff_file_bpatch |
| *xdiff_file_rabdiff* | Make binary diff of two files using the Rabin's polynomial fingerprinting algorithm |
| *xdiff_string_bdiff* | Make binary diff of two strings |
| *xdiff_string_bdiff_size* | Read a size of file created by applying a binary diff |
| *xdiff_string_bpatch* | Patch a string with a binary diff |
| *xdiff_string_diff* | Make unified diff of two strings |
| *xdiff_string_diff_binary* | Alias of xdiff_string_bdiff |
| *xdiff_string_merge3* | Merge 3 strings into one |

| | |
|---|---|
| *xdiff_string_patch* | Patch a string with an unified diff |
| *xdiff_string_patch_binary* | Alias of xdiff_string_bpatch |
| *xdiff_string_rabdiff* | Make binary diff of two strings using the Rabin's polynomial fingerprinting algorithm |
| *xhprof_disable* | 停止 xhprof 分析器 |
| *xhprof_enable* | 启动 xhprof 性能分析器 |
| *xhprof_sample_disable* | 停止 xhprof 性能采样分析器 |
| *xhprof_sample_enable* | 以采样模式启动 XHProf 性能分析 |
| *xmlrpc_decode* | 将 XML 译码为 PHP 本身的类型 |
| *xmlrpc_decode_request* | 将 XML 译码为 PHP 本身的类型 |
| *xmlrpc_encode* | 为 PHP 的值生成 XML |
| *xmlrpc_encode_request* | 为 PHP 的值生成 XML |
| *xmlrpc_get_type* | 为 PHP 的值获取 xmlrpc 的类型 |
| *xmlrpc_is_fault* | Determines if an array value represents an XMLRPC fault |
| *xmlrpc_parse_method_descriptions* | 将 XML 译码成方法描述的列表 |
| *xmlrpc_server_add_introspection_data* | 添加自我描述的文档 |
| *xmlrpc_server_call_method* | 解析 XML 请求同时调用方法 |
| *xmlrpc_server_create* | 创建一个 xmlrpc 服务端 |
| *xmlrpc_server_destroy* | 销毁服务端资源 |
| *xmlrpc_server_register_introspection_callback* | 注册一个 PHP 函数用于生成文档 |
| *xmlrpc_server_register_method* | 注册一个 PHP 函数用于匹配 xmlrpc 方法名 |
| *xmlrpc_set_type* | 为一个 PHP 字符串值设置 xmlrpc 的类型、base64 或日期时间 |
| *xml_error_string* | 获取 XML 解析器的错误字符串 |
| *xml_get_current_byte_index* | 获取 XML 解析器的当前字节索引 |
| *xml_get_current_column_number* | 获取 XML 解析器的当前列号 |
| *xml_get_current_line_number* | 获取 XML 解析器的当前行号 |
| *xml_get_error_code* | 获取 XML 解析器错误代码 |
| *xml_parse* | 开始解析一个 XML 文档 |
| *xml_parser_create* | 建立一个 XML 解析器 |
| *xml_parser_create_ns* | 生成一个支持命名空间的 XML 解析器 |
| *xml_parser_free* | 释放指定的 XML 解析器 |
| *xml_parser_get_option* | 从 XML 解析器获取选项设置信息 |
| *xml_parser_set_option* | 为指定 XML 解析进行选项设置 |
| *xml_parse_into_struct* | 将 XML 数据解析到数组中 |
| *xml_set_character_data_handler* | 建立字符数据处理器 |
| *xml_set_default_handler* | 建立默认处理器 |
| *xml_set_element_handler* | 建立起始和终止元素处理器 |
| *xml_set_end_namespace_decl_handler* | 建立终止命名空间声明处理器 |
| *xml_set_external_entity_ref_handler* | 建立外部实体指向处理器 |
| *xml_set_notation_decl_handler* | 建立注释声明处理器 |
| *xml_set_object* | 在对象中使用 XML 解析器 |
| *xml_set_processing_instruction_handler* | 建立处理指令（PI）处理器 |
| *xml_set_start_namespace_decl_handler* | 建立起始命名空间声明处理器 |

| | |
|---|---|
| *xml_set_unparsed_entity_decl_handler* | 建立未解析实体定义声明处理器 |

# y

| 函数 | 说明 |
|---|---|
| *yaml_emit* | Returns the YAML representation of a value |
| *yaml_emit_file* | Send the YAML representation of a value to a file |
| *yaml_parse* | Parse a YAML stream |
| *yaml_parse_file* | Parse a YAML stream from a file |
| *yaml_parse_url* | Parse a Yaml stream from a URL |
| *yaz_addinfo* | Returns additional error information |
| *yaz_ccl_conf* | Configure CCL parser |
| *yaz_ccl_parse* | Invoke CCL Parser |
| *yaz_close* | Close YAZ connection |
| *yaz_connect* | Prepares for a connection to a Z39.50 server |
| *yaz_database* | Specifies the databases within a session |
| *yaz_element* | Specifies Element-Set Name for retrieval |
| *yaz_errno* | Returns error number |
| *yaz_error* | Returns error description |
| *yaz_es* | Prepares for an Extended Service Request |
| *yaz_es_result* | Inspects Extended Services Result |
| *yaz_get_option* | Returns value of option for connection |
| *yaz_hits* | Returns number of hits for last search |
| *yaz_itemorder* | Prepares for Z39.50 Item Order with an ILL-Request package |
| *yaz_present* | Prepares for retrieval (Z39.50 present) |
| *yaz_range* | Specifies a range of records to retrieve |
| *yaz_record* | Returns a record |
| *yaz_scan* | Prepares for a scan |
| *yaz_scan_result* | Returns Scan Response result |
| *yaz_schema* | Specifies schema for retrieval |
| *yaz_search* | Prepares for a search |
| *yaz_set_option* | Sets one or more options for connection |
| *yaz_sort* | Sets sorting criteria |
| *yaz_syntax* | Specifies the preferred record syntax for retrieval |
| *yaz_wait* | Wait for Z39.50 requests to complete |
| *yp_all* | Traverse the map and call a function on each entry |
| *yp_cat* | Return an array containing the entire map |
| *yp_errno* | Returns the error code of the previous operation |
| *yp_err_string* | Returns the error string associated with the given error code |
| *yp_first* | Returns the first key-value pair from the named map |
| *yp_get_default_domain* | Fetches the machine's default NIS domain |
| *yp_master* | Returns the machine name of the master NIS server for a map |
| *yp_match* | Returns the matched line |
| *yp_next* | Returns the next key-value pair in the named map |

| | |
|---|---|
| *yp_order* | Returns the order number for a map |

**Z**

| *函数* | 说明 |
|---|---|
| *zend_logo_guid* | 获取 Zend guid |
| *zend_thread_id* | 返回当前线程的唯一识别符 |
| *zend_version* | 获取当前 Zend 引擎的版本 |
| *zip_close* | 关闭一个 ZIP 档案文件 |
| *zip_entry_close* | 关闭目录项 |
| *zip_entry_compressedsize* | 检索目录项压缩过后的大小 |
| *zip_entry_compressionmethod* | 检索目录实体的压缩方法 |
| *zip_entry_filesize* | 检索目录实体的实际大小 |
| *zip_entry_name* | 检索目录项的名称 |
| *zip_entry_open* | 打开用于读取的目录实体 |
| *zip_entry_read* | 读取一个打开了的压缩目录实体 |
| *zip_open* | 打开 ZIP 存档文件 |
| *zip_read* | 读取 ZIP 存档文件中下一项 |
| *zlib_decode* | Uncompress any raw/gzip/zlib encoded data |
| *zlib_encode* | Compress data with the specified encoding |
| *zlib_get_coding_type* | Returns the coding type used for output compression |