**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.



Respuesta 1

a\_ valido

b\_ invalido, porque empieza con un número.

c\_ valido, pero mala práctica al no separar las palabras con guion bajo y escribir una mayúscula.

d\_ invalido, porque utiliza es un carácter reservado .

e\_ invalido, utiliza ñ

f\_ Invalido, palabra reservada

g\_ Invalido, utiliza un carácter especial

h\_ Invalido, palabra reservada

i\_ valido

j\_ valido

k\_ invalido, utiliza un carácter reservado

l\_ valido

m\_ invalido, empieza con un numero

n\_ invalido, palabra reservada

o\_ invalido, empieza en mayúscula y utiliza carácter reservado

p\_ invalido, por utilizar una operación matemática

q\_ valido

r\_ Válido, mala práctica

s\_ Válido, solo para variables especiales

t\_ invalido, utiliza ñ

u\_ valido

v\_ invalido, utiliza un operador lógico

W\_ invalido, empieza con número y tiene mayúscula

X\_ invalido, utiliza un carácter especial.



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\_ 13

f\_ 8

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



a\_ Su tipo de dato es decimal

b\_ Su tipo de dato es decimal

c\_ Su tipo de dato es entero

d\_ Su tipo de dato es entero

e\_ Su tipo de dato es string

f\_ Su tipo de dato es string

g\_ Su tipo de dato es string

h\_ Su tipo de dato es entero

i\_ Su tipo de dato es entero

j\_ Su tipo de dato es decimal

k\_ Su tipo de dato es decimal

l\_ Su tipo de dato es string

m\_ Su tipo de dato es booleano

n\_ Su tipo de dato es booleano

o\_ Su tipo de dato es booleano

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



a\_ valido

b\_ valido

c\_ invalido

d\_ invalido

e\_ invalido

f\_ valido

g\_ valido

h\_ valido

i\_ invalido

j\_ invalido

k\_ invalido

l\_ invalido

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



* Int: var1 = 3
* Float: var2 = 3.32
* Complex: var3 = 3+2i
* String: var4 = “soldado”
* Bool: var5 = VERDADERO
* List: var6 = [“gato”, “perro”, “loro”]
* Tuple: var7 = (10, “horno”, 2)
* Dic: var8= {“plato”: “ensalada”, “ingrediente”: “lechuga”}
* Null: var\_nulo = 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]
3. frase[-1]
4. frase[0:8]
5. frase[::3]

a\_ “a”

b\_ “.”

c\_ “Caminante”

d\_ “Cin,oaci,ea molnr”

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’?

a\_ frase[::-1]

b\_ frase[29:32]

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.’

a\_ ‘lucas mauricio barros’.title()

b\_ ‘El qUe No arRiesGa, nO gANa.’.lower()

c\_ ‘El qUe No arRiesGa, nO gANa.’.upper()

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



a\_ b/2-4\*a\*c

b\_ 3\*x\*y-5\*x+12\*x-17

c\_ (b+d)/(c+4)

d\_ x\*y/y + 2

e\_ 1/y + (3\*x)/z +1

f\_ 1/(y+3) + x/y + 1

g\_ a\*\*2 + b\*\*2

h\_ (a+b)\*\*2

i\_ b\*\*(1/3)+34

j\_x/y\*(z+w)\*math.pi

k\_ (x+y)/(u+w/b)

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



a\_ x = (-b+)

b\_

c\_

d\_ 4ac

e\_ (a-b)²+(c-d)³

f\_

g\_

h\_

1. Dada la siguiente expresión aritmética:



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

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

5+2\*(5-3)+(7+6)/8

5+2\*2+13/8

5+4+13/8 = 10,625

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



a\_ suma = 5+3

b\_ promedio = (4+7+9)/3

c\_ area = 8\*5

d\_

num = input(“ingrese un número: ”)

if num%2==0 :

print(“Es par”)

else :

print(“No es par)

e\_ doble = 16\*2

f\_ var1 = (8-3)\*6

g\_ var2 = 2\*3-(4+3)

h\_

num = int (input(“ingrese un número: ”))

if num%2==0 y num%3==0 :

print(“Es múltiplo de 2 y 3”)

else :

print(“No es multiplo de 2 y 3”)

i­\_

precio = flow( input(“asigne un precio: ”))

if precio>=15 y precio<90 :

print(“El precio si es mayor o igual que 15 y menor a 90”)

else :

print(“El precio no cumple con ser mayor o igual que 15 y menor a 90”)

j\_ N +=12

k\_ N -= 5

l\_ N \*= 3

m\_ N /= 2

1. ¿Qué resultado (True/False) dan las siguientes operaciones?

Respuesta:

a\_ false

b\_ true

c\_ false

d\_ false

e\_ false

f\_ true

g\_ true

h\_ true

i\_ true



1. Siendo x una variable de tipo entera, con valor 5, determine qué se mostrará por pantalla en cada caso.



a\_ 6

b\_ 3

c\_ 25

d\_ 1



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



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



En la posición 3 está el color amarillo, accedemos a esta posición escribiendo: colores[3]

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

El rojo se encuentra en la posición 0, mientra que el rosa se encuentra en la posición 7.

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



lista\_num = [‘tres’, ‘dos’, ‘cinco’, ‘cuatro’, ‘uno’]



1. Imprime la segunda posición de esta tupla.



print(colores[1])

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[2] + numeros[3] – numeros[1]



1. Cuenta la cantidad de elementos del siguiente diccionario.



El diccionario tiene 4 elementos.

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.

num1 = input(“Ingrese el primer número: ”)

num2 = input(“Ingrese el segundo número: ”)

suma = num1 + num2

print(suma)

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

edad = input(“Ingrese su edad: ”)

faltante = 100 – edad

print(“Le faltan “, faltante, “ años para que cumpla 100 años”)

1. Operadores ternarios.



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

1. Comprobar si un número es par o impar.

var1 = “es par” if num%2 == 0 else “no es par”

1. Obtener el valor absoluto de un número.

var1 = num if num>0 else -num

1. Comparar dos números y obtener el mayor.

el\_mayor = num1 if num1>num2 else num2