universidad cenfotec
La U de la informática

Principios de Programación.

Funciones y procedimientos con listas.

Objetivos

universidad cenfotec
La U de la informática

- Comprender el concepto de parámetros por valor y referencia.
- Aplicar el concepto de parámetros utilizando tipos de datos simples y listas
- Crear algoritmos que incluyan el paso de parámetros por valor y referencia con el lenguaje de programación Python.

Introducción

Imagine que usted debe crear un programa que ejecute diferentes cálculos a partir de listas. El programa debe permitir al usuario seleccionar una de las siguientes opciones:

- Leer una lista e invertir la posición de sus valores en el lista.
- Leer una lista, calcular el promedio de sus valores e imprimir el resultado.
- Leer una lista, calcular el promedio de sus valores e imprimir los valores que sean mayores que el resultado.
- Leer una lista y calcular el valor máximo y mínimo del mismo

¿Cuáles son las acciones que se repiten constantemente en el enunciado anterior?

¿Cómo se puede evitar la reescritura del código y promover su reutilización?

Rutinas con listas

universidad cenfotec
La U de la informática

Las **rutinas** son pequeños bloques de código que corresponden a la solución de pequeños problemas identificados en un problema más grande. Las **funciones** son rutinas que, dentro de sus instrucciones, permiten retornar valores de tipos de dato simple, pero además, permiten retornar listas. Siendo una gran ventaja, ya que es posible retornar un conjunto de valores de un mismo tipo.

Por lo que, para dar solución a los puntos que se plantean en el ejercicio anterior se podría:

- Crear una función que permita leer los valores de una lista y retornar estos valores.
- Crear una función que permita calcular y retornar el promedio de los valores dentro de una lista.

Parámetros actuales y formales

universidad cenfotec
La U de la informática

Es importante recordar que a la hora de trabajar con rutinas se manejan dos conceptos importantes, los cuales son:

- **Parámetros actuales**: son los que se usan en el llamado de una rutina.
- **Parámetros formales**: son los que se especifican a la hora de definir una rutina, y con los cuales se hacen cálculos o se ejecutan instrucciones dentro de la rutina.

```
import math
                              Parámetro formal
def calcular_area_cuadrado(pValor):
  # pValor recibe la copia de numero y se modifica el valor del
parámetro formal
  pValor = math.pow(pValor, 2);
  return pValor;


# Flujo principal
numero = 25
resultado = 0
print(f'Valor inicial del número: {numero}')
# Al hacer el llamado se crea la copia del parámetro actual numero
                                  Parámetro actual
resultado = calcular_area_cuadrado(numero);
# Se imprime el valor de numero después del llamado
print(f'Valor final del número: {numero}')
# Se imprime el resultado final de la operación
print(f'Resultado de la operación: {resultado}')
```

Paso por parámetros

universidad cenfotec
La U de la informática

Cuando se crea la solución, es importante tomar en cuenta las dos formas existentes de **pasar parámetros** a las rutinas. Las cuales son:

Paso por valor

- A la hora de hacer el llamado a una rutina, se **crea una copia** del valor original del **parámetro actual** y se pasa a la rutina al **parámetro formal**.
- La rutina no puede modificar el valor original del parámetro actual, es decir que, si se modifica el valor del parámetro dentro de la rutina, el valor original se mantiene intacto.
- Ejemplo de uso: Tipos de dato simple.

Paso de una referencia

- A la hora de hacer el llamado de una rutina, se pasa una **referencia del valor original** del **parámetro actual**.
- La rutina modifica el parámetro actual al cambiar los valores del parámetro formal, es decir que, si se modifica el parámetro formal dentro de la rutina, se está modificando el parámetro actual también.
- Ejemplo de uso: listas.

Paso de parámetros por valor

universidad cenfotec_
La U de la informática

Dentro de la memoria principal de la computadora, al pasar la variable *numero* como parámetro desde el ***flujo principal*** a la rutina ***calcular_area_cuadrado***, se genera una copia del valor en el parámetro *pValor*. Aunque en la rutina se modifique el valor del parámetro formal *pValor*, el parámetro actual *numero* no cambia.

```
import math
def calcular_area_cuadrado(pvalor):
  # pvalor recibe la copia de numero y se modifica el valor del parámetro formal
  pvalor = math.pow(pvalor, 2);
  return pvalor;
# Flujo principal
numero = 25
resultado = 0
print(f'Valor inicial del número: {numero}')
# Al hacer el llamado se crea la copia del parámetro actual número
resultado = calcular_area_cuadrado(numero);
# Se imprime el valor de número después del llamado
print(f'Valor final del número: {numero}')
# Se imprime el resultado final de la operación
print(f'Resultado de la operación: {resultado}')
```

Paso de parámetros por valor

universidad cenfotec
La U de la informática

Al iniciar la ejecución del código anterior, se puede observar que el valor de la variable *numero* se mantiene igual antes y después de ejecutar la rutina ***calcularCuadrado*** incluso cuando el parámetro formal fue modificado. El resultado mostrado en pantalla es el siguiente:

```
Valor inicial del número: 25.0
Valor final del número: 25.0
Resultado de la operación: 625.0
```

Paso de parámetros de una referencia (parte 1)

universidad cenfotec_
La U de la informática

Al pasar el lista *números* como parámetro desde el ***flujo principal*** a la rutina ***calcular_area_cuadrado***, se pasa una referencia de *números* al parámetro *pValores*. Cuando se pasa una referencia, toda modificación que se le haga al parámetro formal *pValores* dentro de la rutina, se verá reflejada en el parámetro actual *números*. A continuación, la implementación del ***flujo principal***:

```
numeros = [10, 25, 85, 100]
print("\nValores iniciales del lista: ")

# Se imprimen los valores iniciales del lista
imprimir_lista(numeros);
print("\n\nValores finales del lista: ")

# Al hacer el llamado se pasa la referencia del parámetro actual números
calcular_area_cuadrado(numeros)

# Se imprimen los valores finales del lista
imprimir_lista(numeros)
```

Paso de parámetros de una referencia (parte 2)

universidad cenfotec_
La U de la informática

Dentro de la rutina ***calcular_area_cuadrados*** cada uno de los elementos del lista *pvalores* se eleva al cuadrado, ya que *pvalores* recibe una referencia, toda modificación realizada a *pvalores*, se verá reflejada en el lista que se pase como parámetro actual.

```
import math


def calcular_area_cuadrados(pvalores):
    # pvalores recibe la referencia de números
    # El contenido del parámetro formal y actual se modifican
    i = 0
    tamanno = len(pvalores)
    while (i < tamanno):
        pvalores[i] = math.pow(pvalores[i], 2)
        i += 1
```

Paso de parámetros de una referencia (parte 3)

universidad cenfotec
La U de la informática

La rutina ***imprimirlista*** imprime todos los elementos del lista *pValores*. Esta rutina no modifica el parámetro formal, solo imprime sus valores, a pesar de que recibe también una referencia.

```
def imprimirlista(pValores):
    i = 0
    tamanno = len(pValores)
    while (i < tamanno):
        print(f'{pValores[i]}');
        i += 1
```

Paso de parámetros de una referencia (parte 3)

universidad cenfotec
La U de la informática

Al iniciar la ejecución del código se puede ver que los valores iniciales del lista *numeros* y que estos valores se modifican después de ejecutar la rutina ***calcularCuadrados***. El resultado que se muestra por pantalla es el siguiente:

```
Valores iniciales del lista:
10.0    25.0    85.0    100.0

Valores finales del lista:
100.0   625.0   7225.0  10000.0
```

Procedimiento para llenar una lista de enteros

universidad cenfotec
La U de la informática

A continuación, se muestra el proceso de **análisis, diseño e implementación en código** de un procedimiento donde se pasa la referencia de una lista por parámetros y dentro de la rutina se llenan todas las posiciones del mismo.

Procedimiento para llenar una lista de enteros

universidad cenfotec
La U de la informática

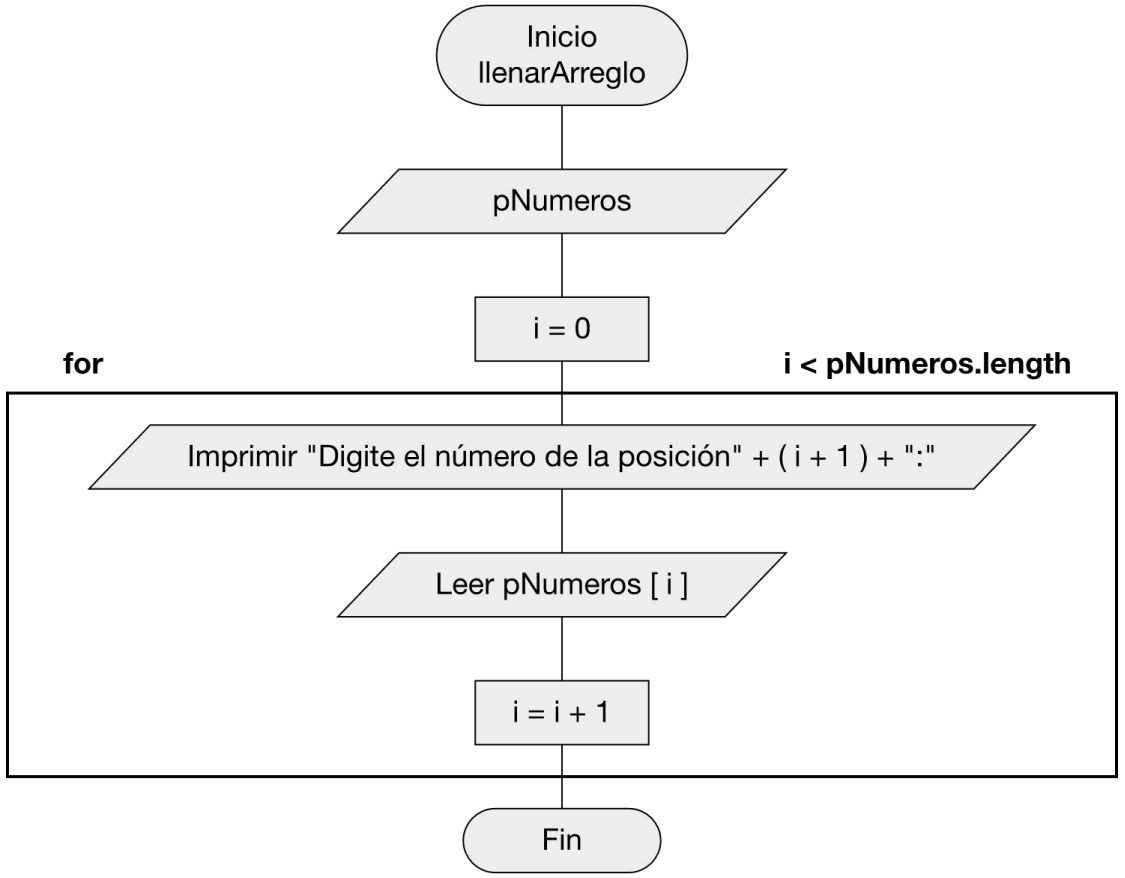
1. Análisis.
2. Diseño o modelado.
3. Implementación.

Entrada / Salida			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
lista de enteros que se recibe por parámetros.	pNumeros	lista	[3, 7, 9]

Procedimiento para llenar una lista de enteros

universidad cenfotec
La U de la informática

1. Análisis.
2. Diseño o modelado.
3. Implementación.



Procedimiento para llenar una lista de enteros

universidad cenfotec
La U de la informática

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Implementación del *flujo principal*

```
numeros = []
llenarlista(numeros) # Llama al procedimiento que lee el lista.
```

Procedimiento para llenar una lista de enteros

universidad cenfotec
La U de la informática

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Implementación de *llenarlista*

```
def llenarlista (pNumeros):
 i = 0
 tamanolista  = int(input(("Digite el tamaño del lista: ")))
 while (i < tamanolista):
     numero = int(input(f"Digite el número de la posición {i + 1} : "))
     pNumeros.append(numero)
     i += 1
```

Dado que el lista *pNumeros* dentro de la rutina ***llenarlista***, recibe una referencia al parámetro actual, no es necesario retornar los cambios realizados en las casillas del lista.

En este caso cuando empieza la ejecución del procedimiento ***llenarlista***, el lista está vacío (sin datos), y cuando termina el lista está lleno (con datos en las casillas).

Función que calcula el promedio del lista

Ahora adicionalmente, se quiere crear una **función** que calcule el promedio de los valores que se agregaron en el lista.

Es importante notar que, a diferencia de la rutina anterior, esta función sí debe retornar un valor y este es el **promedio obtenido**.

Función que calcula el promedio del lista

universidad cenfotec
La U de la informática

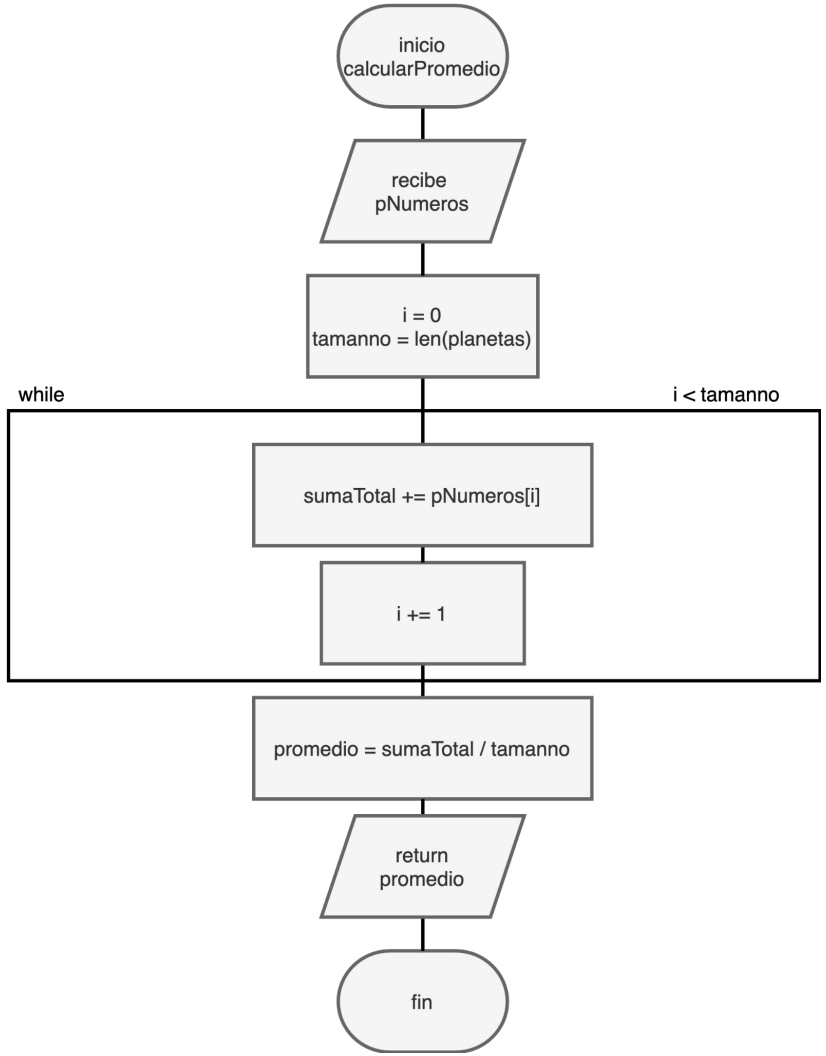
1. Análisis.
2. Diseño o modelado.
3. Implementación.

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de enteros que se recibe por referencia.	pNumeros	lista	[11,15,…20]
Intermedia			
Variable entera que almacena la suma total de los valores en el lista.	sumaTotal	int	322
Salida			
Variable real que almacena el promedio del lista.	promedio	float	64.4

Función que calcula el promedio del lista

universidad cenfotec
La U de la informática

1. Análisis.
2. Diseño o modelado.
3. Implementación.



Función que calcula el promedio del lista

universidad cenfotec
La U de la informática

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Implementación del *flujo principal*

```
numeros = []
llenarlista(numeros) # Llama al procedimiento que lee el lista.
promedio = calcularPromedio(numeros); # Se calcula el promedio
print(f"El promedio del lista es: {promedio}")
```

Función que calcula el promedio del lista

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Implementación del *calcularPromedio*

```
def calcularPromedio(pNumeros):
    i = 0
    promedio = 0
    sumaTotal = 0
    tamanno = len(pNumeros)
    while (i < tamanno):
        sumaTotal += pNumeros[i];
        i += 1
    promedio = sumaTotal / tamanno
    return promedio;
```

Función que calcula el promedio del lista

Al iniciar la ejecución del código, se observa cómo se solicita por teclado el tamaño y los valores del lista. Posteriormente, se ejecuta la función ***calcularPromedio*** que recibe como parámetro la referencia del lista *numeros*, esta realiza el cálculo y retorna el promedio al ***flujo principal***. Finalmente, en el ***flujo principal*** se imprime el resultado del promedio en pantalla.

```
Digite el tamaño del lista: 5
Digite el número de la posición 1: 56
Digite el número de la posición 2: 25
Digite el número de la posición 3: 74
Digite el número de la posición 4: 86
Digite el número de la posición 5: 100

El promedio del lista es: 68.2
```


universidad
cenfotec_
La U de la informática