**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.



- Identificadores validos:

1. alumno1
2. primerNombre
3. \_$nombre
4. primer\_nombre
5. num\_mayor
6. \_123
7. ValorTotal
8. DESCUENTO
9. mes\_actual
10. Identificadores no validos:

a) 1alumno: Los identificadores no pueden comenzar con un dígito.

b) /apellido: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos

c) tamaño\_máximo: Los identificadores no pueden contener caracteres especiales como la tilde

d) for: “for” es una palabra reservada en Python y no puede ser utilizada como un identificado.

e) global: “global” es una palabra reservada en Python y no puede ser utilizada como un identificador

f) menor-num: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos.

g) dni@alumno: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos.  
h) 5var: Los identificadores no pueden comenzar con un dígito.  
i) with: “with” es una palabra reservada en Python y no puede ser utilizada como un identificador  
 j) auto-seleccionado: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos.

k) %aumento: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos.   
l) año: Los identificadores no pueden contener caracteres especiales como la tilde.  
m) apellido&nombre: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos  
n) 89GW5: Los identificadores no pueden comenzar con un dígito1.   
ñ) valido?: Los identificadores solo pueden contener letras (mayúsculas o minúsculas), dígitos y guiones bajos.



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



a) 30  
b) 30  
c) 25  
d) 8  
e) 15  
f) 8

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



a) Entero

b) Entero

c) Entero

d) Entero

f) String

g) String

h) String

i) Entero

j) Entero

k) error(no se puede convertir un string en un flotante),

l) float

m) string

n) boolean,

ñ) boolean,

o) boolean

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



No son validas las siguientes: c, d,e,f,i,j,k,l

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



mi\_entero = 5 # int

mi\_flotante = 3.14 # float

mi\_complejo = 2 + 3j # complex

mi\_cadena = "Hola" # string

mi\_booleano = True # bool

mi\_lista = [1, 2, 3] # list

mi\_tupla = (4, 5, 6) # tuple

mi\_diccionario = {"clave": "valor"} # dict

mi\_nulo = None # null

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]

a) frase[5] devuelve el caracter en la posicion 5 de la cadena, que es 'a'

b) frase[-1] devuelve el ultimo caracter de la cadena, que es '.'.

c) frase[0:8] devuelve una subcadena que va desde la posicion 0 hasta la posición 7 (inclusive) de la cadena, por lo que da "caminant".

d) frase[::3] devuelve una subcadena que tiene cada tercer caracter de la cadena original, empezando desde el principio. Da esto: "Cnenn a aa,haa.aaod.".

1. Usando la variable del ejercicio anterior:
2. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”
3. ¿Cómo obtenemos la subcadena ‘hace’?

para obtener la cadena al revez haría: frase\_al\_reves = frase[::-1]

Para obtener ‘hace’ haría esto: subcadena\_hace = frase[30:34]

1. 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.
4. a) nombre = 'lucas mauricio barros'
5. nombre\_modificado = nombre.title()
6. print(nombre\_modificado)
7. b) frase\_mayusculas = 'El qUe No arRiesGa, nO gANa.'
8. frase\_minusculas = frase\_mayusculas.lower()
9. print(frase\_minusculas)  
     
   c) frase\_minusculas = 'El qUe No arRiesGa, nO gANa.'
10. frase\_mayusculas = frase\_minusculas.upper()
11. print(frase\_mayusculas)
12. 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/3)+1
6. 1/(y+3) + (x/y)+1
7. (a\*a)+(b\*b)
8. (a+b)\*(a+b)
9. (b\*b\*b)+34
10. x/y\*(z+w)\*3.14
11. (x+y)/(u+(u/b)
12. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.

›

A screenshot of a math problem

Description automatically generated

1. Dada la siguiente expresión aritmética:



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

a + b\*(5 - c/2) + (7 - x) / (y + 4)

5 + 2\*(5 - 6/2) + (7 - (-6)) / (4 + 4)

5 + 2\*(5 - 3) + (7 + 6) / (4 + 4)

5 + 2\*2 + 13 / 8

5 + 4 + 13 / 8

9 + 13 / 8

9 + 1.625 = 10.625

Rta: a=5, b=2, c=6, x=(-6) y y=4 es 10.625.

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



1. X=5+3
2. Promedio=(4+7+9)/3
3. Área= 8x5
4. esPar= (numero%2==0)
5. doble=16x2
6. resultado=6x(8-3)
7. diferencia=(2x6)-(4+3)
8. esmultiplo=(n%2==0)&&(n%3==0)
9. esvalido=(precio>=15)&&(precio<90)
10. N=n+12
11. N=n-5
12. N=3xn
13. N=n/2
14. ¿Qué resultado (True/False) dan las siguientes operaciones?



1. False
2. False
3. True
4. False
5. False
6. False
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.



X=5

X+=1 = 6

X-=2 = 2  
X\*=5= 20  
X/=5= 4.0



1. 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.

A) Amarillo

B) Rojo= 0 , Rosa=7

C) Numeros= [“tres”,”dos”,”cinco”,””cuatro”,”uno”] D) print(colores[1])

E)operacion = numeros[0] + numeros[3] - numeros[1] - numeros[2] F) cantidad\_elementos = len(diccionario)

G)valor\_c = 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.
2. Solicita la edad de una persona, calcula cuántos años faltan para que cumpla 100 años e imprime el resultado.

a) # Solicita al usuario que ingrese el primer número

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

# Solicita al usuario que ingrese el segundo número

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

# Realiza la suma de los dos números

suma = numero1 + numero2

# Imprime el resultado

print(f"La suma de {numero1} y {numero2} es igual a: {suma}")

b)  
# Solicita al usuario que ingrese su edad

edad = int(input("Ingresa tu edad actual: "))

# Calcula cuántos años faltan para que la persona cumpla 100 años

anios\_faltantes = 100 - edad

# Imprime el resultado

print(f"Faltan {anios\_faltantes} años para que cumplas 100 años.")

1. Operadores ternarios.



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

1. Comprobar si un número es par o impar.  
   numero = int(input("Ingresa un número: "))

resultado = "par" if numero % 2 == 0 else "impar"

print(f"El número {numero} es {resultado}.")

1. Obtener el valor absoluto de un número.  
   numero = float(input("Ingresa un número: "))

valor\_absoluto = numero if numero >= 0 else -numero

print(f"El valor absoluto de {numero} es {valor\_absoluto}.")

1. Comparar dos números y obtener el mayor.

mayor = numero1 if numero1 > numero2 else numero2

print(f"El número mayor entre {numero1} y {numero2} es {mayor}.")