

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

a) alumno1	m) 5var
b) 1alumno	n) with
c) primerNombre	o) Auto-seleccionado
d) /apellido	p) %aumento
e) tamaño_máximo	q) _123
f) for	r) ValorTotal
g) _\$nombre	s) DESCUENTO
h) global	t) año
i) primer_nombre	u) mes_actual
j) num_mayor	v) apellido&nombre
k) menor-num	w) 89GW5
l) dni@alumno	x) valido?

- a = es válido.
- b = no es válido porque empieza con un número.
- c = es válido pero no es buena práctica.
- d = no es válido porque empieza con un operador.
- e = es válido pero no es buena práctica.
- f = no es válido porque es una palabra reservada.
- g = no es válido porque tiene un carácter especial.
- h = no es válido porque es una palabra reservada.
- i = es válido.
- j = es válido.
- k = no es válido porque tiene un carácter especial.
- l = no es válido porque tiene un carácter especial.
- m = no es válido porque empieza con un número.
- n = no es válido porque tiene una palabra reservada.
- o = no es válido porque tiene un carácter especial.
- p = no es válido porque empieza con un operador.
- q = es valido.
- r = es válido pero no es buena práctica.
- s = es válido, porque es una variable global.
- t = es válido pero no es buena práctica.
- u = es válido.
- v = no es válido porque tiene un operador
- w = no es válido porque empieza con un número.
- x =no es válido porque tiene un carácter especial.

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

- a) $x=30$
- b) $x=30$
- c) $x=25$
- d) $x=8$
- e) $x=13$
- f) $x=8$

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

- a)float
- b)float
- c)integer
- d)integer
- e)string
- f)string
- g)string
- h)integer
- i)integer
- j)float
- k)float
- l)string
- m)booleano
- n)booleano
- o)booleano

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

- a) Operación Válida
- b) Operación Válida
- c) No es Válida
- d) No es Válida
- e) No es válida
- f) Operación Válida
- g)Operación Válida
- h) Operación Válida
- i) No es Válida
- j) No es Válida

k) No es Válida

l) No es Válida

5. Declara una variable de cada tipo de dato y asigne un valor.

INT	FLOAT	COMPLEX	STRING	BOOL	LIST	TUPLE	DICT	NULL
var1 = 12	var1 = 10.2	var3 = 3+2j	var4 = "Hola Niño"	var1 = true	var6 = "a" var5 = "b" var7="c" var_lista = [var6, var5, var7]	var6 = "a" var5 = "b" var7="c" var_lista = (var6, var5, var7)	diccionario={"Nombre":"Andy", "Edad":19}	variable = None

- int
- float
- complex
- string
- bool
- list
- tuple
- dict
- null

Sabías que en Python al momento de declarar una variable es necesario darle un valor. Por lo que para que esta sea 'vacía', podemos declarar de la siguiente forma:

```
var_nula = None
```

6. Teniendo la variable de tipo string: frase = "Caminante, no hay camino , se hace camino al andar." Indica que obtendríamos si aplicáramos:

- frase[5] = a
- frase[-1] = r
- frase[0:8] = Caminante
- frase[:3] = Cin,oaci,ea moln

7. Usando la variable del ejercicio anterior:

- ¿Cómo obtendremos la cadena al revés?
- ¿Cómo obtendremos la subcadena 'hace'?

7. Usando la variable del ejercicio anterior:

a) ¿Cómo obtenemos la cadena al revés? “.radna la onimac ecah es ,onimac yah on ,etnanimac”

```
print(frase[::-1])
```

b) ¿Cómo obtenemos la subcadena ‘hace’?

```
var8=frase[-30::-1]  
print(var8[0:4])
```

8. Métodos, upper(),lower(), title()

a. Pon en mayúsculas la primera letra de cada palabra del siguiente nombre: ‘lucas mauricio barros’.

```
var1 = 'lucas mauricio barros'  
print(var1.title())
```

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

```
1 var1 = 'El qUe No arRiesGa, nO gANa.'  
2  
3 print(var1.lower())
```

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

```
1 var1 = 'El qUe No arRiesGa, nO gANa.'  
2  
3 print(var1.upper())
```

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

a) $\frac{b}{2} - 4ac$

g) $a^2 + b^2$

b) $3xy - 5x + 12x - 17$

h) $(a + b)^2$

c) $\frac{b+d}{c+4}$

i) $\sqrt[3]{b} + 34$

d) $\frac{xy}{y} + 2$

j) $\frac{x}{y}(z + w)\pi$

e) $\frac{1}{y} + \frac{3x}{z} + 1$

k) $\frac{x+y}{u+\frac{w}{b}}$

f) $\frac{1}{y+3} + \frac{x}{y} + 1$

a. Punto A:

```
1 a = 1
2 b = 2
3 c = 3
4
5 print((b/2) - (4*a*c))
```

b. Punto B:

```
1 x = 1
2 y = 2
3
4 print((3*x*y) - (5*x) + (12*x) - 17)
```

c. Punto C:

```
1 b = 1
2 c = 2
3 d = 3
4
5 print((b+d) / (c+4))
```

d. Punto D:

```
clases / Clase dos.py > ...  
1 x = 1  
2 y = 2  
3  
4 print(((x*y) / y) + 2)
```

e. Punto E:

```
clases / Clase dos.py > ...  
1 x = 1  
2 y = 2  
3 z = 3  
4  
5 print((1/y) + ((3*x)/z) + 1)
```

f. Punto F:

```
clases / Clase dos.py > ...  
1 x = 1  
2 y = 2  
3  
4 print((1/(y+3)) + (x/y) + 1)
```

g. Punto G:

```
1 a = 1  
2 b = 2  
3  
4 print(a**2 + b**2)
```

h. Punto H:

```
1 a = 1  
2 b = 2  
3  
4 print((a+b)**2)
```

i. Punto I:

```
1 b = 2  
2  
3 print(b**(1/3) + 34)
```

j. Punto J:

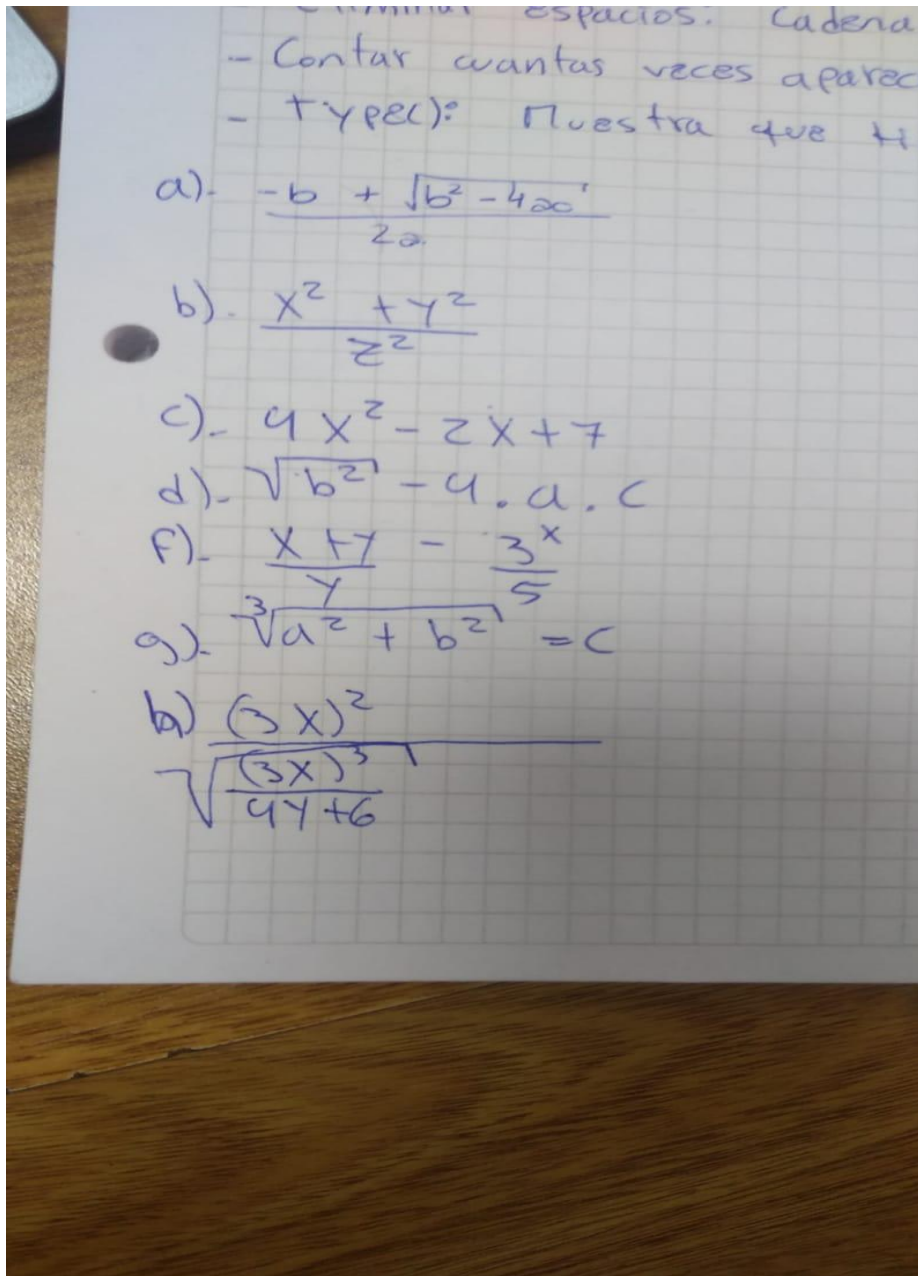
```
1 import math
2
3 pi = math.pi
4
5 x = 1
6 y = 2
7 z = 3
8 w = 4
9 print((x/y)*(z+w)* pi)
```

k. Punto K:

```
1 x = 1
2 y = 2
3 u = 3
4 w = 4
5 b = 6
6
7 print((x+y) / u + (w/b))
```

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

- a) $x = (-b + (b^2 - 4ac)^{1/2}) / (2a)$
- b) $(x^2 + y^2) / (z^2)$
- c) $4x^2 - 2x + 7$
- d) $(b^2)^{1/2} - 4ac$
- e) $(a-b)^2 + (c-d)^3$
- f) $(x+y) / y - (3x) / 5$
- g) $(a^2 + b^2)^{1/3} = c$
- h) $3x^2 / (3x^3 / (4y+6))^{1/2}$



11. Dada la siguiente expresión aritmética:

$$a + b * \left(5 - \frac{c}{2}\right) + (7 - x)/(y + 4)$$

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

Respuesta: 10.625

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

- a) Suma los números 5 y 3.
- b) Calcula el promedio de los números 4, 7 y 9.
- c) Calcula el área de un rectángulo con base 8 y altura 5.
- d) Verifica si un número es par.
- e) El doble de 16.
- f) Seis veces la diferencia de 8 y 3.
- g) La diferencia entre el producto de 2 por 6 y la suma de 4 y 3.
- h) Comprobar si un número entero N es múltiplo de 2 y de 3.
- i) Comprobar si el contenido de la variable precio es igual o mayor que 15 y menor que 90.
- j) Modificar el valor de la variable entera N incrementándolo en 12.
- k) Modificar el valor de la variable entera N disminuyéndolo en 5.
- l) Modificar el valor de la variable entera N triplicando su valor.
- m) Modificar el valor de la variable entera N por su mitad.

Respuesta:

- a) $\text{suma} = 5 + 3$
- b) $\text{suma} = 4 + 7 + 9$
 $\text{promedio} = \text{suma} / 3$
- c) $\text{base} = 8$
 $\text{altura} = 5$
 $\text{area_rectangulo} = \text{base} * \text{altura}$
- d) $\text{resto} = \text{num} \% 2$
 $\text{es_par} = \text{resto} == 0$
- e) $\text{exponente} = 16 * 2$
- f) $\text{resta} = 8 - 3$
 $\text{multiplicacion_por_seis} = \text{resta} * 6$
- g) $\text{producto} = 2 * 6$
 $\text{suma} = 4 + 3$
 $\text{diferencia} = \text{producto} - \text{suma}$
- h) $\text{par} = N \% 2$
 $\text{impar} = N \% 3$
 $\text{es_multiplo} = \text{par} \& \text{impar} == 0$

- i) `condicion= precio > 15 & precio < 90`
- j) `incremento = N + 12`
- k) `disminucion = N - 5`
- l) `triple = N * 3`
- m) `mitad = N / 2`

13. ¿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

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

- a) `print(x += 1)`
- b) `print(x -= 2)`
- c) `print(x *= 5)`
- d) `print(x /= 5)`

a = 6
b = 3
c = 25
d = 1

15. Tipos list, tuple y dict.

Una **lista** es una variable con múltiples valores. Pueden contener cualquier tipo de dato soportado por Python en cualquier orden.

Por ejemplo: `lista = ['texto', 10, 15.6, 'texto']`

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

```
colores = ["rojo", "azul", "verde", "amarillo", "marrón", "lila", "negro", "rosa"]
```

La posición 3 se mostrara en amarillo

b) ¿En qué posición se encuentra el color 'rojo'? ¿Y el 'rosa'?

rojo = 0

rosa = 7

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

- 'uno' en la posición 4.
- 'dos' en la posición 1.
- 'tres' en la posición 0.
- 'cuatro' en la posición 3.
- 'cinco' en la posición 2.

`lista_num = ['tres', 'dos', 'cinco', 'cuatro', 'uno']`

Las **tuplas** son como las listas pero con dos diferencias clave. La primer diferencia es que las tuplas se escriben con paréntesis `()` y las listas con corchetes `[]`. La segunda diferencia es que las tuplas son inmutables y las listas no.

Las listas son capaces de variar, podemos introducir datos, ordenarlos, eliminarlos, etc. En cambio, las tuplas no pueden, son como listas constantes que no se pueden modificar.

Aquí tienes un ejemplo de como se ve una tupla: `tupla = ('texto', 10, 15.6, 'texto')`

d) Imprime la segunda posición de esta tupla.

```
colores = ('rojo', 'azul', 'verde', 'amarillo', 'marrón', 'lila', 'negro', 'rosa', 'blanco', 'naranja')
```

```
print(colores[1])
```

e) 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.

```
numeros = (10, 1, 5, 11)
```

Un **diccionario** en Python es una estructura de datos que permite almacenar cualquier tipo de información. Los valores de un diccionario se guardan utilizando un par de valores que siempre van enlazados. Una es la denominada como Key o Clave, que es la que nos permite encontrar un dato dentro del diccionario. Cada clave está acompañada por el dato o valor al que representa. Veamos un ejemplo para comprender mejor: `diccionario = {'nombre': 'Antonio', 'apellido': 'López', 'edad': 35, 'peso': 72.6}`

```
print(numeros[0]-numeros[1] + numeros[2]+numeros[3])
```

f) Cuenta la cantidad de elementos del siguiente diccionario.

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

```
print(len(diccionario))
```

g) Accede al valor de la clave 'c' en el diccionario.

```
print(diccionario["c"])
```

16. Vamos a practicar el uso de las funciones **input()** y **print()**.

Ejemplo: Solicita el nombre de una persona e imprime un mensaje de bienvenida.

```
nombre = input("Ingresa tu nombre: ")  
print("¡Hola", nombre + "! Bienvenido(a).")
```

a) Solicita dos números al usuario, súmalos e imprime el resultado.

```
19 num1 = int(input("Ingrese el primer número: "))
20 num2 = int(input("Ingrese el segundo número: "))
21
22 print(f"El resultado entre \"{num1}\" y \"{num2}\" es: \"{num1+num2}\"")
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1> & 'C:\Users\Lenovo\AppData\Local\Microsoft\Windows\apps\python\python-2023.15.12191008\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\adapter\Programacion 1\Clases\clase-3.py'

Ingrese el primer número: 1
Ingrese el segundo número: 2
El resultado entre "1" y "2" es: "3"
PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1> |

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

```
19 num1 = int(input("Ingrese la edad de la persona: "))
20
21 print(f'La edad que le falta para que se cumplan 100 años es: ', 100-num1)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1> & 'C:\Users\Lenovo\AppData\Local\Microsoft\Windows\apps\python\python-2023.15.12191008\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\adapter\Programacion 1\Clases\clase-3.py'

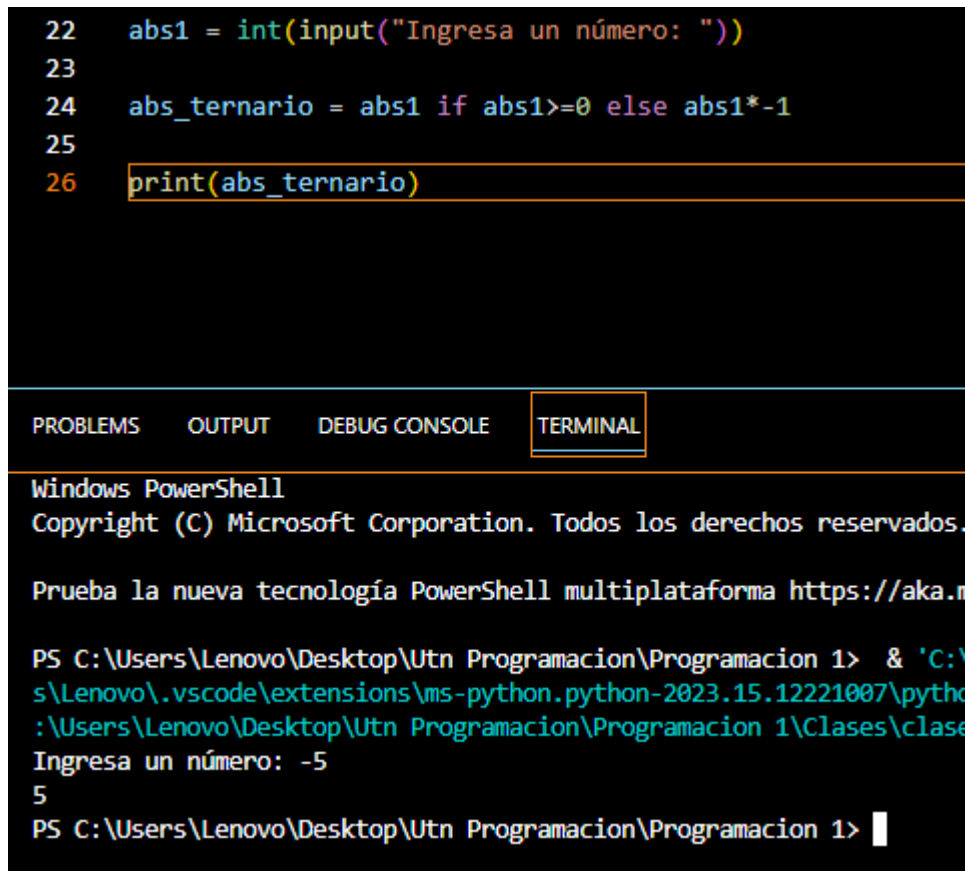
Ingrese la edad de la persona: 22
La edad que le falta para que se cumplan 100 años es: 78
PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1> |

17). Operadores ternarios.

a). Comprobar si un número es par o impar.

```
numero = int(input("ingresa un numero: "))  
  
num_par = "es par" if numero %2==0 else "no es par"  
  
print(num_par)
```

b). Obtener el valor absoluto de un número.



```
22  abs1 = int(input("Ingresa un número: "))  
23  
24  abs_ternario = abs1 if abs1>=0 else abs1*-1  
25  
26  print(abs_ternario)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/powershell>

PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1> & 'C:\Users\Lenovo\.vscode\extensions\ms-python.python-2023.15.12221007\python\Scripts\python.exe' C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1\Clases\clase1.py
Ingresa un número: -5
5
PS C:\Users\Lenovo\Desktop\Utn Programacion\Programacion 1>

c)

```
numero = int(input("ingresa un numero: "))  
  
numero2 = int(input("ingresa un segundo numero: "))  
  
num = "numero 1 es mayor" if numero > numero2 else "numero 2 es mayor"  
  
print(num)
```