**Trabajo práctico nro. 1**

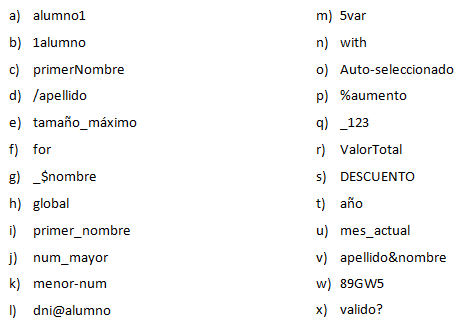
|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Programación I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: |
|  | **Horas semestrales:**  *Cantidad estimada de horas semestrales/anuales.* |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**

|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
| **Cinthia Rigoni** |  | 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.a)válido.

1.b)No válido, porque el número no puede ir antes.

1.c)valido, no es buena práctica.

1.d)No válido, está usando un operador reservado al inicio.

1.e)No valido, utiliza ñ y tilde.

1.f)No válido, palabra reservada para ciclo for.

1.g)No válido, usa un carácter especial.

1.h)No es valido, es una palabra reservada.

1.i)Válido.

1.j)Válido

1.k)No válido, está usando una resta.

1.l)Valido.

1.m)No válido, no puede empezar con un número.

1.n)No válido, es una función reservada.

1.o)No válido, usa una resta.

1.p)No es válido está usando un operador aritmético.

1.q)Válido,

1.r)Valido, no es una buena práctica

1.s) Valido. se utiliza para variables globales.

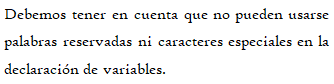
1.t)No es válido, está usando la ñ debería ser ‘anio’

1.u)Valido.

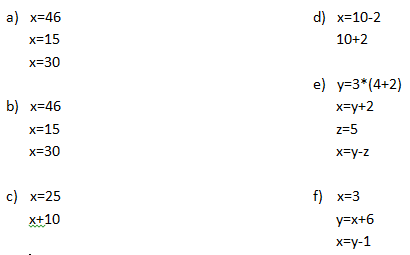
1.v)No es válido, utiliza un operador lógico.

1.w)No es válido es el código del operador ‘and’

1.x)No es válido, tiene signo de pregunta.

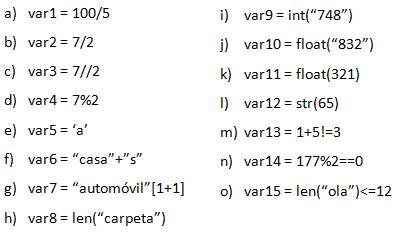


1. **Indica qué dato se guarda en la variable x en cada caso, suponiendo una ejecución secuencial del programa.**



2.a)x=30. 2.b)x=30. 2.c)x=25 2.d)x=8 2.e)x=13 2.f) x=8

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



3.a)Float

3.b)Float

3.c)Int.

3.d)Entero

3.e)String

3.f)String

3.g)String

3.h)entero

3.i)entero

3.j)float

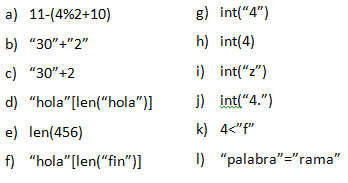
3.l)float

3.m)booleano

3.n)booleano

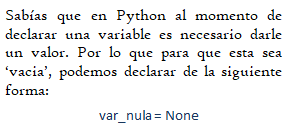
3.o)booleano

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



No son válidas : c,d,e,,j,k,

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



num= 4 (tipó int)

num\_float= 4.5 (tipo float)

var\_complex= 4i (tipo complex)

var\_string = “Hola como estas” (tipo string)

var\_list= ["lista ", " uno", "dos "] (tipo lista)

t = (nombre, 5, 2) (tipo tuple)

var\_dict={‘a’:1,’b’:2}(tipo dict)

var\_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,oaci,ea molnr”
6. **Usando la variable del ejercicio anterior:**
7. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”

Se obtiene usando [::-1] por ejemplo:

cadena =".radna la onimac ecah es ,onimac yah on ,etnanimaC"

cadena\_alreves = cadena[::-1]

print(cadena\_alreves)

1. ¿Cómo obtenemos la subcadena ‘hace’?

cadena=”Caminante, no hay camino, se hace camino al andar.”

cadena\_hace=cadena[ 29:33]

print(cadena\_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 = nombre.title()

print(nombre)

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

frase = "El qUe No arRiesGa, nO gANa"

frase = frase.lower()

print(frase)

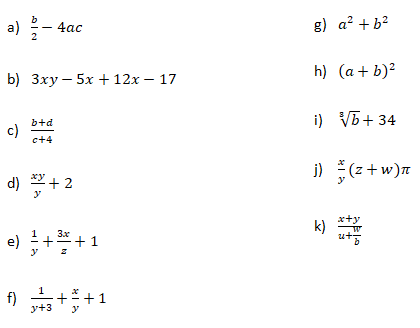
1. Deja esta frase totalmente en letras mayúsculas: ‘El qUe No arRiesGa, nO gANa.’

frase2 = " El qUe No arRiesGa, nO gANa"

frase2 = frase2.upper()

print(frase2)

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

¨

9.a) (b/2) - (4\*a\*c)

9.b) (3\*x\*y) - (5\*x) + (12\*x) - 17

9.c)(b+d)/(c+4)

9.d)x\*y/y +2

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

9.f)1/(y+3)+x/y +1

9.g) a\*\*2+b\*\*2

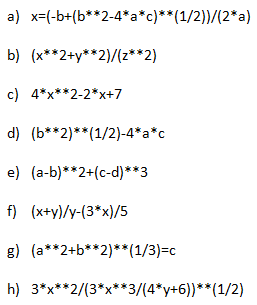
9.h) (a+b)\*\*2

9.i) b\*\*(1/3)+34

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

9.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^2 - (4\*a\*c)) / (2\*a)

b\_(x^2 + y^2) / z^2

c\_4 \* x^2 - 2 x+7

d\_(√b^2)-4ac

e\_(a-b)^2+(c-d)^3

f\_(x+y) / (y - (3x/ 5) )

g\_3√ a^2+b^2

h\_3\*x^2/raiz(3x^3/(4y+6))

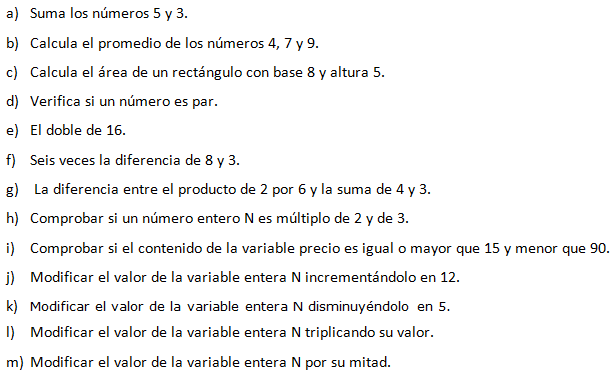
1. **Dada la siguiente expresión aritmética:**



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

Resultado=10.625

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



a) Print (5+3)

b) Print ((4 + 7 + 8)/3 )

c) Print (8\*5)

d) Print (4%2==0)

e) Print (16\*2)

f) Print ((8-3)\*6)

g) Print ((2\*6)-(4+3))

h) Print ((5%3==0) and (5%2==0))

i) precio = 25

j) if 15 <= precio <= 90:

print("es mayor a 15 y menor a 90")

else:

print("fuera de los parametros")

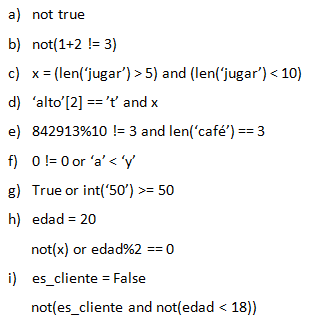
a) Print (N+12)

b) Print (N-5)

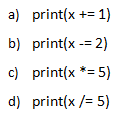
c) Print (N\*3)

d) Print (N/2)

1. **¿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.

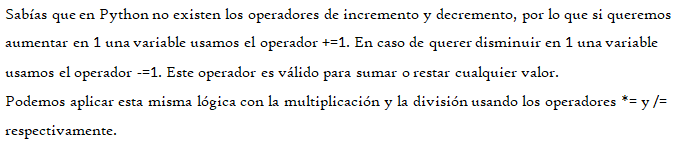


a\_6

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?



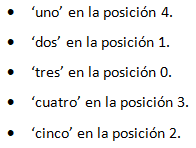
En la tercer posición esta el color “amarillo”, se accede a la posicion colocando el codigo:

print (colores[3])

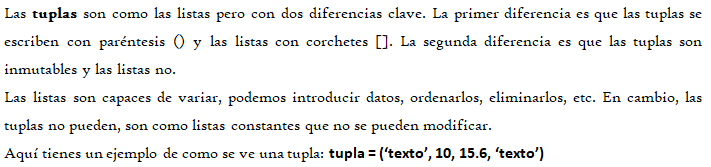
1. ¿En qué posición se encuentra el color ‘rojo’? ¿Y el ‘rosa’?

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

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



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



1. Imprime la segunda posición de esta



Impresión de la segunda posición

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

print ( colores[1])

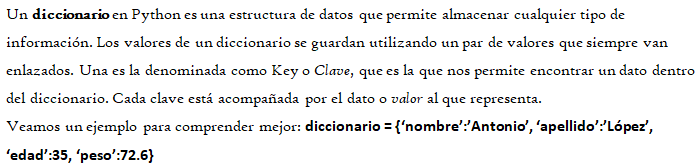
1. Utiliza los símbolos de suma y resta para obtener el resultado 25 a partir de los elementos de la siguiente tupla en una variable llamada operación.



numeros = ( 10,1,5,11)

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

print(operacion)



1. Cuenta la cantidad de elementos del siguiente diccionario.



son 4 elementos

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

diccionario = {"a":1,"b":2,"c":3,"d":4}

clave\_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.

num1=int(input("ingrese el primer numero: "));

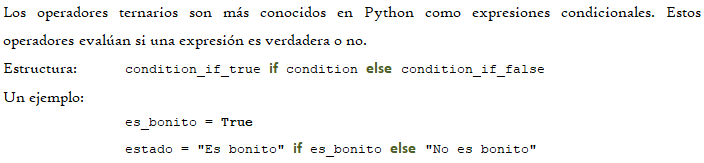
num2=int(input("ingrese el segundo numero: "));

print("la suma de los numeros ingresados es: ",(num1+num2));

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: "));

print("faltan ", (100-edad)," anios 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.

num = 4

numero\_par\_o\_impar= "par" if num%2==0 else "impar"

print(numero\_par\_o\_impar)

1. Obtener el valor absoluto de un número.

num1 = -4

num\_absoluto= num1 if num1>0 else abs(num1)

print( num\_absoluto)

1. Comparar dos números y obtener el mayor.

num1 = 4

num2=5

num\_mayor= num1 if num1>num2 else num2

print(num\_mayor)