

## Covariant Script 函数及变量列表

### 全局

char 字符类型  
number 数字类型  
boolean 逻辑类型  
string 文字类型  
list 链表类型  
array 数组类型  
pair 映射类型  
hash\_map 哈希表类型  
system 系统命名空间  
runtime 运行时名称空间  
types 类型名称空间  
number to\_integer(var) 将一个变量转换为整数并返回  
string to\_string(var) 将一个变量转换为文字并返回  
boolean is\_char(var) 判断一个变量是否为字符类型  
boolean is\_number(var) 判断一个变量是否为数字类型  
boolean is\_boolean(var) 判断一个变量是否为逻辑类型  
boolean is\_string(var) 判断一个变量是否为文字类型  
boolean is\_list(var) 判断一个变量是否为链表类型  
boolean is\_array(var) 判断一个变量是否为数组类型  
boolean is\_pair(var) 判断一个变量是否为映射类型  
boolean is\_hash\_map(var) 判断一个变量是否为哈希表类型  
var clone(var) 复制一个变量并返回  
void swap(var,var) 交换两个变量的值

### System 名称空间

max 数字类型最大值  
inf 数字类型正无穷  
var input(...) 从标准输入流中获取输入（堵塞，格式化输入）  
void print(...) 向标准输出流中输出内容，仅可输出支持 to\_string 的类型（不换行）  
void println(...) 向标准输出流中输出内容，仅可输出支持 to\_string 的类型（换行）  
string getline() 从标准输入流中获取输入（堵塞，非格式化输入）  
void setprecision(number) 设置输出精度（to\_string 的精度）  
number run(string) 在系统环境中运行一条指令，返回错误码  
string getenv(string) 获取环境变量的值并返回  
void exit(number) 清理资源并退出

### Runtime 名称空间

void info() 输出版本信息

number time() 获取计时器的读数, 单位毫秒  
void delay(number) 使程序暂停一段时间, 单位毫秒  
number rand(number,number) 获取区间内的伪随机数  
number randint(number,number) 获取区间内的伪随机整数  
void error(string) 抛出一个运行时错误  
[namespace] load\_extension(string) 加载一个扩展并返回  
[hash\_value] hash(var) 计算一个变量的哈希值  
[expression] build(string) 构建一个可用于计算的表达式  
var solve([expression]) 计算一个表达式

## Types 名称空间

char 字符类型扩展  
string 文字类型扩展  
list 链表类型扩展  
array 数组类型扩展  
pair 映射类型扩展  
hash\_map 哈希表类型扩展

## 字符类型扩展

boolean isalnum(char) 检查字符是否是字母或数字  
boolean isalpha(char) 检查字符是否是字母  
boolean islower(char) 检查字符是否是小写字母  
boolean isupper(char) 检查字符是否是大写字母  
boolean isdigit(char) 检查字符是否是数字  
boolean iscntrl(char) 检查字符是否是控制字符  
boolean isgraph(char) 检查字符是否是图形字符  
boolean isspace(char) 检查字符是否是空白字符  
boolean isblank(char) 检查字符是否是空格或 **tab**  
boolean isprint(char) 检查字符是否是打印字符  
boolean ispunct(char) 检查字符是否是标点符号  
char tolower(char) 将字符转换为小写  
char toupper(char) 将字符转换为大写

## 文字类型扩展

string append(string,var) 在尾部追加内容  
string insert(string,number,var) 在指定位置处插入内容  
string erase(string,number,number) 将范围内的字符删除  
string replace(string,number,number,var) 将从指定位置开始的指定个数字符替换  
string substr(string,number,number) 从指定位置截取指定长度的子文字  
number find(string,string,number) 从指定位置开始从左向右查找一段文字  
number rfind(string,string,number) 从指定位置开始从右向左查找一段文字  
string cut(string,number) 从尾部删除指定长度的文字  
void clear(string) 清空

number size(string) 获取字符个数

## 链表类型扩展

void push\_front(list,var) 向前部插入数据  
void pop\_front(list) 从前部删除数据  
void push\_back(list,var) 向后部插入数据  
void pop\_back(list) 从后部删除数据  
void remove(list,var) 删除所有与指定变量相等的元素  
var front(list) 获取前部元素  
var back(list) 获取后部元素  
boolean empty(list) 判断是否为空  
void reverse(list) 反转所有元素  
void clear(list) 清空  
number size(list) 获取存储元素的个数

## 数组类型扩展

void push\_front(array,var) 向前部插入数据  
void pop\_front(array) 从前部删除数据  
void push\_back(array,var) 向后部插入数据  
void pop\_back(array) 从后部删除数据  
void clear(array) 清空  
var at(array,number) 获取指定下标元素  
number size(array) 获取存储元素的个数

## 映射类型扩展

var first(pair) 获取第一个元素  
var second(pair) 获取第二个元素

## 哈希表类型扩展

void clear(hash\_map) 清空  
void insert(hash\_map,var,var) 插入一对映射  
void erase(hash\_map,var) 删除键对应的映射  
boolean exist(hash\_map,var) 查找是否存在映射  
var at(hash\_map,var) 访问键对应的值  
number size(hash\_map) 获取存储映射的个数

## 数学扩展

pi 圆周率  
e 自然底数

number abs(number) 绝对值  
number ln(number) 以 e 为底的对数  
number log10(number) 以 10 为底的对数  
number log(number a,number b) 以 a 为底 b 的对数  
number sin(number) 正弦  
number cos(number) 余弦  
number tan(number) 正切  
number asin(number) 反正弦  
number acos(number) 反余弦  
number atan(number) 反正切  
number sqrt(number) 开方  
number root(number a,number b) a 的 b 次方根  
number pow(number a,number b) a 的 b 次方  
number min(number a,number b) a 和 b 的最小值  
number max(number a,number b) a 和 b 的最大值

## 文件扩展

read\_method 读文件  
write\_method 写文件  
[file] open(string path,[method]) 打开一个文件  
boolean is\_open([file]) 判断文件是否打开  
boolean eof([file]) 判断是否到达文件结尾  
string getline([file]) 从文件中获取输入（阻塞，非格式化输入）  
var read([file],...) 从文件中获取输入（阻塞，格式化输入）  
var write([file],...) 向文件中输出内容，仅可输出支持 to\_string 的类型（不换行）

## 图形扩展(Covariant Darwin UCGL 实现)

black 黑色  
white 白色  
red 红色  
green 绿色  
blue 蓝色  
pink 粉色  
yellow 黄色  
cyan 青色  
[pixel] pixel(char,[color] front,[color] back) 创建一个像素  
[drawable] picture(number width,number height) 创建一幅图片  
void load(string path) 加载 Darwin 功能  
void exit(number code) 退出程序并清理资源  
boolean is\_kb\_hit() 判断是否有按键按下  
char get\_kb\_hit() 获取按下的按键  
void fit\_drawable() 使画布适合当前屏幕大小  
[drawable] get\_drawable() 获取画布  
void update\_drawable() 将画布中的内容更新至屏幕上

void set\_frame\_limit(number fps) 设置帧率  
void clear\_drawable([drawable]) 清空画布  
void fill\_drawable([drawable],[pixel]) 填充画布  
void resize\_drawable([drawable],number width,number height) 重新设置画布大小  
number get\_width([drawable]) 获取画布宽度  
number get\_height([drawable]) 获取画布高度  
void draw\_pixel([drawable],number x,number y,[pixel]) 在画布上画点  
void draw\_picture([drawable],number x,number y,[drawable]) 将一幅图片绘制到画布上  
void draw\_line([drawable],number x1,number y1,number x2,number y2,[pixel])  
在画布上画线  
void draw\_rect([drawable],number x,number y,number width,number height,[pixel])  
在画布上绘制线框  
void fill\_rect([drawable],number x,number y,number width,number height,[pixel])  
在画布上填充矩形  
void draw\_triangle([drawable],number x1,number y1,number x2,number y2,number x3,number y3,[pixel])  
在画布上绘制三角形  
void fill\_triangle([drawable],number x1,number y1,number x2,number y2,number x3,number y3,[pixel])  
在画布上填充三角形  
void draw\_string([drawable],number x,number y,string,[pixel]) 在画布上绘制文字  
void message\_box(string title,string message,string button) 弹出一个消息对话框  
var input\_box(string title,string message,string default,boolean format) 弹出一个输入对话框