



Trabajo Práctico N°4

Funciones

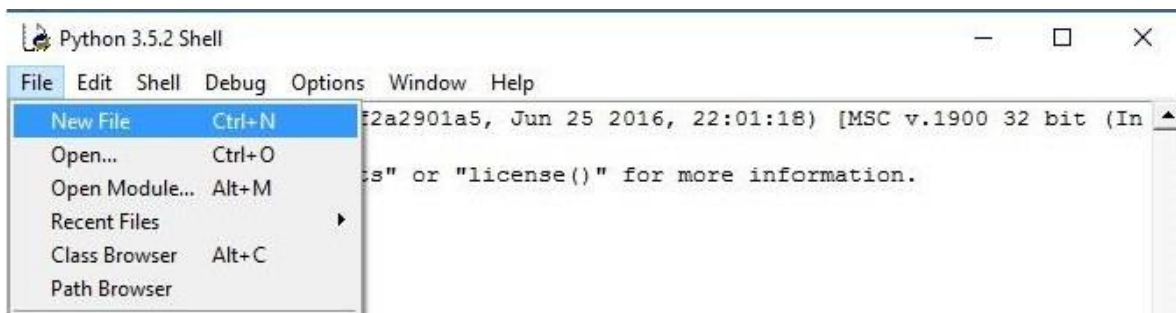
Ejercicio 1 – Hola

1. El objetivo del programa es mostrar por pantalla un saludo mediante una función.
2. Cree un archivo, llamado Hola.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.
5. Solicite al usuario que ingrese otro número y guárdelo en una variable de tipo entera llamada numero2 mediante la conversión de tipos.
6. Agregue la función comparar, dentro del mismo archivo, pero al comienzo del mismo mediante la siguiente línea:

```
def hola():  
    print("hola")  
    return
```
7. Complete el cuerpo de la función comparar para que realice su tarea correctamente.
8. Muestre el resultado en la salida estándar mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 2 – Comparar

1. El objetivo del programa es determinar cuál es mayor de 2 números ingresados por el usuario.
2. Abrir el programa IDLE (Python 3.5)
3. Observar que aparece el prompt esperando ingreso de datos.
4. Ingresar al menú **File** → **New File** o **Ctrl+N**



5. Se abrirá una nueva pantalla en blanco





6. Cree un archivo, llamado Comparar.
7. Solicite al usuario que ingrese un número y guárdelo en una variable llamada numero1.
8. Solicite al usuario que ingrese otro número y guárdelo en otra variable llamada numero2.
9. Agregue la función comparar mediante la palabra reservada def dentro del mismo archivo, pero al comienzo del mismo mediante la siguiente línea:
def comparar(numero1, numero2):
...
return

Observe que dentro de los paréntesis figuran los parámetros enviados a la función para proseguir con su comparación.

10. Complete el cuerpo de la función comparar para que realice su tarea correctamente mediante operadores de comparación (if...else...).
11. Muestre el resultado en la salida estándar mediante un mensaje.
12. Guarde el código del programa con **File → Save** o **Ctrl+S** con el nombre Comparar.py (recuerde que el programa debe tener extensión .py o Tipo "Python Files", de lo contrario no funcionará).
No se olvide de seleccionar la carpeta donde desee que se guarde el archivo (**Recomendación:** Modificar la que aparece por defecto).
13. Ejecute el programa con **F5** o con **Run → Run Module**

```
def comparar(n
if numero1
    print(
else:
    print("el segundo numero es mas grande")
return

numero1=int(input("ingrese un numero: "))
numero2=int(input("ingrese otro numero: "))
comparar(numero1, numero2)
```

14. Se abrirá nuevamente la pantalla principal del IDLE (donde figura el prompt >>>).
15. A continuación comenzará a ejecutarse su programa y se visualizará el resultado del mismo de color azul.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/Python/resoluciones TP4/comparar.py =====
ingrese un numero: 3
ingrese otro numero: 4
el segundo numero es mas grande
>>> |
```

Ejercicio 3 – Calculadora

1. El objetivo del programa es realizar una mini calculadora que realice las 4 operaciones básicas: suma, resta, multiplicación y división.



2. Cree un archivo llamado Calculadora.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.
5. Solicite al usuario que ingrese otro número y guárdelo en una variable de tipo entera llamada numero2 mediante la conversión de tipos.
6. Muestre al usuario un menú con las opciones disponibles para realizar operaciones:
 - 1) Sumar
 - 2) Restar
 - 3) Multiplicar
 - 4) Dividir.Recuerde que con “\n” se puede realizar un salto de línea.
7. Solicite al usuario que ingrese otro número para la opción deseada y guárdelo en una variable de tipo entera llamada opción mediante la conversión de tipos.
8. Agregue las funciones correspondientes a cada operación mediante la palabra reservada def
9. Complete el cuerpo de cada función para que realice su tarea correctamente.
10. Muestre el resultado de la operación seleccionada mediante un mensaje.
11. Guarde el archivo.
12. Ejecute.

Ejercicio 4 – Palabra más larga

1. El objetivo del programa es determinar cuál de dos palabras es la más larga mediante funciones.
2. Cree un archivo llamado Palabras.
3. Solicite al usuario que ingrese una palabra y guárdela en una variable de tipo string llamada palabra1.
4. Solicite al usuario que ingrese otra palabra y guárdela en una variable de tipo string llamada palabra2.
5. Declare la función compararLongitud mediante la palabra reservada def
6. Dentro del cuerpo de la función, determine la longitud de las mismas utilizando la función len().
7. Compare dichas longitudes e indique cual es mayor mediante un mensaje.
8. Muestre el resultado.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 5 – Par o Impar

1. El objetivo del programa es determinar si un número ingresado por teclado es par o impar mediante funciones.
2. Cree un archivo llamado ParImpar.
3. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero.
4. Declare la función EsPar mediante la palabra reservada def
5. Dentro del cuerpo de la función, determine si el número es par mediante condicionales.
6. Muestre el resultado.
7. Guarde el archivo.
8. Ejecute.



Ejercicio 6 – Positivo o Negativo

1. El objetivo del programa es determinar si un número ingresado por teclado es positivo o negativo mediante funciones.
2. Cree un archivo llamado PositivoNegativo.
3. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero.
4. Declare la función Positivo mediante la palabra reservada def
5. Dentro del cuerpo de la función, determine si el número es positivo mediante condicionales.
6. Muestre el resultado.
7. Guarde el archivo.
8. Ejecute.

Ejercicio 7 – Divisor

1. El objetivo del programa es determinar si un número ingresado por teclado es divisor de un 2do número mediante funciones.
2. Cree un archivo llamado Divisor.
3. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero1.
4. Solicite al usuario que ingrese otro número y guárdelo en una variable de tipo entera llamada numero2.
5. Declare la función divisor mediante la palabra reservada def
6. Dentro del cuerpo de la función, determine si un número es divisor de otro mediante condicionales.
7. Muestre el resultado.
8. Guarde el archivo.
9. Ejecute.

Ejercicio 8 – Lista de divisores

1. El objetivo del programa es determinar qué números dentro de una lista son divisores de otro número y formar con ellos una nueva lista.
2. Cree un nuevo archivo llamado ListaDivisores
3. Cree una variable de tipo arreglo llamada lista con contenido vacío.
4. Cree una variable de tipo arreglo llamada listaDivisores con contenido vacío.
5. Solicite al usuario que ingrese la cantidad de elementos que desea en el arreglo.
6. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada dividendo.
7. Solicite al usuario ingresar un número y guárdelo el nombre en una variable entera llamada numero.
8. Utilice un bucle para recorrer el arreglo con la cantidad de elementos ingresada.
9. Indique al programa que vaya colocando los elementos en el arreglo con la instrucción append.
10. Agregue la función correspondiente al cálculo de los divisores mediante la palabra reservada def.
Recuerde enviarle por parámetros a la función la lista con los elementos dentro, el número ingresado por teclado y la lista vacía donde deberá colocar los divisores.
11. Complete el cuerpo de la función para que realice su tarea correctamente.
12. Agregue a la 2da lista los elementos que sean divisores del número ingresado.
13. Muestre el resultado de la operación mediante un mensaje.



14. Guarde el archivo.
15. Ejecute.
- 16.

Recursión

Ejercicio 9 – Factorial

1. El objetivo del programa es determinar el factorial de un número mediante la llamada recursiva de funciones.
2. Cree un archivo llamado Factorial.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.
5. Agregue la función correspondiente al cálculo del factorial mediante la palabra reservada def
6. Complete el cuerpo de la función para que realice su tarea correctamente.
7. Realice la llamada recursiva a la función mediante la siguiente línea:
`numero * factorial(numero-1)`
8. Muestre el resultado de la operación mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 10 – Fibonacci

1. El objetivo del programa es determinar el resultado de aplicar la sucesión de fibonacci a un número mediante la llamada recursiva de funciones.
2. Cree un archivo llamado Fibonacci.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.
5. Agregue la función correspondiente al cálculo de Fibonacci mediante la palabra reservada def

Recuerde que:

$$\text{Fibonacci}(n) \begin{cases} 0 & \rightarrow \text{si } n=0 \\ 1 & \rightarrow \text{si } n=1 \\ \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2) & \rightarrow \text{si } n>1 \end{cases}$$

6. Complete el cuerpo de la función para que realice su tarea correctamente.
7. Realice la llamada recursiva a la función.
8. Muestre el resultado de la operación mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 11 – Sumatoria

1. El objetivo del programa es determinar la sumatoria de los n primeros números ingresados por un usuario mediante la llamada recursiva de funciones.
2. Cree un archivo llamado Sumatoria.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.



5. Agregue la función correspondiente al cálculo de la sumatoria mediante la palabra reservada def
6. Complete el cuerpo de la función para que realice su tarea correctamente.
7. Realice la llamada recursiva a la función
8. Muestre el resultado de la operación mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 12 – División

1. El objetivo del programa es determinar la división de 2 números ingresados por teclado mediante restas sucesivas, realizando la llamada recursiva.
2. Cree un archivo llamado Division.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero1 mediante la conversión de tipos.
5. Solicite al usuario que ingrese otro número y guárdelo en una variable de tipo entera llamada numero2 mediante la conversión de tipos.
6. Agregue la función correspondiente al cálculo de la división por restas sucesivas.
7. Complete el cuerpo de la función para que realice su tarea correctamente.
8. Realice la llamada recursiva a la función.
9. Muestre el resultado de la operación mediante un mensaje.
10. Guarde el archivo.
11. Ejecute.

Ejercicio 13 – Invertir

1. El objetivo del programa es un número ingresado por teclado invirtiendo el orden de sus números, realizando la llamada recursiva.
Ej: si ingresa 123 → muestra: 321
2. Cree un archivo llamado Inverso.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable llamada numero.
Nota: Una solución rápida es no realizar la conversión a tipo entero, para realizar su tratamiento como el de una lista.
5. Agregue la función correspondiente a la concatenación de números.
6. Complete el cuerpo de la función para que realice su tarea correctamente.
7. Realice la llamada recursiva a la función.
8. Muestre el resultado de la operación mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 14 – Sumar Dígitos

1. El objetivo del programa es determinar la sumatoria de los dígitos de un número ingresados por teclado, realizando la llamada recursiva.
Ej: si ingresa 123 → muestra: 6
2. Cree un archivo llamado SumarDigitos.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero mediante la conversión de tipos.
5. Agregue la función correspondiente al cálculo de la sumatoria de dígitos.



6. Complete el cuerpo de la función para que realice su tarea correctamente.
7. Realice la llamada recursiva a la función.
8. Muestre el resultado de la operación mediante un mensaje.
9. Guarde el archivo.
10. Ejecute.

Ejercicio 15 – Máximo Común Divisor

1. El objetivo del programa es determinar el máximo común divisor de 2 números ingresados por teclado, realizando la llamada recursiva.
2. Cree un archivo llamado MaximoComunDivisor.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero1 mediante la conversión de tipos.
5. Solicite al usuario que ingrese un número y guárdelo en una variable de tipo entera llamada numero2 mediante la conversión de tipos.
6. Agregue la función correspondiente al cálculo del máximo común divisor.
7. Complete el cuerpo de la función para que realice su tarea correctamente.
8. Realice la llamada recursiva a la función.
9. Muestre el resultado de la operación mediante un mensaje.
10. Guarde el archivo.
11. Ejecute.

Ejercicio 16 – Sumar Elementos

1. El objetivo del programa es determinar la sumatoria de elementos de una lista de números ingresados por teclado, realizando la llamada recursiva.
2. Cree un archivo llamado SumarElementos.
3. Dentro del archivo cree la sentencia para leer desde el teclado.
4. Solicite al usuario que ingrese la cantidad de elementos de la lista y guárdelo en una variable de tipo entera llamada cantidad mediante la conversión de tipos.
5. Cree una variable de tipo arreglo llamada lista con la longitud ingresada por el usuario.
6. Utilice un bucle para recorrer el arreglo.
7. Solicite al usuario ingresar un número y guárdelo en la lista.
8. Indique al programa que vaya colocando los elementos en el arreglo con la instrucción append vista en Trabajo Práctico N°3.
9. Agregue la función correspondiente al cálculo de la sumatoria de elementos.
10. Complete el cuerpo de la función para que realice su tarea correctamente.
11. Realice la llamada recursiva a la función.
12. Muestre el resultado de la operación mediante un mensaje.
13. Guarde el archivo.
14. Ejecute.