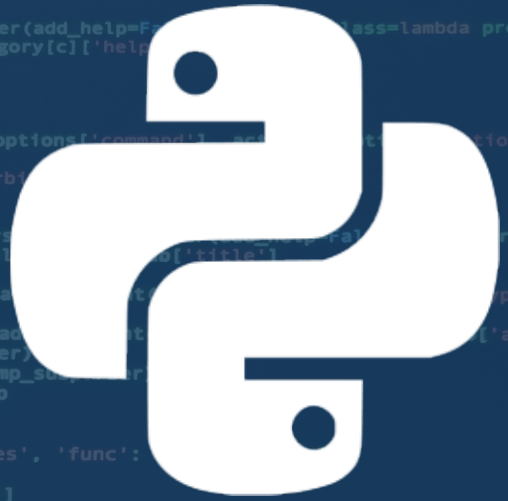

FUNCIONES

PROGRAMACIÓN DE COMPUTADORAS III

Abdel G. Martínez L.

CONCEPTO

- Bloque de código organizado y reutilizable
- Sirve para realizar una tarea específica
- Proveen modularidad a la aplicación
- Python cuenta con sus funciones internas, como `print()`
- Sin embargo, el programador puede crear sus propias funciones

REGLAS DE DEFINICIÓN

- Los bloques de función inician con la palabra reservada `def` seguido por el nombre de la función y paréntesis (`()`)
- Cualquier parámetro de entrada, o argumento, debe ser definidos entre los paréntesis
- El bloque de código de la función inicia luego de los dos puntos (`:`) y debe ser debidamente sangrado
- Debe existir una expresión de retorno, `return`

EJEMPLO

- El siguiente ejemplo toma un texto como argumento e imprime su valor en pantalla, sin retornar nada

```
def imprime(texto):
    "Esta función imprime un texto"
    print texto
    return
```

LLAMANDO UNA FUNCIÓN

- Definir una función solo da un nombre, argumentos y estructura del bloque de código a ejecutar
- El programador puede llamar una función desde otra función o bien globalmente
- Debe indicar el nombre de la función, seguido de paréntesis
- En caso de que tenga argumentos, también debe indicarlos

```
>>> imprime(“Curso”)
```

FUNCIONES ANÓNIMAS

- No son declaradas de la manera estándar
- Se utiliza la sentencia `lambda`
- Contienen una única expresión
- Acceden únicamente a sus variables definidas, no las globales
- No confundir con una función de una línea

EJEMPLO

```
>>> sum = lambda arg1, arg2: arg1 + arg2;
>>> print(”Valor total: ", sum( 10, 20 ))
Valor total:  30
```

CONSIDERACIONES ESPECIALES

- Utilizar las funciones anónimas para tareas específicas que se cumplan en una sola sentencia y sean manejadores de llamados
- Utilizar las funciones tradicionales para tareas generales que serán reutilizables y modulares
- Si defino una variable dentro del bloque de código de la función sería una variable local a la función