**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 válido.
2. No es válido, un identificador no puede comenzar por un número.
3. Es válido, ya que no dará error, pero es buena práctica utilizar todo minúsculas y las palabras separadas por un guión bajo.
4. No es válido, no se pueden utilizar caracteres especiales.
5. No es válido, no puede contener “ñ” ni acentos.
6. No es válido, es una palabra reservada.
7. No es válido, no se pueden utilizar caracteres especiales.
8. No es válido, es una palabra reservada.
9. Es válido.
10. Es válido.
11. No es válido, las palabras deben separarse con guión bajo, no guión medio.
12. No es válido, no se puede utilizar carácter especial @.
13. No es válido, no puede comenzar con un número.
14. No es válido, es una palabra reservada.
15. No es válido, contiene guión medio como separador.
16. No es válido, contiene carácter especial.
17. Es válido, pero se recomienda colocar un nombre descriptivo para una fácil lectura.
18. Es válido, ya que no dará error, pero no es recomendable utilizar mayúsculas.
19. Es válido, las constantes se escriben en mayúsculas.
20. No es válido, contiene “ñ”.
21. Es válido.
22. No es válido, contiene carácter especial.
23. No es válido, no puede comenzar con un número.
24. No es válido, no puede contener carácter especial “?”.



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



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



1. Tipo numérico float
2. Tipo numérico float
3. Tipo numérico int
4. Tipo numérico int
5. Tipo texto
6. Tipo texto
7. Tipo texto
8. Tipo numérico int
9. Tipo numérico int
10. Tipo numérico float
11. Tipo numérico float
12. Tipo texto
13. Tipo booleano
14. Tipo booleano
15. Tipo booleano
16. Indica cuáles de las siguientes operaciones no son válidas.



1. Si es válida
2. Si es válida
3. No es válida
4. No es válida
5. No es válida
6. Si es válida
7. Si es válida
8. Si es válida
9. No es válida
10. No es válida
11. No es válida

l) No es válida

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



1. var1= 12
2. var2= 35.1
3. var3= 3+4j
4. var4= “hola”
5. var\_bool=True
6. var6=[s, f, h ]
7. var7=(r, t, y)
8. var8= {'a': 1, 'b': 3, 'z': 8}
9. var9= None
10. Teniendo la variable de tipo **string:** frase = “Caminante, no hay camino, se hace camino al andar.”, indica qué obtendríamos si aplicáramos:
11. frase[5]
12. frase[-1]
13. frase[0:8]
14. frase[::3]
15. Se obtiene “a”
16. Se obtiene “.”
17. Se obtiene “Caminant”
18. Se obtiene ” Cin,oaci,ea molnr”
19. Usando la variable del ejercicio anterior:
20. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”

Se logra colocando frase[::-1]

1. ¿Cómo obtenemos la subcadena ‘hace’?

Se puede obtener así, frase[29:-17]

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



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

asd5= "lucas mauricio barros’"

asd6=asd5.title()

print(asd6)

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

asd3= "El qUe No arRiesGa, nO gANa"

asd4=asd3.lower()

print(asd4)

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

asd= "El qUe No arRiesGa, nO gANa"

asd2=asd.upper()

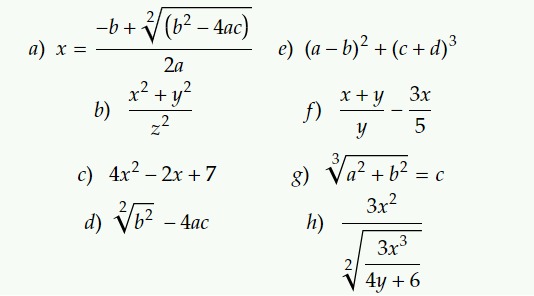
print(asd2)

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)/y+2
5. 1/y+(3\*x)/z+1
6. 1/(y+3)+x/y+1
7. a\*\*2+b\*\*2
8. (a+b)\*\*2
9. b\*\*(1/3)+34
10. (x/y)(z+w) π
11. (x+y)/(u+w/b)
12. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.





1. Dada la siguiente expresión aritmética:



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

Se obtendría 10.625

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



1. 5+3
2. p=(4+7+9)/3
3. a=8\*5
4. x%2==0
5. doble=2\*16
6. dif=6\*(8-3)
7. 2\*6-(4+3)
8. N%2==0 & N%3==0
9. 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. False
5. False
6. True
7. True
8. True
9. True
10. Siendo x una variable de tipo entera, con valor 5, determine qué se mostrará por pantalla en cada caso.



1. Mostrará 6
2. Mostrará 3
3. Mostrará 25
4. Mostrará 1



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



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



El color “amarillo”. Se puede acceder con: colores=[3]

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

El color rojo se encuentra en la posición 0 y el rosa en la 7.

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



lista2=[‘tres’, ’dos’, ’cinco’, ’cuatro’, ’uno’]



1. Imprime la segunda posición de esta tupla.



print(colores[2])

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.



La cantidad de elementos es 4

1. Accede al valor de la clave ‘c’ en el diccionario.

diccionario[“c”]

1. 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.

num0 = int(input("Ingrese el primer numero: "))

num1 = int(input("Ingrese el segundo numero: "))

sum = num0 + num1

print("La suma es: ", sum)

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: "))

rest = 100 - edad

print("Falatan ", rest, " años para que cumpla 100.")

1. Operadores ternarios.



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

1. Comprobar si un número es par o impar.

'Es par' if num % 2 == 0 else 'Es impar'

1. Obtener el valor absoluto de un número.

num if num>0 else –num

1. Comparar dos números y obtener el mayor.

num1 if num1>num2 else num2