

Lógica de programación I

Juan Pablo Restrepo Uribe

Ing. Biomedico

MSc. Automatización y Control Industrial

jprestrepo@correo.iue.edu.co

2023-2

Institución Universitaria de Envigado

Funciones

Una función te permite definir un bloque de código reutilizable que se puede ejecutar muchas veces dentro de tu programa. Las funciones te permiten crear soluciones más modulares.

Si bien Python ya proporciona muchas funciones integradas como `print()` y `len()`, también puedes definir tus propias funciones para usar en tus proyectos.

Una de las grandes ventajas de usar funciones en tu código es que reduce el número total de líneas de código en tu proyecto.

Funciones

La estructura más simple de una función se compone de las siguientes partes:

1. Se utiliza la palabra reservada de Python **def**.
2. Se coloca el **nombre de la función**.
3. Se colocan **paréntesis** y luego se colocan **dos puntos (:)**.
4. Se coloca la **instrucción (o instrucciones)** de la función.

Estructura base simple de una función

```
def nombreFunción():  
    print("Cuerpo de la función")  
    print("Cuerpo de la función")  
  
nombreFunción()
```

Funciones

```
1 def saludo():  
2     print("Bienvenido")  
3     print("A mi programa")  
4  
5 saludo()
```

Funciones con parámetros

```
1 def alerta():  
2     print("Cuidado")  
3     print("Perros bravos en la propiedad")  
4  
5 alerta()  
6 alerta()
```

```
1 def saludoPersonalizado(nombre, apellido):  
2     print("Hola "+nombre+" "+apellido)  
3  
4 saludoPersonalizado()
```

Funciones con parámetros

```
1 def saludoPersonalizado(nombre, apellido):  
2 |     print("Hola "+nombre+" "+apellido)  
3  
4 saludoPersonalizado("Ana", "García")  
5 saludoPersonalizado("Pedro", "Cardenas")
```

```
1 def saludoPersonalizado(nombre, apellido):  
2 |     print("Hola "+nombre+" "+apellido)  
3  
4 saludoPersonalizado("Cardenas", "Pedro")
```

Funciones con parámetros

```
1 def calcularMulta(diasDeRetraso):  
2     if(diasDeRetraso > 7):  
3         return diasDeRetraso*800  
4     else:  
5         return 1000  
6  
7 dias = int(input("Ingrese los días de retraso: "))  
8 deuda = calcularMulta(dias)  
9 print("Debe pagar "+str(deuda)+" pesos de multa")
```

Ejercicios

- Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función EsMultiplo que reciba los dos números, y devuelve si el primero es múltiplo del segundo.
- Crear una función que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que utilizando la función anterior, vaya pidiendo la temperatura máxima y mínima de cada día y vaya mostrando la media. El programa pedirá el número de días que se van a introducir.
- Crea una función “ConvertirEspaciado”, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, “Hola, tú” devolverá “H o l a , t ú “. Crea un programa principal donde se use dicha función.

Ejercicios

- Escribe una función llamada encontrar_palindromos que tome una lista de palabras como argumento y devuelva una lista con las palabras que son palíndromos. Un palíndromo es una palabra que se lee igual hacia adelante y hacia atrás (por ejemplo, "reconocer" o "anilina").

Ejercicios

- El día juliano correspondiente a una fecha es un número entero que indica los días que han transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las siguientes subrutinas:
 - LeerFecha: Nos permite leer por teclado una fecha (día, mes y año).
 - DiasDelMes: Recibe un mes y un año y nos dice los días de ese mes en ese año.
 - EsBisiesto: Recibe un año y nos dice si es bisiesto.
 - Calcular_Dia_Juliano: recibe una fecha y nos devuelve el día juliano.

https://github.com/josedom24/curso_programacion/blob/master/ejercicios/funciones/ejercicio11.psc