

Fundamentos de Programación

TEMA: MANEJO DE FUNCIONES 3 Y TUPLAS

Semana 16

OBJETIVO DEL LABORATORIO

Crea procedimientos o funciones para casos de negocio, con parámetros y con retorno más avanzados. Construye tus propias tuplas.

MARCO TEÓRICO

Saber construir las funciones, nos permite realizarlas de una manera más compleja.

Al conocer cómo trabajar con listas; nos hace más factible el uso de las tuplas, que son muy similares, pero difieren en algunas propiedades.

Como fuente de información, se tiene la misma plataforma del Netacad. Tanto para realizar los ejemplos propuestos en la interfaz integrada, así como para repasar lo visto en clase.

RECURSOS

a. Hardware

- Pc Pentium IV a superior
- Conexión de red

b. Software

- Sistema Operativo Windows XP a superior
- Navegador Chrome o Firefox
- Edube Sandbox de Python desde Netacad

PROCEDIMIENTO

1. Funciones

Supongamos que nos piden que construyamos el teorema de Pitágoras, para calcular el valor de la hipotenusa; teniendo como dato los catetos. Haremos lo siguiente.

Primero, construimos la función:

```
>>> def hipo(a,b):  
>>>     c = ((a ** 2) + (b ** 2)) ** 0.5  
>>>     print(c)
```

¡Listo! Al ya tener la función, pasamos a crear el ingreso de datos

En la consola interactiva, podemos escribir el nombre de un objeto para ver su valor.

```
>>> lado1=float(input("Ingrese el valor del lado 1: "))  
>>> lado2=float(input("Ingrese el valor del lado 2: "))
```

Finalmente, realizamos su invocación:

```
>>> hipo(lado1,lado2)
```



2. Tuplas

Las tuplas, al igual que las listas, son colecciones ordenadas. No obstante, a diferencia de éstas, son inmutables. Es decir, una vez asignados los elementos, no pueden ser alterados. En términos funcionales, podría decirse que las tuplas son un subconjunto de las listas, por cuanto soportan las operaciones con índices para acceder a sus elementos, pero no así las de asignación.

```
# Indicamos los elementos entre paréntesis.
>>> lenguajes = ("Python", "Java", "C", "C++")
>>> lenguajes[0]
'Python'
>>> lenguajes[-1]
'C++'
```

Un intento por remover un elemento de una tupla o asignarle un nuevo valor arrojará un error.

Ahora bien, si las listas hacen todo lo que hace una tupla, ¿para qué, entonces, existen las tuplas? Sencillamente porque una tupla, justamente por carecer de todas las funcionalidades de mutación de elementos, es un objeto más liviano y que se crea más rápido.

La función del programador será la de dirimir si para una determinada colección ordenada de objetos le conviene más una lista o una tupla, según sus elementos deban ser alterados en el futuro o no.

CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA

- Al construir nuestras funciones, hace posible que podamos trabajarlas en función de algunos parámetros, para su posterior procesamiento.
- Trabajar con tuplas, nos da cierto entorno similar a las listas; pero con distintas funcionalidades que no las soportan las listas en sí.
- Se recomienda usar el Edube SandBox para realizar nuestros ejercicios sin necesidad de instalar algún software adicional.
- Así como en algún momento se solicita cálculos con números enteros, sería factible buscar realizar lo mismo; pero con números reales.

ACTIVIDAD VIRTUAL

1. Observa y analiza el siguiente video: <https://www.youtube.com/watch?v=ejEyvxCBsWI>, y responde las siguientes preguntas:
 - ¿Cuáles son las funciones que aparecen en el video?
 - Realiza esos ejemplos en Python
2. Observa y analiza el siguiente video: <https://www.youtube.com/watch?v=Ufqh8aoR9hE>, y responde las siguientes preguntas:
 - ¿Cuáles son las tuplas que aparecen en el video?
 - Realiza esos ejemplos en Python

