**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-

1. Es válido y es una buena práctica
2. No es válido porque no debe iniciar con un número.
3. Es válido, pero es una mala práctica, ya que no se debe usar el método camelCase
4. No es válido, no puede tener simbología en el nombre
5. No es válido porque no se puede usar la letra “Ñ” ni tampoco acentos
6. No es válido porque es una palabra reservada como comando
7. No es válido porque no se puede usar simbología
8. No es válido por que es una palabra reservada como comando
9. Es válido y es una buena practica
10. Es válido y es una buena practica
11. No es válido porque no se pueden usar guiones altos “-”
12. Es valido
13. No es válido porque no puede empezar con un número adelante
14. No es válido por que es una palabra reservada como comando
15. No es válido por que es una mala práctica arrancar con mayúscula y no se puede usar guion alto.
16. No es válido porque no se puede usar simbología, ya que es el comando MOD
17. Es valido
18. Valido, mala practica
19. Valido únicamente para variables especiales.
20. No es válido porque no se puede usar la letra “Ñ”
21. Es valido
22. No es válido la porque la letra “&” es un comando
23. No es válido porque no puede comenzar con un número
24. No es válido porque no se puede usar simbología



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



a) x= 30 b) x= 30 c) x= 25 d) x= 8 e) x= 13 f) x= 8

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



1. Float
2. Con decimal
3. Entera
4. Entera
5. Carácter (string)
6. String
7. String
8. Entero
9. Entero
10. Con decimal
11. Con decimal
12. Carácter (string)
13. Booleano
14. Booleano (Falso)
15. Booleano (Verdadero)
16. Indica cuáles de las siguientes operaciones no son válidas.



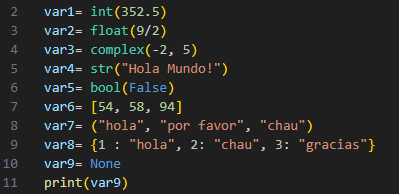
4-

1. Valida
2. Es valida, pero retorna sumando como string
3. Invalida, suma string con numero
4. Es invalida
5. Invalida, para el comando “len” se debe usar un string
6. Valida
7. Valida
8. Valida
9. Invalida
10. Invalida
11. Invalida
12. Invalida
13. Declara una variable de cada tipo de dato y asígnale un valor.





5 -



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]
3. frase[-1]
4. frase[0:8]
5. frase[::3]

6-

1. Se imprime la letra “a” de Caminante, ya que estoy imprimiendo la letra en la sexta posición de la variable.
2. Se imprime “.” que sería la última posición de la variable.
3. Se imprime “Caminant”, ya que imprime de la posición 0 a la 8 de la variable.
4. Se imprime “Cin,oaci,ea molnr”, ya que imprime lo que hay cada 3 posiciones de la variable, ósea la posición 0; 3; 6; 9; etc.
5. Usando la variable del ejercicio anterior:
6. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”
7. ¿Cómo obtenemos la subcadena ‘hace’?

7-

1. Para obtener la cadena al revés se debe utilizar el código - print (frase[::-1])
2. Para obtener la subcadena “hace” se debe utilizar el código - print (frase[29:33])
3. Métodos upper(), lower() y title().



1. Pon en mayúsculas la primera letra de cada palabra del siguiente nombre: ‘lucas mauricio barros’.
2. Deja esta frase totalmente en letras minúsculas: ‘El qUe No arRiesGa, nO gANa.’
3. Deja esta frase totalmente en letras mayúsculas: ‘El qUe No arRiesGa, nO gANa.’

8-

a)



b)



c)



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



9-

1. (b/2) – (4\*a\*c)
2. (3\*x\*y) – (5\*x) + (12\*x) – 17
3. (b+c) / (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\*\*3) + 34
10. (x/y) \* (z+w) \* math.pi
11. (x+y) / (u+(w/b))
12. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.



10 –

1. Dada la siguiente expresión aritmética:



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

11 -

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



12 –

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



13 –

1. False
2. True
3. False
4. False
5. False
6. False
7. True
8. False
9. True
10. Siendo x una variable de tipo entera, con valor 5, determine qué se mostrará por pantalla en cada caso.





14-

1. Imprime = 6
2. Imprime = 3
3. Imprime = 25
4. Imprime = 1
5. Tipos *list*, *tuple* y *dict*.



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



1. ¿En qué posición se encuentra el color ‘rojo’? ¿Y el ‘rosa’?
2. Crea una lista que contenga los siguientes valores en las posiciones indicadas.





1. Imprime la segunda posición de esta tupla.



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.





1. Cuenta la cantidad de elementos del siguiente diccionario.



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

15 –

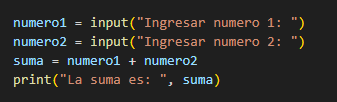
1. El color en la posición 3 es el “amarillo”, y se accede con el código= color[3]
2. Se encuentran, el rojo en la posición 0 y el rosa en la posición 7
3. lista[“tres”, “dos”, ”cinco”, “cuatro”, “uno”]
4. colores[2]
5. operación = números[0] – números[1] + números[2] + números[3]
6. Ese diccionario tiene 4 elementos
7. Diccionario[“c”]
8. Vamos a practicar el uso de las funciones **input()** y **print()**.

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

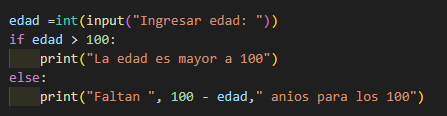


1. Solicita dos números al usuario, súmalos e imprime el resultado.
2. Solicita la edad de una persona, calcula cuántos años faltan para que cumpla 100 años e imprime el resultado.

16-







1. Operadores ternarios.



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

1. Comprobar si un número es par o impar.
2. Obtener el valor absoluto de un número.
3. Comparar dos números y obtener el mayor.

17-

1. espar = "Es par" if num % 2 == 0 else "Es impar"
2. absoluto = num if num>0 else - num
3. mayor = num1 if num1>num2 else num2

Trabajo Practico N°1 – Programación 1

Marcos Di Meco