* 1. **函数定义**

**functon** **函数名()**

--函数体

**end**

* 1. **函数调用**时实参列表外的圆括号**()**：若函数只有一个参数，且此参数为字符串常量或者table构造式，则函数调用时参数列表的圆括号可省略。除此以外的情况，（）不可省略。
  2. **参数传递**：实参和形参的匹配与多重赋值语句类似，多余的实参被忽略，缺少的实参用nil补足。
  3. **多重返回值**：在return后将需要返回的所有值用逗号分隔即可。

原则：函数调用若作为右值的最后一项，则左值要多少，就尽可能给多少；  
若非右值的最后一项，则只返回一个值。

注意：return 语句后面的内容不需要圆括号，使用**圆括号**会强制函数只返回一个值。

* 1. **泛型调用**：以任何实参来调用任何函数，即函数名可变，实参可变。

1. 函数名用一个变量获取；
2. 实参用unpack解包函数获取。将函数需要的所有参数以数组的形式传递给unpack，unpack会返回数组内的所有元素。（整体转换为多个个体）
3. table.unpack (list [, i [, j]])：返回给定列表中的元素。等效于return list[i], list[i+1], ···, list[j]
   1. **变长参数**：函数调用时实参的数量可变。
   2. **形参**使用**...**表示，意为可以接收不同数量的实参。注意这个形参要出现在形参列表的最后一项。函数体内用**表达式...**访问这个变长参数。
   3. 变长参数中包含nil时，需要用select函数访问变长参数。
      * 函数原型：select (index, … )，其中，index是固定参数，表示选择开关；…是变长参数
      * select(‘#’, ...) --返回参数的总数(包括nil)
      * select(n,…) --返回从参数编号n开始的所有参数
   4. **具名参数**：通过名称来指定实参，而不要求实参与形参的按位置传递。需要将所有的实参组织到一个table中，并将这个table作为唯一的实参传给函数。