**Trabajo práctico nro. 1**

|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Programación I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: |
|  | **Horas semestrales:**  *Cantidad estimada de horas semestrales/anuales.* |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 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.



1. Es valido
2. No es válido, los nombres de las variables no pueden iniciar con un número.
3. Es valido pero no es recomendable. primer\_nombre sería mejor.
4. No es valido, no pueden iniciar con un carácter especial.
5. Es correcto. Pero no recomendable.
6. No se puede tener nombre de palabra reservada.
7. No es válido no pueden iniciar con un carácter especial.
8. No se puede tener nombre de palabra reservada.
9. Correcto.
10. Correcto.
11. No es válido poner el – entre los nombres.
12. No es valido poner el @ entre los dos.
13. No es valido, no se pone un numero antes del nombre.
14. No es valido. No se puede poner un nombre de palabra reservada.
15. No es válido poner el – entre los nombres y tampoco iniciar nombres con mayúsculas.
16. No es valido el % antes del nombre de la variable.
17. No es valido poner el \_ .
18. Es Valido pero no es recomendable.
19. Es Valido pero no recomendable.
20. Es valido pero no recomendable.
21. Es Valido.
22. No es valido el & entre el nombre.
23. No es valido.
24. No es valido el ? al final del nombre.



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



1. 30
2. 30
3. 25
4. 8
5. 13
6. 8
7. Indica qué tipo de dato se guarda en cada variable.



1. Float
2. Float
3. Integers
4. Integers
5. String
6. String
7. String
8. Integers
9. Integers
10. Float
11. Float
12. String
13. Boolean
14. Boolean
15. Boolean
16. Indica cuáles de las siguientes operaciones no son válidas.



Las operaciones que no son validas son: c,e,i,j,k,l.

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



A)Int = numero =42

B) Float = numero\_decimales = 3.1416

C) Complex = numero\_complejo = 3+4j

D) String = cadena = “Hola mundo”

E) Bool = boolean = True

F) List = [10,9,8,7,6]

G) Tuple = tupla = (10,20,30)

H) Dict = diccionario = {“nombre” : “Juan” , “edad”: 30}

I) Null= variable\_nula= None

1. Teniendo la variable de tipo **string:** frase = “Caminante, no hay camino, se hace camino al andar.”, indica qué obtendríamos si aplicáramos:
2. frase[5] = “a”
3. frase[-1] = “.”
4. frase[0:8] = “Caminant”
5. frase[::3] = " Cin, yansheanaaa".
6. Usando la variable del ejercicio anterior:
7. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”

frase[::-1]

1. ¿Cómo obtenemos la subcadena ‘hace’?

posicion\_hace = frase.find('hace')

subcadena\_hace = frase[posicion\_hace:posicion\_hace + len('hace')]

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



1. Pon en mayúsculas la primera letra de cada palabra del siguiente nombre: ‘lucas mauricio barros’.

nombre = 'lucas mauricio barros'

nombre\_en\_mayusculas = nombre.title()

print(nombre\_en\_mayusculas)

1. Deja esta frase totalmente en letras minúsculas: ‘El qUe No arRiesGa, nO gANa.’

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

frase\_en\_minusculas = frase.lower()

print(frase\_en\_minusculas)

1. Deja esta frase totalmente en letras mayúsculas: ‘El qUe No arRiesGa, nO gANa.’

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

frase\_en\_mayusculas = frase.title()

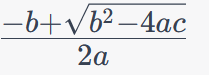
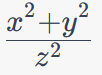
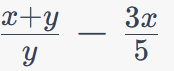
print(frase\_en\_mayusculas)

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



1. (b / 2) - (4 \* A \* C)
2. 3 \* x \* y - 5 \* x + 12 \* x – 17
3. (b + d) / (c + 4)
4. (x \* y) / 2 + 2
5. (1 / y) + (3 \* x / z) + 1
6. (1/y + 3) + (x/y) + 1
7. a \* a + b \* b
8. (a + b) \* (a + b)
9. //
10. (x/y) \* (z+w) \* pi
11. (x + 1/u + w/b)
12. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.



1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. Dada la siguiente expresión aritmética:



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

-El resultado es 85/8.

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



1. 5+3
2. (4+7+9)/3
3. 8\*5
4. N%2==0
5. 2\*16
6. 6\*(8-3)
7. (2\*6)-(4+3)
8. N%2 ==0 Y N%3==0
9. (precio>= 15) ∧ (precio<90)
10. N = N + 12
11. N = N – 5
12. N = N\*3
13. N = N/2
14. ¿Qué resultado (True/False) dan las siguientes operaciones?



1. False
2. True
3. False
4. X is not defined si x estuviera como ‘x’ es True
5. False
6. True
7. True
8. True
9. False
10. Siendo x una variable de tipo entera, con valor 5, determine qué se mostrará por pantalla en cada caso.



1. 6
2. 3
3. 15
4. 1



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



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



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

color\_en\_posicion\_3 = colores[3] // Amarillo

print(color\_en\_posicion\_3)

1. ¿En qué posición se encuentra el color ‘rojo’? ¿Y el ‘rosa’?

Rojo está en la posición 0 .Rosa en la posición 6.

1. Crea una lista que contenga los siguientes valores en las posiciones indicadas.



numeros = ["tres", "dos", "cinco", "cuatro", "uno"]



1. Imprime la segunda posición de esta tupla.



Contando posición 0 la segunda posición es verde:

colores = ("rojo", "azul", "verde", "amarillo", "marron", "lila", "negro", "rosa", "blanco", "naranja")

segunda\_posicion = colores[2]

print(segunda\_posicion)

1. 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.



operacion = numeros[0] + numeros[1] - numeros[2] + numeros[3]



1. Cuenta la cantidad de elementos del siguiente diccionario.



1. Accede al valor de la clave ‘c’ en el diccionario.
2. Vamos a practicar el uso de las funciones **input()** y **print()**.

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



1. Solicita dos números al usuario, súmalos e imprime el resultado.

numero1 = float(input("Ingrese el primer numero:"))

numero2 = float(input("Ingrese el segundo numero:"))

numerosumados= numero1+numero2

print(numerosumados)

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

edad= int(input("Ingrese su edad:"))

falta\_para\_los\_cien = 100-edad

print("Faltan para que cumpla los 100:")

print (falta\_para\_los\_cien)

1. Operadores ternarios.



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

1. Comprobar si un número es par o impar.

numero = int(input("Ingrese un número: "))

if numero % 2 == 0:

print("El número ingresado es par")

else:

print("El número ingresado es impar")

1. Obtener el valor absoluto de un número.

numero = float(input("Ingrese un número: "))

valor\_absoluto = abs(numero)

print("El valor absoluto de", numero, "es:", valor\_absoluto)

1. Comparar dos números y obtener el mayor.

numero1 = float(input("Ingrese el primer número: "))

numero2 = float(input("Ingrese el segundo número: "))

if numero1 > numero2:

mayor = numero1

else:

mayor = numero2

print("El número mayor entre", numero1, "y", numero2, "es:", mayor)