

Trabajo práctico nro. 1

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: Válido. Sigue las convenciones de nombres en Python.

b) 1alumno: No válido. Un identificador no puede comenzar con un número.

c) primerNombre: Válido. Debería ser `primer_nombre` para seguir las convenciones de nombres en Python.

d) /apellido: No válido. Un identificador no puede comenzar con una barra diagonal (/).

e) tamaño_máximo: Válido. Utiliza `snake_case` aunque utilizar "ñ" y acentos no es una práctica común y generalmente no se recomienda.

f) for: No válido. "for" es una palabra reservada en Python y no puede usarse como un identificador.

g) \$nombre: No válido. El signo (\$) no está permitido en los identificadores.

h) global: No válido. "global" es una palabra reservada en Python y no puede usarse como un identificador.

i) primer_nombre: Válido. Sigue las convenciones de nombres en Python.

j) num_mayor: Válido. Sigue las convenciones de nombres en Python.

k) menor-num: No válido. El guion medio (-) no está permitido en los identificadores.

l) dni@alumno: No válido. El signo @ no está permitido en los identificadores.

m) 5var: No válido. Un identificador no puede comenzar con un número.

n) With: No válido. "With" es una palabra reservada en Python y no puede usarse como un identificador.

o) Auto-seleccionado: No válido. El guion medio (-) no está permitido en los identificadores.

p) %aumento: No válido. El signo de porcentaje (%) no está permitido en los identificadores.

q) 123: No válido. Un identificador no puede comenzar con un número.

r) ValorTotal: Válido. Debería ser valor_total para seguir las convenciones de nombres en Python.

s) DESCUENTO: Válido. Es un valor que se considera constante o inmutable en el contexto de un programa.

t) año: Válido. aunque utilizar “ñ” y acentos no es una práctica común y generalmente no se recomienda.

u) mes_actual: Válido. Sigue las convenciones de nombres en Python.

v) apellido&nombre: No válido. El signo "&" no está permitido en los identificadores.

w) 89GW5: No válido. Un identificador no puede comenzar con un número.

x) valido?: No válido. El signo de interrogación (?) no está permitido en los identificadores.

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

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

valor final x = 30

b) $x = 25$
 $x + 10$

valor final x = 25

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

valor final x = 8

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

valor final x = 13

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

valor final x = 8

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

a) var1 = 100/5

Tipo de dato: float (valor decimal)

b) var2 = 7/2

Tipo de dato: float (valor decimal)

c) var3 = 7//2

Tipo de dato: int (valor entero)

d) var4 = 7%2

Tipo de dato: int (valor entero)

e) var5 = 'a'

Tipo de dato: str (cadena de caracteres)

f) var6 = "casa" + "s"

Tipo de dato: str (cadena de caracteres)

g) var7 = "automóvil"[1+1]

Tipo de dato: str (cadena de caracteres)

h) var8 = len("carpeta")

Tipo de dato: int (valor entero)

i) var9 = int("748")

Tipo de dato: int (valor entero)

j) var10 = float("832")

Tipo de dato: float (valor decimal)

k) var11 = float(321)

Tipo de dato: float (valor decimal)

l) var12 = str(65)

Tipo de dato: str (cadena de caracteres)

m) var13 = 1 + 5j = 3 (supongo que se trata de una operación compleja)

Tipo de dato: complex (valor complejo)

n) var14 = 177 % 2 == 0

Tipo de dato: bool (valor booleano, True o False)

o) var15 = len("ola") <= 12

Tipo de dato: bool (valor booleano, True o False)

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

a) 11 - (4 % 2 + 10): Válida. La expresión se evalúa correctamente y da como resultado 1.

b) "30" + "2": Válida. Se realiza una concatenación de cadenas y el resultado es "302".

c) "30" + 2: No válida. No se puede concatenar una cadena con un entero directamente. Debes convertir el entero en una cadena antes de concatenarlo.

d) "hola"[len("hola")]: Válida. Se accede al carácter en la posición de índice igual a la longitud de la cadena "hola", que es 4. Por lo tanto, esto devuelve el último carácter "a".

e) Len(456): No válida. Len se usa para obtener la longitud de secuencias como cadenas, listas, etc., no para números.

f) "hola"[len("fin")]: Válida. Se accede al carácter en la posición de índice igual a la longitud de la cadena "fin", que es 3. Por lo tanto, esto devuelve el último carácter "a".

g) int("4"): Válida. Se convierte la cadena "4" en un entero.

h) int(4): Válida. No es necesario convertir un número entero en otro entero, pero es una operación válida.

i) int("z"): No válida. La cadena "z" no representa un valor numérico válido para convertir a entero.

j) int("4."): No válida. Las cadenas que representan números decimales deben tener un formato adecuado, como "4.0", para ser convertidas correctamente en enteros.

k) 4 < "f": No válida. No puedes comparar un entero directamente con una cadena. Debes convertir la cadena en un valor numérico antes de realizar la comparación.

l) "palabra" = "rama": No válida. El operador de igualdad en Python es "==", no "=". Esta expresión debe ser corregida a "palabra" == "rama" para verificar si son iguales.

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

- **int**
Variable de tipo int (entero)
mi_entero = 42
- **float**
Variable de tipo float (punto flotante)
mi_flotante = 3.14159
- **complex**
Variable de tipo complex (número complejo)
mi_complejo = 2 + 3j
- **string**
Variable de tipo string (cadena de caracteres)
mi_cadena = "Hola, mundo"
- **bool**
Variable de tipo bool (booleano)
mi_booleano = True

- **list**
Variable de tipo list (lista)
`mi_lista = [1, 2, 3, 4, 5]`
- **tuple**
Variable de tipo tuple (tupla)
`mi_tupla = (10, 20, 30)`
- **dict**
Variable de tipo dict (diccionario)
`mi_diccionario = {"nombre": "Juan", "edad": 30}`
- **null**
Variable de tipo None (para representar un valor nulo)
`mi_nulo = None`

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

- `frase[5]`**: Esto obtendría el carácter en la posición de índice 5, que es "n".
- `frase[-1]`**: Esto obtendría el último carácter de la cadena, que es ".".
- `frase[0:8]`**: Esto obtendría una subcadena que abarca desde el índice 0 (incluido) hasta el índice 8 (excluido), lo que daría como resultado "Caminant".
- `frase[::-3]`**: Esto obtendría una subcadena que contiene cada tercer carácter de la cadena original. El resultado sería "Cin,oaci,ea molnr"

7. Usando la variable del ejercicio anterior:

- ¿Cómo obtenemos la cadena al revés? ".radna la onimac ecah es ,onimac yah on ,etnanimac"**



```
1 cadena_original = "Caminante, no hay camino, se hace camino al andar."
2 cadena_al_reves = cadena_original[::-1]
3 #Muestra la cadena invetida ".radna la onimac ecah es ,onimac yah on ,etnanimac"
4 print(cadena_al_reves)
```

b) ¿Cómo obtenemos la subcadena 'hace'?



```
1  cadena_original = "Caminante, no hay camino, se hace camino al andar."
2  # Buscar la posición de "hace" en la cadena
3  inicio = cadena_original.find("hace")
4  # Verificar si se encontró "hace" en la cadena
5  if inicio != -1:
6      # Utilizar la posición encontrada para obtener la subcadena "hace"
7      subcadena = cadena_original[inicio:inicio + 4]
8      print(subcadena) #Muestra la subcadena "hace"
9  else:
10     print("La subcadena 'hace' no se encontró en la cadena original.")
```

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

- a) Pon en mayúsculas la primera letra de cada palabra del siguiente nombre:
'lucas mauricio barros'.



```
1  # a)
2  nombre = 'lucas mauricio barros'
3  nombre_formateado = nombre.title()
4  # Muestra la cadena "Lucas Mauricio Barros"
5  print(nombre_formateado)
```

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



```
1  frase = 'El qUe No arRiesGa, nO gANa.'
2  frase_minusculas = frase.lower()
3  # Muestra la cadena "el que no arriesga, no gana."
4  print(frase_minusculas)
```

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



```
1 frase = 'El qUe No arRiesGa, nO gANa.'
```

```
2 frase_mayusculas = frase.upper()
```

```
3 # Muestra la cadena "EL QUE NO ARRIESGA, NO GANA."
```

```
4 print(frase_mayusculas)
```

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

Expresiones aritméticas convertidas a expresiones algorítmicas:

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

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

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

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

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

i) $b**(1/8)+34$

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

j) $(x/y)*(z+w)*math.pi$

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

Expresiones algorítmicas convertidas a expresiones aritméticas:

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

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

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

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

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

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

g) $\sqrt[3]{a^2 + b^2}$

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.



```
1  # Valores de las variables
2  a = 5
3  b = 2
4  c = 6
5  x = -6
6  y = 4
7  # Evaluación de la expresión
8  resultado = a + b * (5 - c/2) + (7 - x) / (y + 4)
9  print(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.



```
1  import random
2
3  # a) Suma los numeros 5 y 3.
4  resultado_a = 5 + 3
5
6  # b) Calcula el promedio de los numeros 4, 7 y 9.
7  resultado_b = (4 + 7 + 9) / 3
8
9  # c) Calcula el area de un rectangulo con base 8 y altura 5.
10 resultado_c = 8 * 5
11
12 # d) Verifica si un numero es par.
13 numero_d = random.randint(1, 10)
14 es_par_d = numero_d % 2 == 0
15
16 # e) El doble de 16.
17 resultado_e = 16 * 2
18
19 # f) Seis veces la diferencia de 8 y 3.
20 resultado_f = 6 * (8 - 3)
21
22 # g) La diferencia entre el producto de 2 por 6 y la suma de 4 y 3.
23 resultado_g = (2 * 6) - (4 + 3)
24
25 # h) Comprobar si un numero entero N es multiplo de 2 y de 3.
26 N_h = random.randint(1, 100)
27 es_multiplo_de_dos = N_h % 2 == 0
28 es_multiplo_de_tres = N_h % 3 == 0
29
30 # i) Comprobar si el contenido de la variable precio es igual o mayor que 15 y menor que 90.
31 precio = random.uniform(0, 100)
32 cumple_condicion_i = 15 <= precio < 90
33
34 # j) Modificar el valor de la variable entera N incrementandolo en 12.
35 N_j = random.randint(1, 100)
36 N_j += 12
37
38 # k) Modificar el valor de la variable entera N disminuyéndolo en 5.
39 N_k = random.randint(1, 100)
40 N_k -= 5
41
42 # l) Modificar el valor de la variable entera N triplicando su valor.
43 N_l = random.randint(1, 100)
44 N_l *= 3
45
46 # m) Modificar el valor de la variable entera N por su mitad.
47 N_m = random.randint(1, 100)
48 N_m /= 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))

Respuestas:

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

En Python, las operaciones de asignación abreviadas, como '+=', '-=', '*=' y '/=', no puede usarse directamente en una función print(). Por lo tanto, las siguientes operaciones no funcionarán como se espera y arrojarán un error de sintaxis

Para modificar x y luego imprimir su valor, se debería hacer en dos pasos separados:



```
1  x = 5  # Valor inicial de x
2
3  # Modificar el valor de x y luego imprimirlo
4  x += 1
5  print(x) #Resultado 6
6
7  x -= 2
8  print(x) #Resultado 4
9
10 x *= 5
11 print(x) #Resultado 20
12
13 x /= 5
14 print(x) #Resultado 4.0
```

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

- 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"]
```



```
1 colores = ["rojo", "azul", "verde", "amarillo", "marrón", "lila", "negro", "rosa"]
2 color_en_posicion_3 = colores[2] #El índice 2 corresponde a la posición 3
3 print(color_en_posicion_3) #Esto imprimirá "verde"
```

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



```
1 colores = ["rojo", "azul", "verde", "amarillo", "marrón", "lila", "negro", "rosa"]
2
3 # Buscar la posición de 'rojo' y 'rosa' en la lista
4 posicion_rojo = colores.index("rojo")
5 posicion_rosa = colores.index("rosa")
6 print(posicion_rojo) #Imprime 0
7 print(posicion_rosa) #Imprime 7
```

El color 'rojo' se encuentra en la posición 0 de la lista "colores". El color 'rosa' se encuentra en la posición 7. Se puede utilizar `index()` en las listas de Python para encontrar la posición de los elementos 'rojo' y 'rosa' en la lista "colores".

- 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.



```
1 nueva_lista = ["tres", "dos", "cinco", "cuatro", "uno"]
2 # Imprime ['tres', 'dos', 'cinco', 'cuatro', 'uno']
3 print(nueva_lista)
```

d) Imprime la segunda posición de esta tupla.

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



```
1 colores = ('rojo', 'azul', 'verde', 'amarillo', 'marrón', 'lila', 'negro', 'rosa', 'blanco', 'naranja')
2 segunda_posicion = colores[1]
3 print(segunda_posicion) #Imprime azul
```

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 operación.

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



```
1 numeros = (10, 1, 5, 11)
2 operacion = numeros[0] + numeros[3] - numeros[1] - numeros[2]
3 print(operacion) #Imprime 15
```

f) Cuenta la cantidad de elementos del siguiente diccionario.

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



```
1 diccionario = {"a": 1, "b": 2, "c": 3, "d": 4}
2 cantidad_elementos = len(diccionario)
3 print(cantidad_elementos) #Imprime 4
```

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



```
1 diccionario = {"a": 1, "b": 2, "c": 3, "d": 4}
2 valor_c = diccionario["c"]
3 print(valor_c) #Imprime 3
```

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.



```
1 # Solicitar dos números al usuario
2 numero1 = float(input("Ingrese el primer número: "))
3 numero2 = float(input("Ingrese el segundo número: "))
4
5 # Realizar la suma
6 resultado = numero1 + numero2
7
8 # Imprimir el resultado
9 print("El resultado de la suma es:", resultado)
```

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



```
1 # Solicitar la edad de la persona
2 edad = int(input("Ingrese su edad: "))
3
4 # Calcular cuántos años faltan para cumplir 100 años
5 anios_faltantes = 100 - edad
6
7 # Imprimir el resultado
8 print("Faltan", anios_faltantes, "años para cumplir 100 años.")
```

17. Operadores ternarios.

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

a) Comprobar si un número es par o impar.



```
1 import random
2
3 # Generar un número entero aleatorio entre 1 y 100 (ambos inclusive)
4 numero_aleatorio = random.randint(1, 100)
5
6 # Utilizar un operador ternario para determinar si el número es par o impar y construir el mensaje
7 mensaje = f"El número {numero_aleatorio} es par" if numero_aleatorio % 2 == 0 else f"El número {numero_aleatorio} es impar"
8
9 # Imprimir el mensaje
10 print(mensaje)
```

b) Obtener el valor absoluto de un número.



```
1  import random
2
3  # Genera un número entero aleatorio entre -100 y 100
4  numero_aleatorio = random.randint(-100, 100)
5
6  # Utiliza un operador ternario para obtener el valor absoluto
7  # Si el número aleatorio es mayor o igual a 0, se asigna tal cual.
8  # Si el número aleatorio es negativo, se invierte multiplicándolo por -1.
9  valor_absoluto = numero_aleatorio if numero_aleatorio >= 0 else -numero_aleatorio
10
11 # Imprime el valor absoluto del número aleatorio
12 print("Valor absoluto:", valor_absoluto)
```

c) Comparar dos números y obtener el mayor.



```
1  import random
2
3  # Generar dos números enteros aleatorios entre 1 y 100 (ambos inclusive)
4  numero1 = random.randint(1, 100)
5  numero2 = random.randint(1, 100)
6
7  # Determinar el número mayor y asignarlo a numero_mayor
8  numero_mayor = numero1 if numero1 > numero2 else numero2
9
10 # Imprimir el número mayor
11 print(f"El número mayor es {numero_mayor}")
```