



Unidad Azcapotzalco

División de Ciencias y Artes para el Diseño

Posgrado en Diseño y Visualización de la Información

Funciones

L. C. C. Eric Omar Torres Velasco

Reporte de lectura para la UEA Introducción a la programación

Profesora:

Dra. Lizbeth Gallardo López

Ciudad de México, México. A 10 junio de 2019

Una función es un bloque de código que sirve para resolver un subproblema del problema. “En los lenguajes de programación de computadoras, las funciones son sinónimo de: subrutinas, procedimientos o métodos” (González, 2007: 69).

Las funciones pueden o no recibir el intercambio de información a través de los parámetros. Estos se relacionan con su posición y el tipo de dato asociado. De modo que, existen los siguiente dos tipos de parámetros: por valor y por referencia. (González, 2007; Gallardo & Lara, 2012).

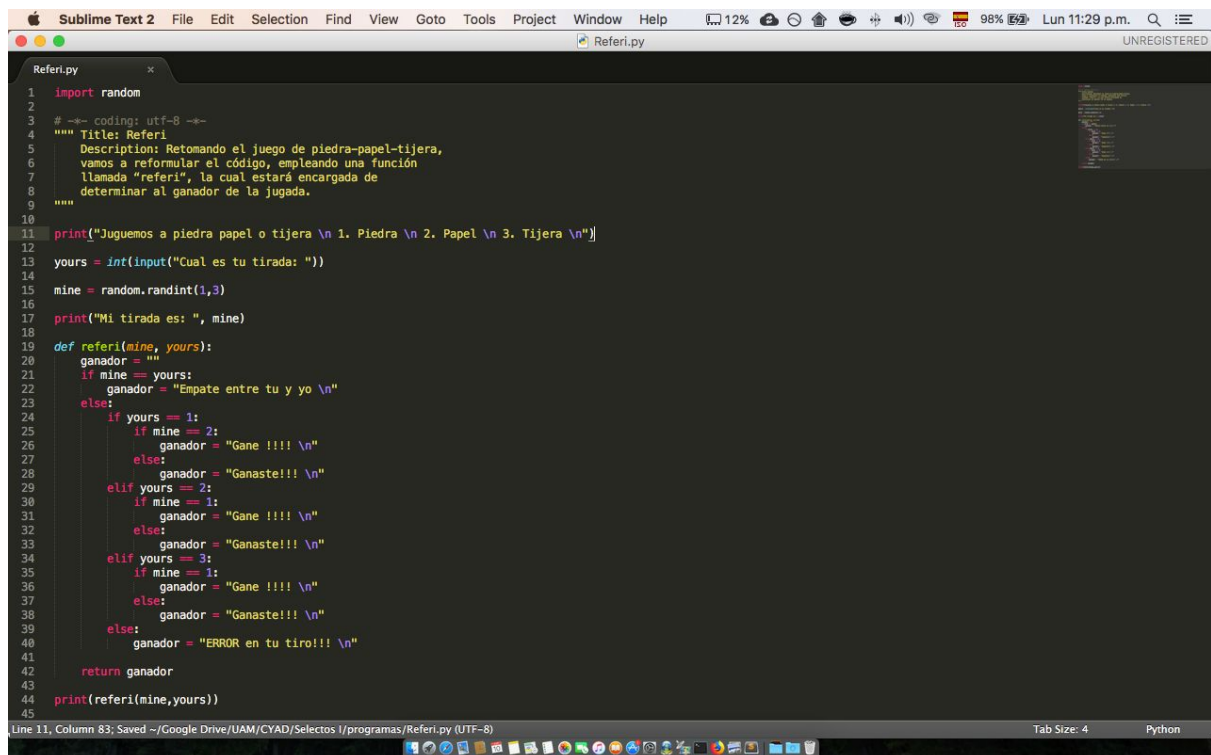
Los parámetros por valor son constantes. En otras palabras, “son copias de la variable que se le da a la función, estos valores no los puede modificar la función; aunque la copia puede cambiar de valor dentro de la función” (González, 2007: 71).

Los parámetros por referencia son variables. Dicho de otro modo, “son la referencia o dirección de memoria de las variables que se le dan a la función; con esto se modifica directamente los valores como parte de la referencia, es decir, la función podrá modificar los valores que se le dieron como argumentos en la invocación (González, 2007: 71).

“Un valor de retorno es el resultado de la evaluación de una expresión realizado dentro del bloque de sentencias de la función y que es devuelto al punto del programa de donde la función fue invocada o llamada (González, 2007: 71). Por lo tanto, una función puede o no devolver un valor.

Para declarar una función en python se utiliza la palabra reservada *def*, después para definir la función significa especificar el grupo de sentencias que realizará cuando se le llame o invoque, esto es, declarar variables, realizar operaciones, regresar valores, etc.. Por lo que, “el llamado a una función implica un «salto» hacia ella para ejecutar su conjunto de instrucciones; al terminar, se devuelve el control de la ejecución a la función que la llamó” (Gallardo & Lara, 2012: 35).

A continuación se propone un ejemplo para expresar una función con intercambio de parámetros por valor y retorno de valor en python que determine el ganador de una jugada de piedra-papel-tijera:



```
1 import random
2
3 # -*- coding: utf-8 -*-
4 """ Title: Referi
5     Description: Retomando el juego de piedra-papel-tijera,
6     vamos a reformular el código, empleando una función
7     llamada "referi", la cual estará encargada de
8     determinar al ganador de la jugada.
9 """
10
11 print("\Juguemos a piedra papel o tijera \n 1. Piedra \n 2. Papel \n 3. Tijera \n")
12
13 yours = int(input("Cual es tu tirada: "))
14
15 mine = random.randint(1,3)
16
17 print("Mi tirada es: ", mine)
18
19 def referi(mine, yours):
20     ganador = ""
21     if mine == yours:
22         ganador = "Empate entre tu y yo \n"
23     else:
24         if yours == 1:
25             if mine == 2:
26                 ganador = "Gane !!!! \n"
27             else:
28                 ganador = "Ganaste!!! \n"
29         elif yours == 2:
30             if mine == 1:
31                 ganador = "Gane !!!! \n"
32             else:
33                 ganador = "Ganaste!!! \n"
34         elif yours == 3:
35             if mine == 1:
36                 ganador = "Gane !!!! \n"
37             else:
38                 ganador = "Ganaste!!! \n"
39         else:
40             ganador = "ERROR en tu tiro!!! \n"
41
42     return ganador
43
44 print(referi(mine,yours))
45
```

Referencias y bibliografía:

Gallardo, Lizbeth & Lara, Pedro. (2012). Notas de programación estructurada. Universidad Autónoma Metropolitana.

González, Oscar. (2007). Introducción a la programación. México: Universidad Autónoma de México.