

## 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>  <input type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> 3°
<b>Ciclo Lectivo:</b> 2023		

### Integrantes de la Cátedra:

#### - DOCENTES:

Nombre del Profesor	Periodo	Cantidad horas materia
		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<br>b) 1alumno<br>c) primerNombre<br>d) /apellido<br>e) tamaño_máximo<br>f) for<br>g) _\$nombre<br>h) global<br>i) primer_nombre<br>j) num_mayor<br>k) menor-num<br>l) dni@alumno | m) 5var<br>n) with<br>o) Auto-seleccionado<br>p) %aumento<br>q) _123<br>r) ValorTotal<br>s) DESCUENTO<br>t) año<br>u) mes_actual<br>v) apellido&nombre<br>w) 89GW5<br>x) valido? |
|---|--|

- a) Identificador valido  
 b) Identificador invalido porque empieza con un N°  
 c) Identificador valido  
 d) Identificador invalido porque contiene /  
 e) Identificador valido

- f) Identificador invalido porque es una palabra reservada
- g) Identificador invalido porque contiene \$
- h) Identificador invalido porque es una palabra reservada
- i) Identificador valido
- j) Identificador valido
- k) Identificador invalido porque contiene -
- l) Identificador invalido porque contiene @
- m) Identificador invalido porque empieza con un N°
- n) Identificador invalido porque es una palabra reservada
- o) Identificador invalido porque contiene -
- p) Identificador invalido porque contiene %
- q) Identificador valido
- r) Identificador valido, pero se considera mala practica
- s) Identificador valido, pero se considera mala practica
- t) Identificador valido, pero se considera mala practica
- u) Identificador valido
- v) Identificador invalido porque contiene &
- w) Identificador invalido porque empieza con un N°
- x) Identificador invalido porque contiene ?
  - Si contiene "ñ" está mal
  - Debe trabajarse: nomb\_apell
  - No nombApell

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

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)  $X=30$
- b)  $X=30$

- c)  $X=25$
- d)  $X=8$

- e)  $X=13$
- f)  $X=8$

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

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

- Integers: c, d, h, i
- Float: a, b, j, k
- String: e, f, g, l
- Boolean: m, n, o

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

- |                                     |                                  |
|-------------------------------------|----------------------------------|
| a) <code>11-(4%2+10)</code>         | 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&lt;"f"</code>         |
| f) <code>"hola"[len("fin")]</code>  | l) <code>"palabra"="rama"</code> |

Las operaciones invalidas son: c, d, e, i, j, k, l

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
```

- |                       |  |
|-----------------------|--|
| • var_int = 8         | • var_list = ["lala1", "lala2"]              |
| • var_float = 2.5     | • var_tupla = ("lala1", "lala2")             |
| • var_complex = 3+5j  | • var_dict = {"bland": "ford", "tear": 1999} |
| • var_string = "lala" | • var_null = None                            |
| • var_bool = 5==0     |  |

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]   | a) a                 |
| b) frase[-1]  | b) .                 |
| c) frase[0:8] | c) Caminant          |
| d) frase[:3]  | d) 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'?
- a) frase[::-1]
- b) frase[28:33]

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'.

```
text = 'lucas mauricio barros'
print(text.title())
```

- b) Deja esta frase totalmente en letras minúsculas: 'El qUe No arRiesGa, nO gANa.'

```
text = 'El qUe No arRiesGa, nO gANa.'
print(text.lower())
```

- c) Deja esta frase totalmente en letras mayúsculas: 'El qUe No arRiesGa, nO gANa.'

```
text = 'El qUe No arRiesGa, nO gANa.'
print(text.upper())
```

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)  $\text{pow}(a+b,2)$

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

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

d)  $x*y/y+2$

j)  $x/y*(z+w)*\text{math.pi}$

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

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

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

b)  $\frac{x^2 + y^2}{z^2}$

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

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

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

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

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: 10.625

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

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)  $842913\%10 != 3$  and  $\text{len}('café') == 3$
- f)  $0 != 0$  or  $'a' < 'y'$
- g) True or  $\text{int}('50') \geq 50$
- 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 | i) True |
| e) False |         |

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

- |                               |        |
|-------------------------------|--------|
| a) <code>print(x += 1)</code> | a) 6   |
| b) <code>print(x -= 2)</code> | b) 3   |
| c) <code>print(x *= 5)</code> | c) 25  |
| d) <code>print(x /= 5)</code> | d) 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"]
```

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



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) Imprime la segunda posición de esta tupla.

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

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)
```

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}
```

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

- a) amarillo
- colores[3]
- b) colores[0]
- colores[70]
- c) lista=["tres", "dos", "cinco", "cuatro", "uno"]
- d) verde
- e) `n25=int(numeros[0]-numeros[1]+numeros[2]+numeros[3])`
- f) `len(diccionario)`  
Da como resultado 4
- 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.
- 
- a) `n1 = int(input("Ingresa el primer numero: "))`  
`n2 = int(input("Ingresa el segundo numero: "))`  
`print ("La suma da:", n1+n2)`
  - b) `n = int(input("Ingresa su edad: "))`  
`print ("Le faltan",100-n,"años para cumplir 100 años")`

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.
- 
- a) `respuesta = 'Es par' if num%2==0 else 'Es impar'`
  - b) `respuesta = num if num>0 else -num`
  - c) `respuesta = num1 if num1>num2 else num2`