**Programación 1**

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

1. alumno1
2. 1alumno
3. primerNombre
4. /apellido
5. tamaño\_máximo
6. for
7. \_$nombre
8. global
9. primer\_nombre
10. num\_mayor
11. menor-num
12. dni@alumno
13. 5var
14. with
15. Auto-seleccionado
16. %aumento
17. \_123
18. ValorTotal
19. DESCUENTO
20. año
21. mes\_actual
22. apellido&nombre
23. 89GW5
24. valido?

b) , m), w) empieza por un numero

e) , t) depende la versión de python, aunque es mejor no usar “ñ”

d) , p) empieza por un carácter especial

f) , h) , n) usa una palabra reservada

k) , o) lleva un guion (puede ser una resta)

l) , v) , g) , x) lleva un carácter especial

Texto

El contenido generado por IA puede ser incorrecto.

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



1. x = 30
2. x = 30
3. x = 25
4. x = 8
5. x = 13
6. x = 8

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



1. float
2. float
3. int
4. int
5. string [no existe “char” en .py]
6. string
7. string
8. int
9. int
10. float
11. float
12. string
13. bool
14. bool
15. bool

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



c) operación entre str ^ int

d) operación entre str ^ int

e) operación entre str (len) ^ int

i) operación que intenta convertir str (“z”) a int

j) operación que intenta convertir str (“4.”) a int

k) operación entre int ^ str

l) operación entre str ^ str

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



int: num = 2

float: num\_f = 2.5

complex: compl = 3+4i

string: greet = “hola”

bool: verdad = true

list: lista = [1, 2, 3, 4] #se puede modificar a futuro

tuple: tupla = (1, 2, 3, 4) #no se puede modificar

dict: datos\_alumn = {"nombre" : "Santiago", "edad":20}

null: nulo = None

