Fundamentos de Programación

TEMA: MANEJO DE FUNCIONES 1
Semana 13

OBJETIVO DEL LABORATORIO

Crea procedimientos o funciones para casos de negocio.

MARCO TEÓRICO
A través de los diversos temas vistos en PYTHON, hemos usado funciones predeterminadas; sin embargo, nosotros podemos crear la función que queramos en base a un escenario en particular.
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 básicas

Existe una distinción teórica que me gusta remarcar acerca de las funciones que, si bien al momento de crear una no es relevante, considero importante hacer. Hay un tipo de función que puede categorizarse junto con los condicionales y los bucles: esto es, sirve para agrupar un conjunto de instrucciones bajo un mismo nombre para evitar repetirlas a lo largo del código. El otro, es el concepto de función tal como se conoce en matemática: una función obtiene un valor o un conjunto de valores de entrada, aplica una serie de operaciones y retorna un valor o un conjunto de valores de salida, un resultado.

Como sea, empecemos por un ejemplo simple. Definamos una función `dup(n)`, muy al estilo matemático, que tome como argumento un número y retorne el doble.

```
def dup(n):
    return n * 2
```

Introdujimos dos palabras reservadas nuevas. def es empleada siempre que se quiera crear una nueva función, seguida de su nombre (que, al igual que las variables, se escribe en minúscula y separada por guiones bajos) y los argumentos entre paréntesis y separados por comas. return debe estar sucedida por una expresión que será el valor de retorno de la función. Iremos aclarando estos conceptos.

Por el momento, hagamos una prueba:

```
# Imprime 10 en pantalla.
print(dup(5))
```

Ahora consideremos esta otra definición.

```
def saludar(nombre):
    print("Hola", nombre)


saludar("mundo")
```

En este caso, nuestra función saludar() no retorna ningún valor. Aunque, en efecto, Python le asigna un valor de retorno por defecto llamado None. Se trata de un tipo de dato como los que hemos estado trabajando, pero un tanto más particular, ya que intenta indicar que una variable está vacía.

```
>>> a = None
>>> b = 3.14
>>> a is None
True
>>> b is not None
True
```

La única diferencia, como se observa, es que para realizar comparaciones respecto de `None` utilizamos las palabras reservadas `is` e `is not` en lugar de `==` y `!=`.

CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA

Trabajar con funciones, hace que podamos crear las nuestras propias; con la finalidad de cubrir algún requerimiento que tengamos y no sólo usar las que nos provee el PYTHON.
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=hF85etcCghY, y responde las siguientes preguntas:

- ¿Cuál es la función construida que aparece en el video? y ¿cómo funciona?
- Realiza ese ejemplo en Python