**Trabajo práctico nro. 1**

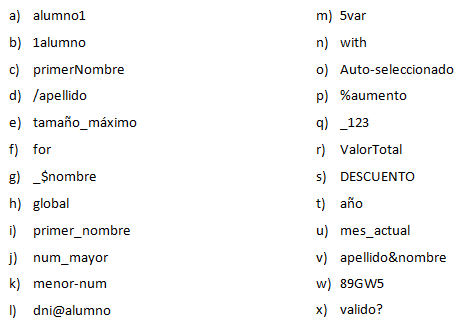
|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Programación I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: |
|  | **Horas semestrales:**  *Cantidad estimada de horas semestrales/anuales.* |
| **Carrera**: *Tecnicatura Universitaria en Programación*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

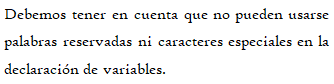
**Integrantes de la Cátedra: Matias Rodriguez, Matias Games, Santiago Bazan, Jorge Garcia.**

* **DOCENTES:**

|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
| **Cinthia Rigon** | Primer Semestre | 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.

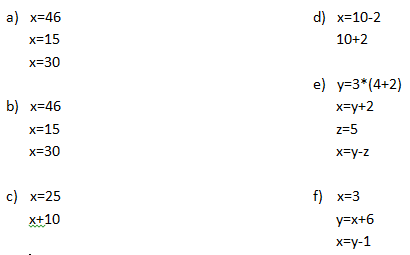




Respuestas (1):

|  |  |
| --- | --- |
| Válidos | No válidos |
| * a) valída * c) valída * i) valida * j) válida * l) válida * r) válida * s) válida pero es una mala práctica usar mayúsculas * u) valida. | * b) invalida, porque empieza con numero * d) invalida, porque empieza con una tecla especial * e) inválida porque usa teclas especiales y ñs * f) invalida porque usa un apalabra reservada * g) invalida usa teclas especiales * h) invalida porque es un palabra reservada del lenguaje * k) invalida usa tecla especiales de por medio * m) invalida porque usa un número al principio * n) invalida porque es una palabra reservada del lenguaje * o) invalida usa tecla especiales de por medio * p) invalida usa teclas especiales * q) valida * t) invalida porque usa ñs * v) invalida usa tecla especiales de por medio * w) invalida porque usa un número al principio * x)invalida usa tecla especiales de por medio |

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

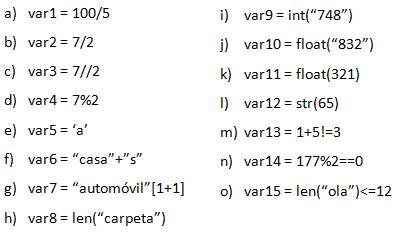


Respuestas (2):

1. x = 30 b) x = 30 c) x = 25

d) x = 8 e) x = 13 f) x = 8

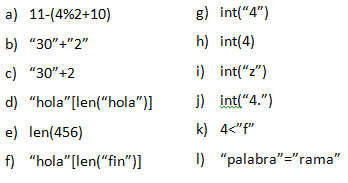
1. Indica qué tipo de dato se guarda en cada variable.



Respuestas (3):

|  |
| --- |
| variables |
| a ) var1 = float(flotante)  b) var2 = float(flotante)  c) var3 = int(Entero)  d) var4 = int(Entero)  e) var5 = str(string)  f) var6 = str(string)  g) var7 = str(string)  h) var8 = int(Entero)  i) var9 = = int(Entero)  j) var10 = float(flotante)  k) var11 = float(flotante)  l) var12 = str(string)  m) var13 = Booleano  n) var14 = Booleano  o) var15 = Booleano |

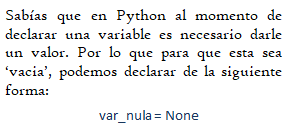
1. Indica cuáles de las siguientes operaciones no son válidas.



Respuestas (4):

|  |
| --- |
| No válidas |
| * c) “30” + 2 * d) “hola” * e) len(456) * i) int(“z”) * j) int(“4.”) * k) 4 < “f” * l) “palabra” = “rama” |

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



Respuestas (5):

* int = 5
* float = 2.56
* complex = 4 - 3i
* string = “hola mundo”
* bool = True
* list = [3, 5, 7, 9,]
* tuple = (2, “hola”, [1,2], 5.4)
* dict = {“nombre” = “matias”, “nombre2” = “rodrigo”}
* 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 (6):

1. a
2. .
3. caminant
4. Cin, oaci, ea, maolnr
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(7)

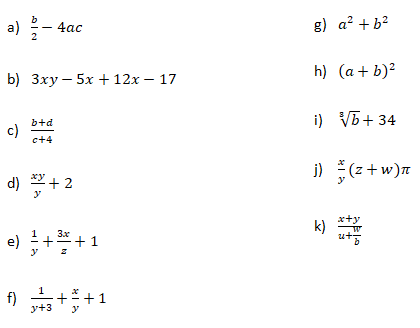
1. Para recorrer un string (o cualquier iterable) de atrás hacia adelante sin el bucle for, usamos los subíndices: frase[::-1]
2. invocamos con un print(x[29:33])
3. Métodos upper(), lower() y title().



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 (8):

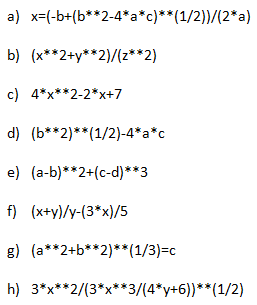
1. print(“lucas mauricio barros”.title()) —> Lucas Mauricio Barros
2. print(“El qUe No arRiesGa, nO gANa”. lower()) —> el que no arriesga no gana.
3. print(“El qUe No arRiesGa, nO gANa”. upper()) —> EL QUE NO ARRIESGA NO GANA.
4. Convierte en expresiones algorítmicas las siguientes expresiones algebraicas. Coloca paréntesis solamente donde sean necesarios.



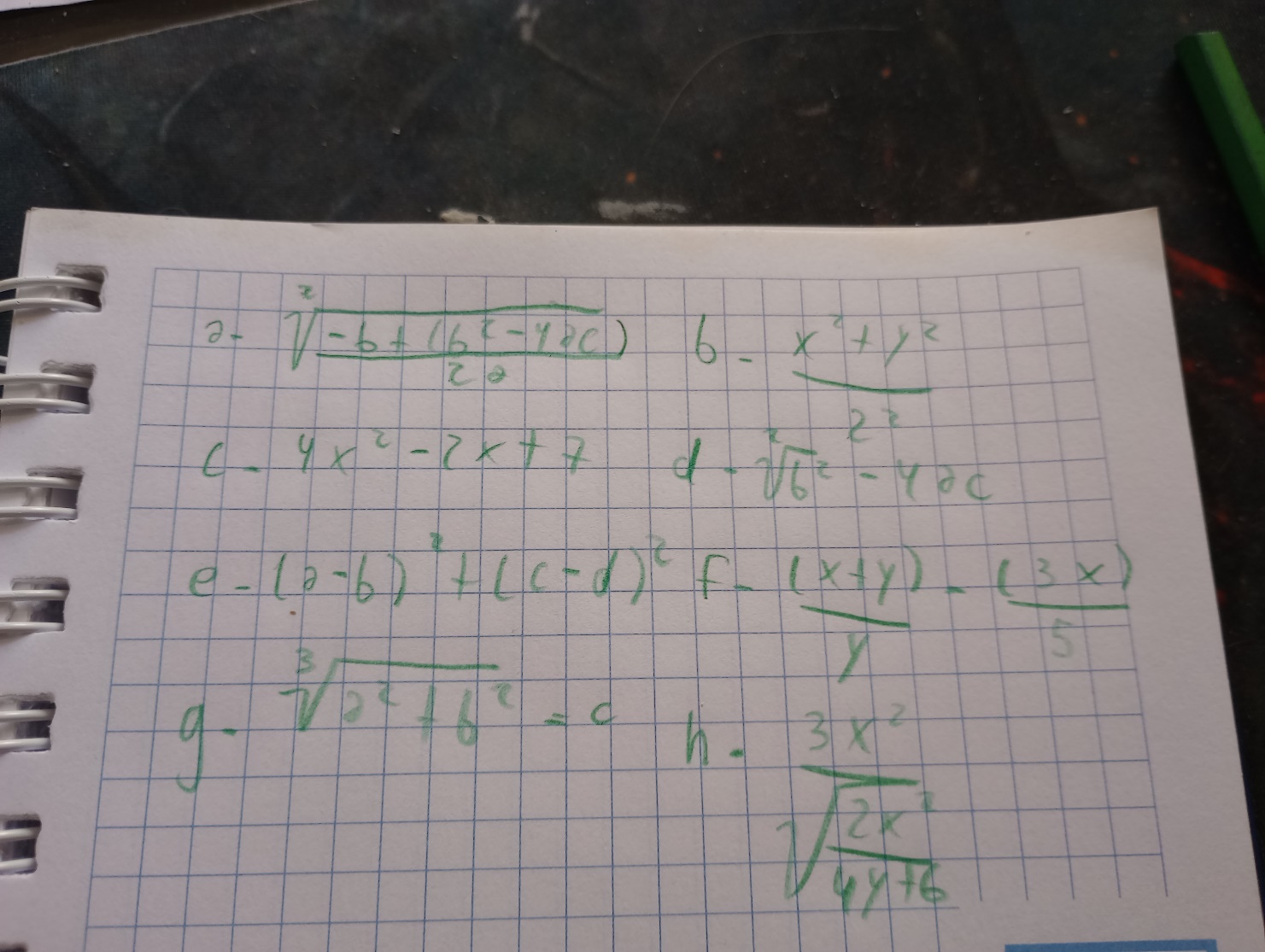
Respuestas (9):

* 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) / 2) + 1
* f) (1/(y+3)) +( x / y ) + 1
* g) a \*\* 2 + b \*\* 2
* i ) b \*\* (⅓) + 34
* j) x/y \* (z+w) \* 3.1416
* k) (x+y) / (u + (w/b))

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



Respuestas (10):



1. Dada la siguiente expresión aritmética:

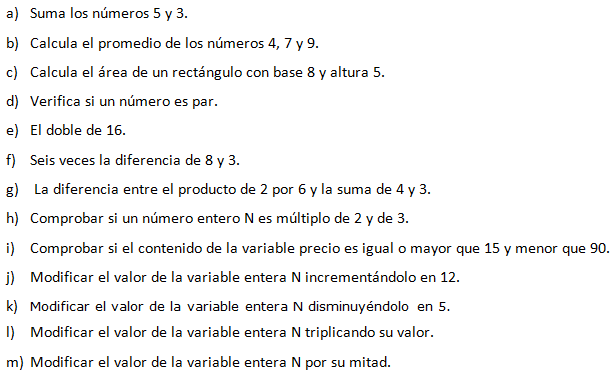


Determinar qué resultado obtendremos si a=5, b=2, c=6, x=(-6) y y=4.

Respuestas (11):

la respuesta es: 10.625

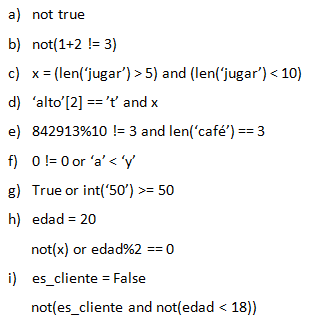
1. Escribe las expresiones algorítmicas equivalentes a los siguientes enunciados:



Respuestas (12):

|  |
| --- |
| * a) print(5+3) * b) print((4+7+9) / 3) * c) base = 8; altura = 5   print(base\*altura)   * d) if (num % 2 == 0):   print(“El numero es PAR”)   * e) print(16\*\*2) * f) print((8-3) \* 6) * g) print((2 \*6) - (4+3)) * if (num % 2 == 0) and (num % 3 == 0):   print(“El numero es multiplo de 2 y 3”)   * f (precio >= 15) and (precio < 90):   print(“El precio es menor que 90 y mayor que 15”)   * j) N = int(input(“numero: “))   N += 12  print(N)   * k) N = int(input(“numero: “))   N -= 5  print(N)   * l) N = int(input(“numero: “))   N \*= 3  print(N)   * m) N = int(input(“numero: “))   N /= 2  print(N) |

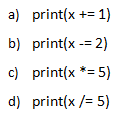
1. ¿Qué resultado (True/False) dan las siguientes operaciones?

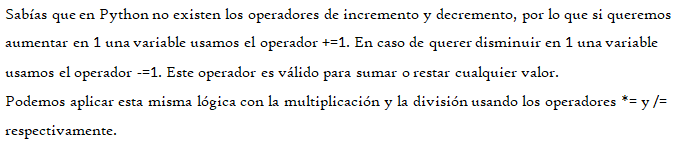


Respuestas (13):

|  |
| --- |
| * 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.





Respuestas (14):

* a) 6
* b) 3
* c) 25
* d) 1

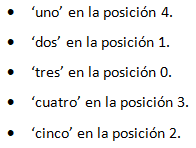
1. Tipos *list*, *tuple* y *dict*.

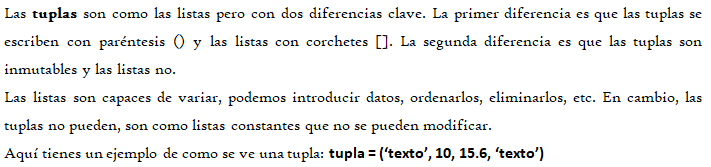


1. De la siguiente lista, ¿qué color está en la posición 3?, ¿cómo accedemos a esta posición?



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.



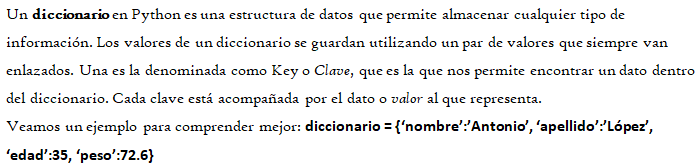


1. Imprime la segunda posición de esta tupla.



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.

Respuesta (15):

* a) muestra el color “amarillo” y lo invocamos con: print(colores[3])
* b) El color “rojo” esta en la posición 1 y el color “rosa” 7
* c) números = [“tres”, “dos”, “cinco”, “cuatro”, “uno”]
* d) print(colores[2])
* e) operación = numeros[0] - numeros[1] + numeros[2] + numeros[3]
* f) El diccionario contiene 4 elementos
* g) print(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.

Respuestas (16):

* a) num1 = int(input(“numero 1: ”))

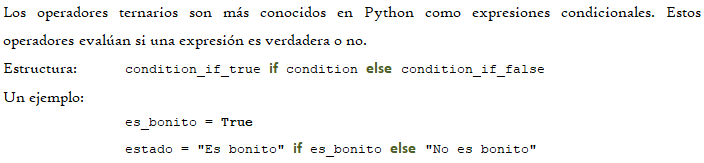
num2 = int(input(“numero 2: ”))

print(num1 + num2)

* b) edad = int(input(“Ingrese su edad: ”)

print(f”A usted le falta {100 - edad} 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

num = 4

par\_nopar = “Es par” if num % 2 == 0 else “No es par”

1. Obtener el valor absoluto de un número.

num = 4

valor\_absoluto = num \* -1 if num < 0 else num

1. Comparar dos números y obtener el mayor.

num1 = 5

num2 = 7

mayor = num1 if num1 > num2 else num2