

Trabajo práctico nro. 1

	Asignatura: Programación I	
	Cursado: Primer Trimestre	Horas semanales:
		Horas semestrales: <i>Cantidad estimada de horas semestrales/anuales.</i>
	Carrera: <i>Tecnicatura Universitaria en Programación</i>	Nivel (Año): <input type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> 3°
Ciclo Lectivo: 2023		

Integrantes de la Cátedra:

- DOCENTES:

Nombre del Profesor	Periodo	Cantidad horas materia
		6 horas

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

- | | |
|--|--|
| a) alumno1 (válido) | m) 5var (inválido/xq empieza con N°) |
| b) 1alumno (inválido/xq empieza con N°) | n) with (inválido/palabra reservada) |
| c) primerNombre(válido/no buena práctica) | o) Auto-seleccionado(inválido/se debe usar _) |
| d) /apellido (inválido/xq usa carácter especial) | p) %aumento (inválido/usa carácter especial) |
| e) tamaño_máximo (válido) | q) _123 (válido) |
| f) for (inválido/palabra reservada) | r) ValorTotal (válido/no buena práctica) |
| g) _\$nombre (válido) | s) DESCUENTO (válido/no buena práctica) |
| h) global (inválido/palabra reservada) | t) año (válido) |
| i) primer_nombre (válido) | u) mes_actual (válido) |
| j) num_mayor (válido) | v) apellido&nombre (inválido/usa caracteres esp) |
| k) menor-num (inválido/debe usar _) | w) 89W5 (inválido/xq empieza con N°) |
| l) dni@alumno (inválido/usa caracteres esp.) | x) valido? (inválido/usa caracteres especiales) |

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

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

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

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

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

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

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

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

a) salida 30
b) salida 30
c) salida 25

d) salida 8
e) salida 13
f) salida 8

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

a) $\text{var1} = 100/5$

b) $\text{var2} = 7/2$

c) $\text{var3} = 7//2$

d) $\text{var4} = 7\%2$

e) $\text{var5} = 'a'$

f) $\text{var6} = "casa"+"s"$

g) $\text{var7} = "automóvil"[1+1]$

h) $\text{var8} = \text{len}("carpeta")$

i) $\text{var9} = \text{int}("748")$

j) $\text{var10} = \text{float}("832")$

k) $\text{var11} = \text{float}(321)$

l) $\text{var12} = \text{str}(65)$

m) $\text{var13} = 1+5!=3$

n) $\text{var14} = 177\%2==0$

o) $\text{var15} = \text{len}("ola")\leq 12$

a) float
b) float
c) int
d) int
e) str
f) str
g) str
h) int

i) int
j) float
k) float
l) str
m) bool
n) bool
o) bool

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

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

- | | |
|-------------|-------------|
| a) Válido | g) válido |
| b) válido | h) válido |
| c) inválido | i) inválido |
| d) inválido | j) inválido |
| e) inválido | k) inválido |
| f) válido | l) inválido |

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

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

```
1 num_int = 34
2 num_float = 7.7
3 var_complex = 5 + 5j
4 var_string = "hola"
5 var_boolean = 7 % 2 == 0
6 ejem_list = [1, 2, 3, 4, 5]
7 ejem_tuple = (6, 7, 8, 9)
8 ejem_dict = {"Nombre": "Sara", "Edad": 27, "Documento": 1003882}
9 var_null = None
```

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
```

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] Se obtiene (a)
- b) frase[-1] Se obtiene (.)
- c) frase[0:8] Se obtiene (Caminant)
- d) frase[::3] Se obtiene (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"

Se obtiene si usamos esta expresión -> print(frase[::-1])

- b) ¿Cómo obtenemos la subcadena 'hace'?

Una manera sería:

```
frase = "Caminante, no hay camino, se hace camino al andar."
```

```
# Buscamos la posición de inicio de la palabra "hace"
```

```
inicio = frase.find("hace")
```

```
# Verificamos si la palabra fue encontrada
```

```
if inicio != -1:
```

```
    # La palabra "hace" comienza en la posición 'inicio'
```

```
    subcadena = frase[inicio:inicio + len("hace")]
```

```
    print(subcadena)
```

```
else:
```

```
    print("La palabra 'hace' no se encontró en la frase.")
```

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

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.'

```
1 name = "lucas mauricio barros"
2 print(name.title())
3 frase = " El qUe No arRiesGa, nO gANa."
4 print(frase.lower())
5 print(frase.upper())
6
```

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

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

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

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

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

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

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

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

d) $((x*y)/y) + 2$

j) $(x/y)*(z+w)*3.14$

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

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

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

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

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

b) $(x^{**2} + y^{**2}) / (z^{**2})$

c) $4 * x^{**2} - 2 * x + 7$

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

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

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

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

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

a) $x = \sqrt{-b + (b^2 - 4ac)}$

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

2a

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

d) $\sqrt{(b^2)^2 - 4ac}$

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

f) $\frac{x + y}{y - 3x/5}$

g.) $\sqrt[3]{(a^2 + b^2)} = c$

h) $\frac{3x^2}{\sqrt{\frac{3x^3}{4y + 6}}}$

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 $y=4$.

Resultado 10625

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.

- | | |
|-----------------------|--|
| a) $5+3$ | g) $(2*6)-(4+3)$ |
| b) $(4+7+9)/3$ | h) $N\%2==0 \ \&\& \ N\%3==0;$ |
| c) $A=8*5$ | i) $\text{precio} \geq 15 \ \&\& \ \text{precio} < 90$ |
| d) $\text{num}\%2==0$ | j) $n +=12$ |
| e) $16*2$ | k) $n -=5$ |
| f) $(8-3)*6$ | l) $n *=3$ |
| g) $(2*6)-(4+3)$ | m) $n /= 2$ |

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

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

- | | |
|----------|---------|
| a) False | f) True |
| b) True | g) True |
| c) False | h) True |

- d) False
- e) False

- i) True
- f) 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)`

- a) Salida 6
- b) Salida 3
- c) Salida 25
- d) Salida 1.0

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.

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']`

- a) De la siguiente lista, ¿qué color está en la posición 3?, ¿cómo accedemos a esta posición?

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

Respuesta: El color es el amarillo. Para acceder escribimos `colores[3]`

- b) ¿En qué posición se encuentra el color 'rojo'? ¿Y el 'rosa'?

Respuesta: El rojo está en la posición 0 y el rosa en la posición 7

- c) 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.


```
Numeros = ["tres","dos","cinco","cuatro","uno"]
```

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) Imprime la segunda posición de esta tupla.

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

```
print(colores[1])
```

e) 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 operacion.

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

```
1 numeros = (10,1,5,11)
2 operacion = numeros[0]+numeros[2]+numeros[3]-numeros[1]
3 print(operacion)
```

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}**

f) Cuenta la cantidad de elementos del siguiente diccionario.

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

```
print(len(diccionario))
```

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

```
print(diccionario.get('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.

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

b) Solicita la edad de una persona, calcula cuántos años faltan para que cumpla 100 años e imprime el resultado.

```
edad = int(input("Ingresa su edad: "))
resultado = 100 - edad
print(resultado)
```

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.

```
num = int(input("ingrese numero: "))
resultado = "par" if num % 2 == 0 else "impar"
print(f"El número {num} es {resultado}.")
```

b) Obtener el valor absoluto de un número.

```
num = int(input("ingrese numero: "))
valor_absoluto = num if num >= 0 else -num
print("El valor absoluto de", num, "es", valor_absoluto)
```

c) Comparar dos números y obtener el mayor.

```
num1 = int(input("ingrese numero: "))
num2 = int(input("ingrese numero: "))
mayor = num1 if num1 > num2 else num2
```

```
print("El número mayor es:", mayor)
```