**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.





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



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



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



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



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]
6. Usando la variable del ejercicio anterior:
7. ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimaC”
8. ¿Cómo obtenemos la subcadena ‘hace’?
9. 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. Convierte en expresiones algorítmicas las siguientes expresiones algebraicas. Coloca paréntesis solamente donde sean necesarios.



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



1. Dada la siguiente expresión aritmética:



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

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



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



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





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.
2. 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.
3. 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. Válido. | m) Inválido. El nombre de una variable no  puede empezar con un número. |
| 1. Inválido. El nombre de una variable no puede empezar con un número. | n) Inválido. La palabra “with” es una palabra  reservada de Python. |
| 1. Es válido, pero es una muy mala práctica en éste idioma. | o) Inválido. El signo “-“ representa una resta  en Python, por lo que arrogará un error en  éste caso, ya que no se puede llevar a cabo  una resta entre cadenas. |
| 1. Inválido. El símbolo “/” está reservado por Python. | p) Inválido. Una variable no puede comenzar  con un operador. |
| 1. Válido. | q) Válido. |
| 1. Inválido. La palabra “for” está reservada por Python. | r) Inválido. El nombre de una variable no  puede comenzar con una mayúscula. |
| 1. Válido. | s) Válido, pero es una muy mala práctica. |
| 1. Inválido. La palabra “global” es una palabra reservada de Python. | t) Inválido. La letra “ñ” no es una letra para  Python y lanzará un error. |
| 1. Válido. | u) Válido. |
| 1. Válido. | v) Inválido. El signo “&” está reservado por  Python. |
| 1. Inválido. El signo “-“ representa una resta en Python, por lo que arrogará un error en éste caso, ya que no se puede llevar a cabo una resta entre cadenas. | w) Inválido. El nombre de una variable no  puede empezar con un número. |
| 1. Válido. | x) Válido. |

|  |  |
| --- | --- |
| 1. El dato que se guarda en la variable x es: 30 | 1. El dato que se guarda en la variable x es: 8 |
| 1. El dato que se guarda en la variable x es: 30 | 1. El dato que se guarda en la variable x es: 15 |
| 1. El dato que se guarda en la variable x es: 25 | 1. El dato que se guarda en la variable x es: 8 |

|  |  |
| --- | --- |
| 1. Tipo: Float | 1. Tipo: Integer |
| 1. Tipo: Float | 1. Tipo: Integer |
| 1. Tipo: Integer | 1. Tipo: Integer |
| 1. Tipo: Integer | 1. Tipo: String |
| 1. Tipo: String | 1. Tipo: Integer |
| 1. Tipo: String | 1. Tipo: Boolean |
| 1. Tipo: String | 1. Tipo: Boolean |
| 1. Tipo: Integer |  |

|  |  |
| --- | --- |
| 1. Válida. | g) Válida. |
| 1. Válida. | h) Inválida. |
| 1. Inválida. | i) Inválida. |
| 1. Inválida. | j) Inválida. |
| 1. Inválida. | k) Inválida. |
| 1. Inválida. | l) Válida. |

|  |  |
| --- | --- |
| variable\_01: 18.9 | variable\_06: [2, 4, 6, 8, 10] |
| variable\_02: 23 | variable\_07: (2, 4, 6, 8, 10) |
| variable\_03: 531+(3\*45) | variable\_08: {“h”: 3, “u”: 9, “f”: 1, “l”: 5} |
| variable\_04: “Hola” | variable\_09: None |
| variable\_05: True |  |

|  |  |
| --- | --- |
| 1. “a” | c) “Caminant” |
| 1. “.” | 1. “maenh mosheanaaa” |

1. La obtenemos usando [::-1].
2. La obtenemos haciendo uso de éste código:

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

first= frase.find(“hace”)

last= first + len(“hace”)

print(frase[first:last]

Y sino, usando [29:32:1]

1. nombre= ‘lucas mauricio barros’

nombre= nombre.title()

1. frase= ‘El qUe No arRiesGa, nO gANa.’

frase= frase.lower()

1. frase= ‘El qUe No arRiesGa, nO gANa.’

frase= frase.upper()

|  |  |
| --- | --- |
| 1. b / 2 – 4 \* a \* c | 1. (a \*\* 2) + (b \*\* 2) |
| 1. 3 \* x \* y – 5 \* x + 12 \* x – 17 | 1. (a + b) \*\* 2 |
| 1. (b + d) / (c + 4) | 1. (b \*\* (1 / 3)) + 34 |
| 1. ((x \* y) / y) + 2 | 1. (x / y) \* (z + w) \* 1.14 |
| 1. (1 / y) + ((3 \* x) / z) + 1 | 1. (x + y) / (u + (w / b)) |
| 1. (1 / (y + 3)) + (x / y) + 1 |  |

|  |  |
| --- | --- |
|  |  |
|  | 1. - |
| 1. 4-2x+7 |  |
| 1. -4ac |  |

1. El resultado que obtendremos dados esos valores es 10.625.

|  |  |
| --- | --- |
| 1. 5+3 | 1. if N % 2 == 0 and N % 3 == 0: |
| 1. 3/(4+7+9) | 1. if 15 <= precio < 90: |
| 1. 8\*5 | 1. N += 12 |
| 1. X%2=0 | 1. N -= 5 |
| 1. 16\*2 | 1. N \*= 3 |
| 1. (8-3)\*\*6 | 1. N /= 2 |
| 1. (2\*6)-(4+3) |  |

|  |  |
| --- | --- |
| 1. Da “false”. | 1. Da “true”. |
| 1. Da “false”. | 1. Da “true”. |
| 1. Da “true” | 1. Da error o “true”. |
| 1. Da error o “false”. | 1. Da “true”. |
| 1. Da “false”. |  |

En todos los casos, daría un error de sintaxis, ya que no es posible realizar éstas operaciones DENTRO de un “print()”, fuera sí, pero dentro no. En todo caso, si esto fuera válido o estuviera bien hecho, éstos serían los valores que se mostrarían:

|  |  |
| --- | --- |
| 1. Se mostrará “6”. | 1. Se mostrará “25”. |
| 1. Se mostrará “3”. | 1. Se mostrará “1.0”. |

1. En la posición 3 se encuentra el color “amarillo” y se puede acceder a ésta posición de la siguiente manera: colores[3]
2. El color rojo se encuentra en la posición 0, y el color rosa en la posición 7.
3. list\_valores= [“tres”, “dos”, “cinco”, “cuatro”, “uno”]
4. “verde”
5. operación= 10 + 11 – 1 + 5
6. Dentro del diccionario especificado se encuentran 4 elementos en total.
7. diccionario[“c”]
8. num1= input(“Ingrese el primer número que desee: “)

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

suma= num1 + num2

print(“La suma de los dos números ingresados da ”, suma)

1. edad= input(“Ingrese su edad: ”)

faltante= edad – 100

print(“A usted le faltan ”, faltante + “ años para alcanzar a sus 100 años de vida”)

1. num= 18

def= true if num % 2 == 0 else false

print(def)

1. numero= 5

abs\_numero= numero if numero >= 0 else –numero

print(abs\_numero)

1. valor1= 30

valor2= 20

mayor= valor1 if valor1 > valor2 else valor2

print(mayor)