Silva Gustavo Nicolás – Programación 1 – Turno Mañana

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.

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

a) Válido
b) Inválido, comienza con un número
c) Válido, pero se recomienda el uso de snake_case
d) Inválido, comienza con un símbolo
e) Válido, pero el uso de ñ y de acentos es mala práctica
f) Inválido, es una palabra reservada
g) Válido
h) Inválido, es una palabra reservada
i) Válido
j) Válido
k) Inválido, el símbolo "-" es un operador
l) Inválido, contiene un símbolo
m) Inválido, comienza con un numero
n) Inválido, es una palabra reservada
o) Inválido, "-" es un operador
p) Inválido, comienza con un símbolo
q) Válido
r) Válido, pero se recomienda el uso de snake_case
s) Válido, pero el uso de mayúsculas en declaración de variables es una mala practica
t) Válido, pero el uso de ñ en declaración de variables es una mala practica
u) Válido, pero el uso de ñ en declaración de variables es una mala practica
v) Inválido, contiene un símbolo
w) Inválido, comienza con un numero
x) Inválido, contiene un símbolo

Debemos tener en cuenta que no pueden usarse palabras reservadas ni caracteres especiales en la declaración de variables.

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

a)
```
x=46
x=15
x=30
```

b)
```
x=46
x=15
x=30
```

c)
```
x=25
x+10
```

d)
```
x=10-2
10+2
```

e)
```
y=3*(4+2)
x=y+2
z=5
x=y-z
```

f)
```
x=3
y=x+6
x=y-1
```

a) 30
b) 30
c) 45
d) 8
e) 20
f) 8

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

a) var1 = 100/5
b) var2 = 7/2
c) var3 = 7//2
d) var4 = 7%2
e) var5 = 'a'
f) var6 = "casa"+"s"
g) var7 = "automóvil"[1+1]
h) var8 = len("carpeta")
i) var9 = int("748")
j) var10 = float("832")
k) var11 = float(321)
l) var12 = str(65)
m) var13 = 1+5!=3
n) var14 = 177%2==0
o) var15 = len("ola")<=12

a) float
b) float
c) int
d) int
e) str
f) str
g) str
h) int
i) int
j) float
k) float
l) str
m) bool
n) bool
o) bool

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

a) 11-(4%2+10)
b) "30"+"2"
c) "30"+2
d) "hola"[len("hola")]
e) len(456)
f) "hola"[len("fin")]
g) int("4")
h) int(4)
i) int("z")
j) int("4.")
k) 4<"f"
l) "palabra"="rama"

a) Válida → 1
b) Valida → 302
c) Inválida → El tipo de dato de aquellos a los lados del operador no coinciden
d) Válida → Pero retorna un error de índice fuera de rango, pues el resultado de la operación es el índice 4 de la cadena "hola", que contiene solo hasta el 3 (0, 1, 2 y 3)
e) Inválida → La función len no es aplicable a números
f) Válida → "a"
g) Válida → 4
h) Válida → 4
i) Inválida → Una letra no es convertible a entero, se obtiene un error "ValueError"
j) Inválida → La cadena representa un valor float, no es posible convertirla en entero sin antes convertirla en float
k) Inválido → Es posible comparar una letra y un numero siempre y cuando ambos sean comparados como cadena, en tal caso sería True
l) Inválido → No es posible asignar el valor de una cadena a otra

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

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

int → num_1 = 12

float → mi_altura = 1.8

complex → num_complejo = 5 + 2j

string → mi_nombre = "Nicolas"

bool → es_mayor = 5>2

list → lista_numeros = [1, 2, 3, 4, 5]

tuple → tupla_numeros = (1, 2, 3, 4, 5)

dict → diccionario_datos = {"nombre" : "Nico", "edad" : 22, "país" : "Argentina"}

null → variable_vacia = None

Sabías que en Python al momento de declarar una variable es necesario darle un valor. Por lo que para que esta sea 'vacia', 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 qué obtendríamos si aplicáramos:

   a) frase[5] → "a"
   b) frase[-1] → "."
   c) frase[0:8] → "Caminant"
   d) frase[::3] → "Cin,oaci,ea molnr"

7. Usando la variable del ejercicio anterior:

   a) ¿Cómo obtenemos la cadena al revés? ".radna la onimac ecah es ,onimac yah on ,etnanimaC"

   frase[::-1]

   b) ¿Cómo obtenemos la subcadena 'hace'?

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
frase = "Caminante, no hay camino, se hace camino al andar."
inicio = frase.find("hace")  # Encuentra la posición de inicio de "hace"
if inicio != -1:  # Verifica si se encontró la palabra "hace"
    subcadena_hace = frase[inicio:inicio + 4]  # Obtiene la subcadena "hace"
    print(subcadena_hace)
else:
    print("La palabra 'hace' no se encuentra en la frase.")
```

```
[Running] python -u "d:\Practica y testeo - Python\Testeo - TP1.py"
hace

[Done] exited with code=0 in 0.084 seconds
```

8. Métodos upper(), lower() y title().

El método **title()** cambia la primera letra de cada palabra a mayúscula.

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

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```
1   nombre = "lucas mauricio barros"
2   nombre_en_mayusculas = nombre.title()
3   print(nombre_en_mayusculas)
```

```
[Running] python -u "d:\Practica y testeo - Python\Testeo - TP1.py"
Lucas Mauricio Barros

[Done] exited with code=0 in 0.092 seconds
```

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

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```
1   frase = "El qUe No arRiesGa, nO gANa."
2   frase_minus = frase.lower()
3   print(frase_minus)
```

```
[Running] python -u "d:\Practica y testeo - Python\Testeo - TP1.py"
el que no arriesga, no gana.

[Done] exited with code=0 in 0.098 seconds
```

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

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```
1   frase = "El qUe No arRiesGa, nO gANa."
2   frase_minus = frase.upper()
3   print(frase_minus)
```

```
[Running] python -u "d:\Practica y testeo - Python\Testeo - TP1.py"
EL QUE NO ARRIESGA, NO GANA.

[Done] exited with code=0 in 0.099 seconds
```

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

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

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

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

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

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

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

g) $a^2 + b^2$

h) $(a+b)^2$

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

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

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

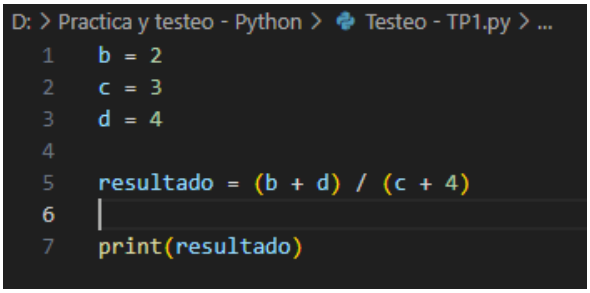
a)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   a=1
2   b=2
3   c=3
4
5   resultado = (b/2) - 4 * a * c
6
7   print(resultado)
```

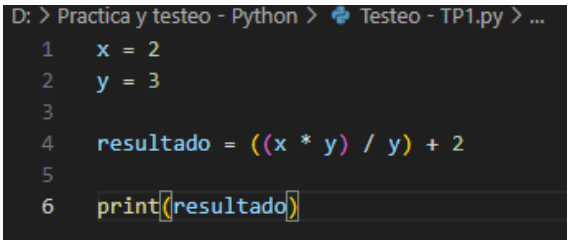
b)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   x = 2
2   y= 4
3
4   resultado = (3 * x * y) - (5 * x) + (12 * x) - 17
5
6   print(resultado)
```

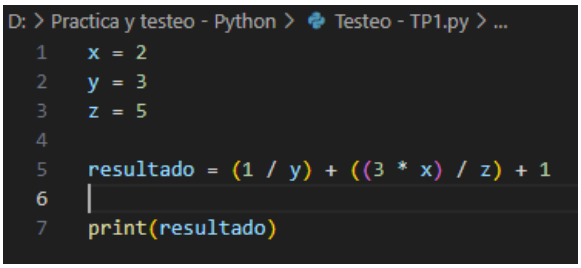
c)


D: > Practica y testeo - Python > Testeo - TP1.py > ...
b = 2
c = 3
d = 4
resultado = (b + d) / (c + 4)
print(resultado)

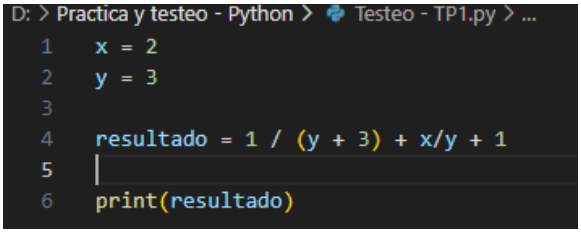
d)


D: > Practica y testeo - Python > Testeo - TP1.py > ...
x = 2
y = 3
resultado = ((x * y) / y) + 2
print(resultado)

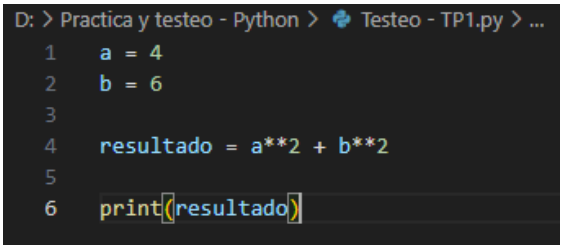
e)


D: > Practica y testeo - Python > Testeo - TP1.py > ...
x = 2
y = 3
z = 5
resultado = (1 / y) + ((3 * x) / z) + 1
print(resultado)

f)


D: > Practica y testeo - Python > Testeo - TP1.py > ...
x = 2
y = 3
resultado = 1 / (y + 3) + x/y + 1
print(resultado)

g)


D: > Practica y testeo - Python > Testeo - TP1.py > ...
a = 4
b = 6
resultado = a**2 + b**2
print(resultado)

h)

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
a = 4
b = 6

resultado = (a + b)**2

print(resultado)
```

i)

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
b= 5

resultado = (b ** 1/3) + 34

print(resultado)
```

j)

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
import math

x = 2
y = 4
z = 5
w = 12

resultado = (x / y) * ( z + w ) * math.pi

print(resultado)
```

k)

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
x = 2
y = 4
w = 12
u = 8
b = 10

resultado = (x + (y / w)) / (u + w / b)

print(resultado)
```

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

a) x=(-b+(b**2-4*a*c)**(1/2))/(2*a)

b) (x**2+y**2)/(z**2)

c) 4*x**2-2*x+7

d) (b**2)**(1/2)-4*a*c

e) (a-b)**2+(c-d)**3

f) (x+y)/y-(3*x)/5

g) (a**2+b**2)**(1/3)=c

h) 3*x**2/(3*x**3/(4*y+6))**(1/2)

a)

$$x = \frac{-b + \sqrt{(b^2 - 4 * a * c}}{2 * a}$$

b)

$$\frac{x^2 + y^2}{z^2}$$

c)

$$4 * x^2 - 2 * x + 7$$

d)

$$\sqrt{b^2} - 4 * a * c$$

e)

$$(a - b)^2 + (c - d)^2$$

f)

$$\frac{\frac{x + y}{y - 3 * x}}{5}$$

g)

$$\sqrt[3]{a^2 + b^2} = c$$

h)

$$\frac{3 * x^2}{\sqrt{\frac{3 * x^3}{4 * y + 6}}}$$

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.

Se obtiene 10,625

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   a = 5
2   b = 2
3   c = 6
4   x = -6
5   y = 4
6
7   resultado = a + b *(5 - c/2) + (7 - x)/(y+4)
8
9   print(resultado)
```

```
PS C:\Users\Silva> python -u
py"
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.

a)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   suma = 5 + 3
```

b)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   promedio = (4 + 7 + 9) / 3
```

c)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   base = 8
2   altura = 5
3
4   area = base * altura
```

d)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   num = int(input("Ingrese un numero:"))
2
3   if num % 2 == 0:
4       print("Es par")
5   else:
6       print("No es par")
```

e)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   el_doble = 16 * 2
```

f)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   diferencia = 8 - 3
2   diferencia_seis_veces = diferencia * 6
```

g)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   producto = 2 * 6
2   suma = 4 + 3
3   diferencia = producto - suma
```

h)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   num = int(input("Ingrese un numero entero:"))
2
3   es_multiplo = num % 2 == 0 and num % 3 == 0
4
5   print(type(es_multiplo))
6
7   if es_multiplo:
8       print("El numero es multiplo de 2 y 3")
9   else:
10      print("El numero no es multiplo de ambos numeros")
```

i)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
precio = int(input("Ingrese el precio:"))

es_mayor_a = precio >= 15
es_menor_a = precio <= 90

if es_menor_a and es_mayor_a:
    print("El precio es menor a 90 y mayor a 15")
else:
    print("El numero es mayor a 90 y/o menor que 15")
```

j)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
variable_n = int(input("Ingrese el valor de N:"))

variable_incrementada = variable_n + 12
```

k)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
variable_n = int(input("Ingrese el valor de N:"))

variable_disminuida = variable_n - 5
```

l)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
variable_n = int(input("Ingrese el valor de N:"))

variable_triplicada = variable_n * 3
```

m)

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
variable_n = int(input("Ingrese el valor de N:"))

variable_mitad = variable_n / 2
```

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

a) not true

b) not(1+2 != 3)

c) x = (len('jugar') > 5) and (len('jugar') < 10)

d) 'alto'[2] == 't' and x

e) 842913%10 != 3 and len('café') == 3

f) 0 != 0 or 'a' < 'y'

g) True or int('50') >= 50

h) edad = 20

not(x) or edad%2 == 0

i) es_cliente = False

not(es_cliente and not(edad < 18))

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)

En los 4 casos se obtiene un error de sintaxis, pues las operaciones abreviadas no pueden usarse dentro de la función print(). Debe realizarse de forma separada:

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```python
x = 5

x += 1
print(x)

x -= 2
print(x)

x *= 5
print(x)

x /= 5
print(x)
```

```
PS C:\Users\Silva> python
P1.py"
6
4
20
4.0
```

Sabías que en Python no existen los operadores de incremento y decremento, por lo que si queremos aumentar en 1 una variable usamos el operador +=1. En caso de querer disminuir en 1 una variable usamos el operador -=1. Este operador es válido para sumar o restar cualquier valor.

Podemos aplicar esta misma lógica con la multiplicación y la división usando los operadores *= y /= respectivamente.

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"]
```

En la posición 3 se encuentra "verde", corresponde al índice [2], pues el conteo comienza en 0

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   colores = ["rojo", "azul", "verde", "amarillo", "marron", "lila", "negro", "rosa"]
2
3   tercera_posicion = colores[2]
4
5   print(tercera_posicion)
```

```
PS C:\Users\Silva> python -u
verde
```

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

"rojo" se encuentra en la posición 0, "rosa" en la posición 7

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   colores = ["rojo", "azul", "verde", "amarillo", "marron", "lila", "negro", "rosa"]
2
3   posicion_rojo = colores.index("rojo")
4   posicion_rosa = colores.index("rosa")
5
6   print(posicion_rojo)
7
8   print(posicion_rosa)
```

```
PS C:\Users\Silva> python -u
0
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.

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   lista_numeros = ["tres", "dos", "cinco", "cuatro", "uno"]
2
3   print(lista_numeros)
```

```
PS C:\Users\Silva> python -u "d:\Practica
P1.py"
['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')
```

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   colores = ("rojo", "azul", "verde", "amarillo", "marron", "lila", "negro", "rosa", "blanco", "naranja")
2   segunda_posicion = colores [1]
3
4   print(segunda_posicion)
```

```
PS C:\Users\Silva> python
azul
```

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)
```

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   numeros = (10, 1, 5, 11)
2
3   operacion = numeros[0] + numeros[3] + numeros[2] - numeros[1]
4
5   print(operacion)
```

```
PS C:\Users\Silva> python
P1.py"
25
```

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}**

f) Cuenta la cantidad de elementos del siguiente diccionario.

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

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   diccionario = {"a":1, "b":2, "c":3, "d":4}
2
3   numero_elementos = len(diccionario)
4
5   print(numero_elementos)
```

```
PS C:\Users\Silva> python -u
P1.py"
4
```

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

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   diccionario = {"a":1, "b":2, "c":3, "d":4}
2
3   valor_c = diccionario["c"]
4
5   print(valor_c)
```

```
PS C:\Users\Silva> python -u
P1.py"
3
```

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.

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```
1  numero1 = float(input("Ingresa el primer número: "))
2  numero2 = float(input("Ingresa el segundo número: "))
3
4  suma = numero1 + numero2
5
6  print(f"La suma de {numero1} y {numero2} es igual a {suma}")
```

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

D: > Practica y testeo - Python > Testeo - TP1.py > ...

```
1  edad = int(input("Ingresa tu edad actual: "))
2
3  anios_restantes = 100 - edad
4
5  print(f"Faltan {anios_restantes} años para que cumplas 100 años.")
```

17. Operadores ternarios.

Los operadores ternarios son más conocidos en Python como expresiones condicionales. Estos operadores evalúan si una expresión es verdadera o no.

Estructura: `condition_if_true if condition else condition_if_false`

Un ejemplo:

```
es_bonito = True
estado = "Es bonito" if es_bonito else "No es bonito"
```

¡Practiquemos! Crear las variables necesarias para realizar la ejercitación.

a) Comprobar si un número es par o impar.

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   import random
2
3   numero_aleatorio = random.randint(1, 100000)
4
5   if numero_aleatorio % 2 == 0:
6       print(f"El número {numero_aleatorio} es par.")
7   else:
8       print(f"El número {numero_aleatorio} es impar.")
```

b) Obtener el valor absoluto de un número.

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   import random
2
3   numero_aleatorio = random.randint(-100000, 100000)
4
5
6   valor_absoluto = abs(numero_aleatorio)
7   print(f"El valor absoluto de {numero_aleatorio} es {valor_absoluto}")
```

c) Comparar dos números y obtener el mayor.

```
D: > Practica y testeo - Python > Testeo - TP1.py > ...
1   import random
2
3   numero1 = random.randint(1, 100)
4   numero2 = random.randint(1, 100)
5
6   print(f"Numero 1: {numero1}")
7   print(f"Numero 2: {numero2}")
8
9   if numero1 > numero2:
10      mayor = numero1
11  elif numero2 > numero1:
12      mayor = numero2
13  else:
14      mayor = None
15
16  if mayor is not None:
17      print(f"El número mayor es: {mayor}")
18  else:
19      print("Los números son iguales.")
```