

## Trabajo práctico nro. 1

	<b>Asignatura: Programación I</b>	
	<b>Cursado:</b> Primer Trimestre	<b>Horas semanales:</b>
		<b>Horas semestrales:</b> <i>Cantidad estimada de horas semestrales/anuales.</i>
	<b>Carrera:</b> <i>Tecnicatura Universitaria en Programación</i>	<b>Nivel (Año):</b>  <div style="display: flex; align-items: center; justify-content: center;"> <span style="color: green; font-size: 1.5em; margin-right: 5px;">✓</span> <span>1°</span> <span style="margin: 0 10px;"><input type="checkbox"/></span> <span>2°</span> <span style="margin: 0 10px;"><input type="checkbox"/></span> <span>3°</span> </div>
<b>Ciclo Lectivo:</b> 2023		

### Integrantes de la Cátedra:

#### - DOCENTES:

Nombre del Profesor	Periodo	Cantidad horas materia
Cinthia Rigoni	2023	6 horas

1. Indica si los siguientes identificadores son válidos en Python. En el caso de que el identificador no sea válido, explica el motivo.

a) alumno1	Si	m) 5var	No, inicia con n°
b) 1alumno	No, inicia con número	n) with	No, palabra reservada
c) primerNombre	No, mala práctica	o) Auto-seleccionado	No, empieza con Mayusc.
d) /apellido	No, tiene símbolo	p) %aumento	No, inicia con símbolo
e) tamaño_máximo	No, caracteres especiales	q) _123	Si
f) for	No, palabra reservada	r) ValorTotal	No, mala práctica
g) _\$nombre	No, tiene símbolo	s) DESCUENTO	Si
h) global	No, palabra reservada	t) año	No, caracter especial
i) primer_nombre	Si	u) mes_actual	Si
j) num_mayor	Si	v) apellido&nombre	No, tiene símbolo
k) menor-num	No, mala práctica	w) 89GW5	No, inicia con n°
l) dni@alumno	Si	x) valido?	No, tiene símbolo

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

Debemos tener en cuenta que no pueden usarse palabras reservadas ni caracteres especiales en la declaración de variables.

2. Indica qué dato se guarda en la variable **x** en cada caso, suponiendo una ejecución secuencial del programa.

a)  $x=46$   
 $x=15$   
 $x=30$

d)  $x=10-2$   
 $10+2$

b)  $x=46$   
 $x=15$   
 $x=30$

e)  $y=3*(4+2)$   
 $x=y+2$   
 $z=5$   
 $x=y-z$

c)  $x=25$   
 $x+10$   
.

f)  $x=3$   
 $y=x+6$   
 $x=y-1$

→ Solución:

- a)30
- b)30
- c)35
- d)8
- e)13
- f)8

3. Indica qué tipo de dato se guarda en cada variable.

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

- |                            |                           |
|----------------------------|---------------------------|
| a) var1 = 100/5            | i) var9 = int("748")      |
| b) var2 = 7/2              | j) var10 = float("832")   |
| c) var3 = 7//2             | k) var11 = float(321)     |
| d) var4 = 7%2              | l) var12 = str(65)        |
| e) var5 = 'a'              | m) var13 = 1+5!=3         |
| f) var6 = "casa"+"s"       | n) var14 = 177%2==0       |
| g) var7 = "automóvil"[1+1] | o) var15 = len("ola")<=12 |
| h) var8 = len("carpeta")   |                           |

→ Solución:

- |                                 |                                  |
|---------------------------------|----------------------------------|
| a) Se trata del tipo "float".   | i) Se trata del tipo "integer".  |
| b) Se trata del tipo "float".   | j) Se trata del tipo "float".    |
| c) Se trata del tipo "integer". | k) Se trata del tipo "float".    |
| d) Se trata del tipo "integer". | l) Se trata del tipo "string".   |
| e) Se trata del tipo "string".  | m) Se trata del tipo "Booleano". |
| f) Se trata del tipo "string".  | n) Se trata del tipo "Booleano". |
| g) Se trata del tipo "string".  | o) Se trata del tipo "Booleano". |
| h) Se trata del tipo "integer". |                                  |

4. Indica cuáles de las siguientes operaciones no son válidas.

- |                        |                      |
|------------------------|----------------------|
| a) 11-(4%2+10)         | g) int("4")          |
| b) "30"+"2"            | h) int(4)            |
| c) "30"+2              | i) int("z")          |
| d) "hola"[len("hola")] | j) <u>int</u> ("4.") |
| e) len(456)            | k) 4<"f"             |
| f) "hola"[len("fin")]  | l) "palabra"="rama"  |

→ Solución:

a) es válida

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

- b) es válida.
- c) NO es válida.
- d) NO es válida.
- e) NO es válido.
- f) es válido.
- g) es válido.
- h) es válido
- i) NO es válido.
- j) NO es válido.
- k) NO es válido.
- l) NO es válido.

5. Declara una variable de cada tipo de dato y asígnale un valor.

- |           |               |
|-----------|---------------|
| • int     | • list        |
| • float   | • tuple       |
| • complex | • dict        |
| • string  | • <u>null</u> |
| • bool    |               |

Sabías que en Python al momento de declarar una variable es necesario darle un valor. Por lo que para que esta sea 'vacía', podemos declarar de la siguiente forma:

```
var_nula = None
```

→ Solución:

- int: num=10
- float: num1=0.35
- var\_complex: 4+5j
- string: persona1= "Maria"
- bool: x = true    y = false

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

- list: lista\_alum = [1,"maria",2,"lucho"]
- var\_tuple: (1,"hola",5.4)
- dict: menu={"empanada" : 200, "pastel" : 500, "ensalada" : 300}
- null: varN= None

6. Teniendo la variable de tipo **string**: frase = "Caminante, no hay camino, se hace camino al andar.", indica qué obtendríamos si aplicáramos:

- a) frase[5]
- b) frase[-1]
- c) frase[0:8]
- d) frase[:3]

→ Solución:

- a)Obtenemos "a".
- b)Obtenemos ".".
- c)Obtenemos "Caminant".
- d)Obtenemos "Cin,oaci,ea molnr".

7. Usando la variable del ejercicio anterior:

- a) ¿Cómo obtenemos la cadena al revés? ".radna la onimac ecah es ,onimac yah on ,etnanimac"
- b) ¿Cómo obtenemos la subcadena 'hace'?

→ Solución

- a) Lo obtenemos con frase[::-1]
- b) Lo obtenemos con frase[29:33]

8. Métodos upper(), lower() y title().

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

El método **title()** cambia la primera letra de cada palabra a mayúscula.

- a) Pon en mayúsculas la primera letra de cada palabra del siguiente nombre: 'lucas mauricio barros'.
- b) Deja esta frase totalmente en letras minúsculas: 'El qUe No arRiesGa, nO gANa.'
- c) Deja esta frase totalmente en letras mayúsculas: 'El qUe No arRiesGa, nO gANa.'

→ Solución:

- a) Lucas Mauricio Barros.

En código:

```
frase = ("lucas mauricio barros")  
frase=frase.title()  
print(frase)
```

- b) el que no arriesga, no gana.

En código:

```
frase = ("El qUe No arRiesGa, nO gANa.")  
frase=frase.lower()  
print(frase)
```

- c) EL QUE NO ARRIESGA NO GANA.

En código:

```
frase = ("El qUe No arRiesGa, nO gANa.")  
frase=frase.upper()  
print(frase)
```

- 9. Convierte en expresiones algorítmicas las siguientes expresiones algebraicas. Coloca paréntesis solamente donde sean necesarios.

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

a)  $\frac{b}{2} - 4ac$

g)  $a^2 + b^2$

b)  $3xy - 5x + 12x - 17$

h)  $(a + b)^2$

c)  $\frac{b+d}{c+4}$

i)  $\sqrt[3]{b} + 34$

d)  $\frac{xy}{y} + 2$

j)  $\frac{x}{y}(z + w)\pi$

e)  $\frac{1}{y} + \frac{3x}{z} + 1$

k)  $\frac{x+y}{u+\frac{w}{b}}$

f)  $\frac{1}{y+3} + \frac{x}{y} + 1$

→ Solución:

a)  $b/2-4*a*c$

b)  $3*x*y-5*x+12*x-17$

c)  $(b+d)/(c+4)$

d)  $x*y/y+2$

e)  $(1/y)+(3*x/2)+1$

f)  $1/(y+3)+x/y+1$

g)  $a**2+b**2$

h)  $(a+b)**2$

i)  $b**(1/3)+34$

j)  $x/z*(z+w)*\pi$

k)  $(x+y)/(u+(w/b))$

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

10. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.

a)  $x = (-b + (b^2 - 4ac)^{(1/2)}) / (2a)$

b)  $(x^2 + y^2) / (z^2)$

c)  $4x^2 - 2x + 7$

d)  $(b^2)^{(1/2)} - 4ac$

e)  $(a-b)^2 + (c-d)^3$

f)  $(x+y)/y - (3x)/5$

g)  $(a^2 + b^2)^{(1/3)} = c$

h)  $3x^2 / (3x^3 / (4y+6))^{(1/2)}$

→ Solución:

a)  $x = -b + ((b^2) - 4ac)^{(1/2)} / (2a)$

b)  $x = (x^2 + y^2) / z^2$

c)  $4x^2 - 2x + 7$

d)  $(b^2)^{(1/2)} - 4.a.c$

e)  $(a-b)^2 + (c-d)^3$

f)  $(x+y)/y - (3.x)/5$

g)  $(a^2 + b^2)^{(1/3)} = 3$



Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

h)  $3 \cdot (x^2) / (3 \cdot (x^3) / (4 \cdot y) + 6)^{(1/2)}$

11. Dada la siguiente expresión aritmética:

$$a + b * \left(5 - \frac{c}{2}\right) + (7 - x) / (y + 4)$$

Determinar qué resultado obtendremos si a=5, b=2, c=6, x=(-6) y y=4.

→ Solución:

El resultado de la expresión es 10.62

12. Escribe las expresiones algorítmicas equivalentes a los siguientes enunciados:

- a) Suma los números 5 y 3.
- b) Calcula el promedio de los números 4, 7 y 9.
- c) Calcula el área de un rectángulo con base 8 y altura 5.
- d) Verifica si un número es par.
- e) El doble de 16.
- f) Seis veces la diferencia de 8 y 3.
- g) La diferencia entre el producto de 2 por 6 y la suma de 4 y 3.
- h) Comprobar si un número entero N es múltiplo de 2 y de 3.
- i) Comprobar si el contenido de la variable precio es igual o mayor que 15 y menor que 90.
- j) Modificar el valor de la variable entera N incrementándolo en 12.
- k) Modificar el valor de la variable entera N disminuyéndolo en 5.
- l) Modificar el valor de la variable entera N triplicando su valor.
- m) Modificar el valor de la variable entera N por su mitad.

→ Solución:

- a) suma=5+3
- b) promedio=((4+7+9)/3)
- c) area: 8\*5

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

- d)  $\text{num} \% 2 = 0$
- e)  $16 * 2$
- f)  $p = 6 * (8 - 3)$
- g)  $n = (2 * 6) - (4 + 3)$
- h)  $(N \% 2 = 0) \text{ AND } (N \% 3 = 0)$
- i)  $\text{precio} \geq 15 \text{ AND } \text{precio} < 90$
- j)  $n = n + 12$
- k)  $n = n - 5$
- l)  $n = n * 3$
- m)  $n = n / 2$

13. ¿Qué resultado (True/False) dan las siguientes operaciones?

- a) not true
- b) not(1+2 != 3)
- c)  $x = (\text{len}('jugar') > 5) \text{ and } (\text{len}('jugar') < 10)$
- d) 'alto'[2] == 't' and x
- e)  $842913 \% 10 != 3 \text{ and } \text{len}('café') == 3$
- f)  $0 != 0 \text{ or } 'a' < 'y'$
- g) True or int('50') >= 50
- h) edad = 20  
 $\text{not}(x) \text{ or } \text{edad} \% 2 == 0$
- i) es\_cliente = False  
 $\text{not}(\text{es\_cliente} \text{ and } \text{not}(\text{edad} < 18))$

Solución

- a) False
- b) True
- c) False

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

- d) False
- e) False
- f) True
- g) True
- h) True
- i) True

14. Siendo x una variable de tipo entera, con valor 5, determine qué se mostrará por pantalla en cada caso.

- a) `print(x += 1)`
- b) `print(x -= 2)`
- c) `print(x *= 5)`
- d) `print(x /= 5)`

Sabías que en Python no existen los operadores de incremento y decremento, por lo que si queremos aumentar en 1 una variable usamos el operador `+=1`. En caso de querer disminuir en 1 una variable usamos el operador `-=1`. Este operador es válido para sumar o restar cualquier valor.

Podemos aplicar esta misma lógica con la multiplicación y la división usando los operadores `*=` y `/=` respectivamente.

→ Solución:

- a) Se imprime "6".
- b) Se imprime "3".
- c) Se imprime "25".
- d) Se imprime "1.0"

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: `tupla = ('texto', 10, 15.6, 'texto')`

#### 15. Tipos *list*, *tuple* y *dict*.

Una **lista** es una variable con múltiples valores. Pueden contener cualquier tipo de dato soportado por Python en cualquier orden.

Por ejemplo: `lista = ['texto', 10, 15.6, 'texto']`

```
colores = ["rojo", "azul", "verde", "amarillo", "marrón", "lila", "negro", "rosa"]
```

- De la siguiente lista, ¿qué color está en la posición 3?, ¿cómo accedemos a esta posición?
- ¿En qué posición se encuentra el color 'rojo'? ¿Y el 'rosa'?
- Crea una lista que contenga los siguientes valores en las posiciones indicadas.

- 'uno' en la posición 4.
- 'dos' en la posición 1.
- 'tres' en la posición 0.
- 'cuatro' en la posición 3.
- 'cinco' en la posición 2.

- Imprime la segunda posición de esta tupla.

```
colores = ('rojo', 'azul', 'verde', 'amarillo', 'marrón', 'lila', 'negro', 'rosa', 'blanco', 'naranja')
```

- Utiliza los símbolos de suma y resta para obtener el resultado 25 a partir de los elementos de la siguiente tupla en una variable llamada operación.

```
numeros = (10, 1, 5, 11)
```

Un **diccionario** en Python es una estructura de datos que permite almacenar cualquier tipo de información. Los valores de un diccionario se guardan utilizando un par de valores que siempre van enlazados. Una es la denominada como Key o *Clave*, que es la que nos permite encontrar un dato dentro del diccionario. Cada clave está acompañada por el dato o *valor* al que representa.

Veamos un ejemplo para comprender mejor: `diccionario = {'nombre':'Antonio', 'apellido':'López', 'edad':35, 'peso':72.6}`

- Cuenta la cantidad de elementos del siguiente diccionario.

```
diccionario = {"a": 1, "b": 2, "c": 3, "d": 4}
```

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

g) Accede al valor de la clave 'c' en el diccionario.

→ **Solución:**

- a) "amarillo". Accedemos: colores=[3]
- b) "Rojo" en la posición 0. Y "rosa" en la posición 7.
- c) números=["tres","dos","cinco","cuatro","uno"]
- d) print(colores[1]) > azul
- e) operacion =( numeros[0]-numeros[1]+numeros[2]+numeros[3])
- f) Hay 4 elementos.
- g) diccionario=["c"]

16. Vamos a practicar el uso de las funciones **input()** y **print()**.

Ejemplo: Solicita el nombre de una persona e imprime un mensaje de bienvenida.

```
nombre = input("Ingresa tu nombre: ")
print("¡Hola", nombre + "! Bienvenido(a).")
```

- a) Solicita dos números al usuario, súmalos e imprime el resultado.
- b) Solicita la edad de una persona, calcula cuántos años faltan para que cumpla 100 años e imprime el resultado.

→ **Solución:**

a)

```
num1=int(input("Ingresa el primer numero:"))
num2=int(input("Ingresa el segundo numero:"))
print(num1+num2)
```

b)

```
num2=int(input("Ingresa su edad:"))
años_p=100-num2
print(f"Para que usted cumpla 100 años faltan {años_p}")
```

Las **tuplas** son como las listas pero con dos diferencias clave. La primera diferencia es que las tuplas se escriben con paréntesis () y las listas con corchetes []. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: **tupla = ('texto', 10, 15.6, 'texto')**

## 17. Operadores ternarios.

Los operadores ternarios son más conocidos en Python como expresiones condicionales. Estos operadores evalúan si una expresión es verdadera o no.

Estructura: `condition_if_true if condition else condition_if_false`

Un ejemplo:

```
es_bonito = True
estado = "Es bonito" if es_bonito else "No es bonito"
```

*¡Practiquemos!* Crear las variables necesarias para realizar la ejercitación.

- a) Comprobar si un número es par o impar.
- b) Obtener el valor absoluto de un número.
- c) Comparar dos números y obtener el mayor.

→ **Solución:**

a)

```
num1= int(input("Ingrese un numero para comprobar si es par o impar"))
resultado="Es par" if num1%2==0 else"Es impar"
print(resultado)
```

b)

```
num1=float(input("Ingrese un número para obtener su valor absoluto: "))
print(abs(num1))
```

c)

```
num1= int(input("Ingrese el primer número"))
num2= int(input("Ingrese el segundo número"))
print("El numero mayor de los dos números es: ")
resultado= num1 if num1>num2 else num2
print(resultado)
```