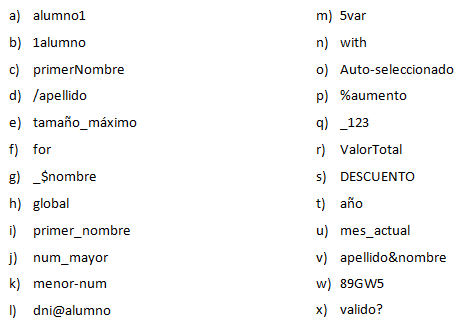
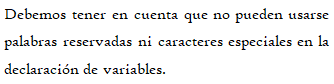
**PROGRAMACIÓN I**

**Trabajo Práctico Nro. 1**

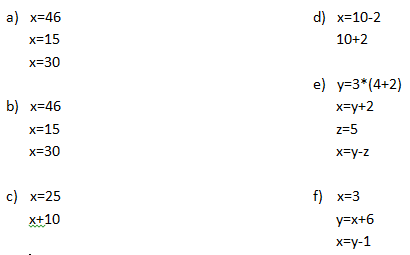
1. Indica si los siguientes identificadores son válidos en Python. En el caso de que el identificador no sea válido, explica el motivo.





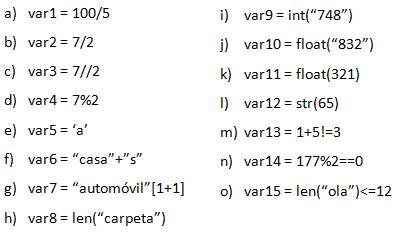
*Respuestas:*

1. *Válido.*
2. *No válido. Comienza con un número.*
3. *No Válido. No se ajusta al estilo de nomenclatura de snake\_case.*
4. *No válido. Comienza con un símbolo.*
5. *No válido. Usa el caracter “ñ”.*
6. *No válido. Palabra reservada.*
7. *No Válido. Usa un símbolo.*
8. *No válido. Palabra reservada.*
9. *Válido.*
10. *Válido.*
11. *No Válido. No se ajusta al estilo de nomenclatura de snake\_case.*
12. *No Válido. No se ajusta al estilo de nomenclatura de snake\_case y usa un símbolo.*
13. *No válido. Comienza con un número.*
14. *No válido. Palabra reservada.*
15. *No Válido. Comienza con mayúscula y usa guión medio para separar.*
16. *No válido. Comienza con un símbolo.*
17. *No válido. Comienza con un número.*
18. *No Válido. Comienza con mayúscula.*
19. *Válido para nombrar variables constantes.*
20. *No válido. Usa el caracter “ñ”.*
21. *Válido.*
22. *No Válido. Usa un símbolo.*
23. *No válido. Comienza con un número.*
24. *No Válido. Usa un símbolo.*
25. Indica qué dato se guarda en la variable x en cada caso, suponiendo una ejecución secuencial del programa.



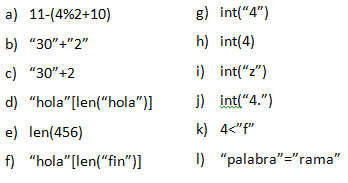
*Respuestas:*

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



*Respuestas:*

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



*Respuestas:*

*c.*

*e.*

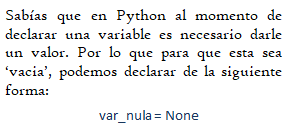
*i.*

*j.*

*k.*

*l.*

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



*Respuestas:*

* *Integer: my\_integer = 10*
* *Floating: my\_float = 3.14*
* *Complex: my\_complex = 2 + 3j*
* *String: my\_string = "Hello, world!"*
* *Boolean: my\_bool = True*
* ***List: my\_list = [var\_1, var\_2, var\_3, var\_4, var\_5]***
* *Tuple: my\_tuple = (1, "hello", 2.5)*
* *Dictionary: my\_dict = {"name": "John", "age": 25, "is\_student": True}*
* *Null: my\_null = None*

1. Teniendo la variable de tipo string: frase = “Caminante, no hay camino, se hace camino al andar.”, indica qué obtendremos si aplicáramos:
2. frase[5]
3. frase[-1]
4. frase[0:8]
5. frase[::3]

*Respuestas:*

1. *a*
2. *.*
3. *Caminant*
4. ***Cin,oaci,ea molnr***
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’?

*Respuestas:*

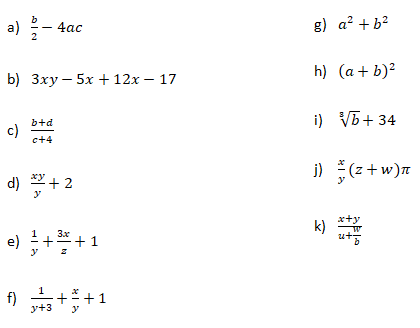
1. *frase[::-1]*
2. *frase[29:33]*
3. Métodos upper(), lower() y title().



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.’

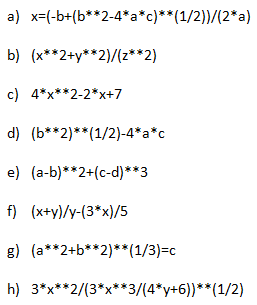
*Respuestas:*

1. *print("lucas mauricio barros".title())*
2. *print("‘El qUe No arRiesGa, nO gANa".lower())*
3. *print("El qUe No arRiesGa, nO gANa".upper())*
4. Convierte en expresiones algorítmicas las siguientes expresiones algebraicas. Coloca paréntesis solamente donde sean necesarios.



*Respuestas:*

1. *resultado = (b / 2) - (4 \* a \* c)*
2. *resultado = (3 \* x \* y) - (5 \* x) + (12 \* x) - 17*
3. *resultado = (b + d) / (c + 4)*
4. *resultado = (x \* y / y) + 2 = x + 2*
5. *resultado = (1 / y) + (3 \* x / z) + 1*
6. *resultado = 1 / (y + 3) + (x / z) + 1*
7. *resultado = resultado = a\*\*2 + b\*\*2*
8. *resultado = (a + b)\*\*2*
9. *resultado = b\*\*(1/3) + 34*
10. *resultado = (v / y) \* (z + w) \* 3.14159*
11. *resultado = (x + y) / (u + (w / b))*
12. Convierte en expresiones algebraicas las siguientes expresiones algorítmicas. Coloca paréntesis solamente donde sean necesarios.



*Respuestas:*

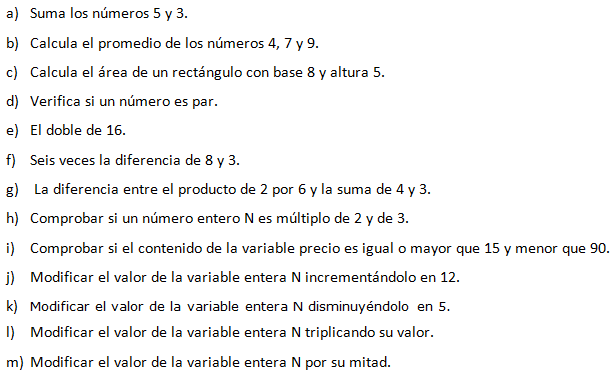
1. *(-b + √(b^2 - 4ac)) / (2a)*
2. *(x^2 + y^2) / z^2*
3. *4x^2 - 2x + 7*
4. *√(b^2) - 4ac*
5. *(a - b)^2 + (c - d)^3*
6. *(x + y) / y - (3x / 5)*
7. *∛(a^2 + b^2) = c*
8. *3x^2 / √(3x^3 / (4y + 6))*
9. Dada la siguiente expresión aritmética:



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

*Respuesta: 11.625*

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



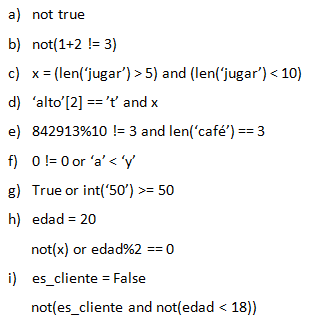
*Respuestas:*

1. *resultado = 5 + 3*
2. *resultado = (4 + 7 + 9) / 3*
3. *base = 8*

*altura = 5*

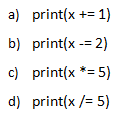
*area = base \* altura*

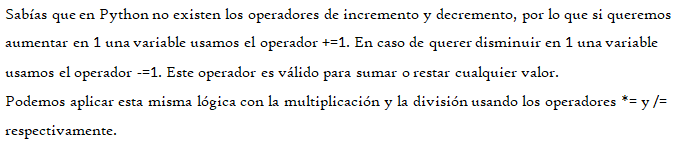
1. *es\_par = num % 2 == 0*
2. *resultado = 16 \* 2*
3. *resultado = 6 \* (8 - 3)*
4. *resultado = (2 \* 6) - (4 + 3)*
5. *es\_multiplo = num % 2 == 0 and num % 3 == 0*
6. *cumple\_condicion = 15 <= precio < 90*
7. *num+= 12*
8. *num -= 5*
9. *num \*= 3*
10. *num /= 2*
11. ¿Qué resultado (True/False) dan las siguientes operaciones?



*Respuestas:*

1. *False*
2. *False*
3. *False*
4. *Error de sintaxis debido a que x no está definida en este contexto*
5. *True*
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.





*Respuestas:*

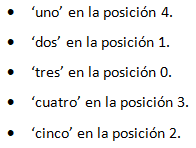
1. *6*
2. *3*
3. *25*
4. *1*
5. Tipos *list*, *tuple* y *dict*.

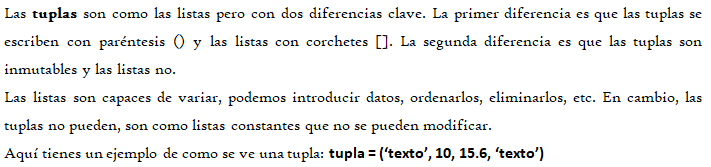


1. De la siguiente lista, ¿qué color está en la posición 3?, ¿cómo accedemos a esta posición?



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.



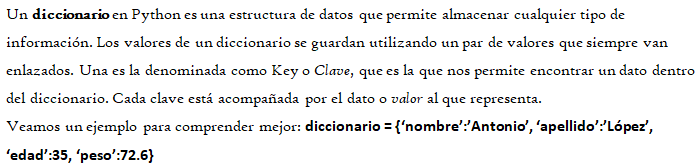


1. Imprime la segunda posición de esta tupla.



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.

*Respuestas:*

1. *color\_en\_posicion\_3 = colores[2]*

*amarillo*

1. *posicion\_rojo = 0*

*posicion\_rosa = 7*

1. ***numeros = ["tres", "dos", "cinco", "cuatro", "uno"]***
2. *segunda\_posicion = colores[1]*

*azul*

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

*4*

1. *valor\_c = diccionario['c']*

*3*

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.

*Respuestas:*

1. *numero1 = float(input("Ingrese el primer número: "))*

*numero2 = float(input("Ingrese el segundo número: "))*

*suma = numero1 + numero2*

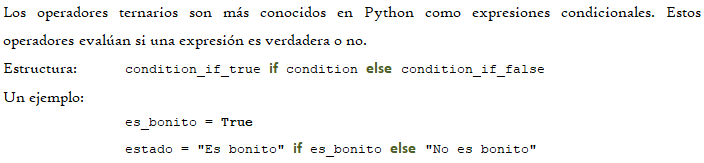
*print("La suma de los números es:", suma)*

1. *edad\_actual = int(input("Ingrese su edad actual: "))*

*anios\_faltantes = 100 - edad\_actual*

*print("Faltan", anios\_faltantes, "años para cumplir 100 años.")*

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.

*Respuestas:*

1. *if numero1 % 2 == 0:*

*resultado\_par\_impar1 = "par"*

*else:*

*resultado\_par\_impar1 = "impar"*

1. *valor\_absoluto\_numero1 = abs(numero1)*
2. *if numero1 > numero2:*

*mayor = numero1*

*elif numero2 > numero1:*

*mayor = numero2*

*else:*

*mayor = "iguales"*