

# Planteamiento Ejercicios

## Variables y Operadores

1. **Ejercicio:** Crea una variable llamada `nombre` y asígnale tu nombre como valor. Luego, imprime la variable.
2. **Ejercicio:** Crea dos variables, `a` y `b`, y asígnales los valores 5 y 10 respectivamente. Luego, imprime la suma de `a` y `b`.
3. **Ejercicio:** Calcula el área de un triángulo con base 10 y altura 5.
4. **Ejercicio:** Calcula el resto de dividir 17 entre 3.

## Condicionales

1. **Ejercicio:** Dado un número, imprime si es positivo o negativo.
2. **Ejercicio:** Dado un número, imprime si es par o impar.
3. **Ejercicio:** Dado tres números, encuentra y muestra el mayor de ellos.

## Bucles

1. **Ejercicio:** Imprime los números del 1 al 10 usando un bucle `for`.
2. **Ejercicio:** Imprime los números pares del 1 al 20 usando un bucle `while`.
3. **Ejercicio:** Usa un bucle para calcular la suma de los números del 1 al 100.

## Funciones

1. **Ejercicio:** Define una función que tome dos números y retorne su suma.
2. **Ejercicio:** Define una función que tome un número y retorne su factorial.
3. **Ejercicio:** Define una función que tome un número y determine si es primo.
4. **Ejercicio:** Define una función que reciba una lista de números y retorne la suma de ellos.
5. **Ejercicio:** Define una función que reciba una cadena de texto y retorne la cadena en reversa.

## Bucles y Funciones

## Ejercicios nivel básico

1. Crea una función para verificar si un número es par o impar y devuelva “El número es par” o “El número es impar” según corresponda.

```
pythonCopy code
def es_par(numero): return numero % 2 == 0
num = int(input("Ingresa un número: "))
if es_par(num): print("Es un número par.") else: print("Es un número impar.")
```

1. Crea una función a la que pases un número como argumento, calcule el factorial de ese número y haga print del resultado.

```
pythonCopy code
def factorial(numero): resultado = 1
for i in range(1, numero+1): resultado *= i
return resultado
num = int(input("Ingresa un número: "))
print("El factorial de", num, "es:", factorial(num))
```

1. Crea una función a la que se le pase un número como argumento, calcule la cantidad de dígitos y haga print de “La cantidad de dígitos es:” y el resultado total de dígitos.

PISTA: Para convertir un número a string usa el método `str()`. Te recordamos que para saber la longitud de una cadena utilizamos `len()`

```
pythonCopy code
def contar_digitos(numero): return len(str(numero))
num = int(input("Ingresa un número: "))
print("La cantidad de dígitos es:", contar_digitos(num))
```

1. Dada una lista de números, crea una función que devuelva el número máximo de la lista.

```
pythonCopy code
def encontrar_maximo(lista): maximo = lista[0]
for numero in lista: if numero > maximo: maximo = numero
return maximo
numeros = [5, 12, 3, 8, 9]
print("El número máximo es:", encontrar_maximo(numeros))
```

1. Crea una función que, dado un número, sume los dígitos de ese número y devuelva el resultado.

```
pythonCopy code
def sumar_digitos(numero): suma = 0
while numero > 0: suma += numero % 10
numero //= 10
return suma
```

```
0return suma num = int(input("Ingresa un número: ")) print("La suma de los dígitos es:", sumar_digitos(num))
```

1. Dados dos números, crea una función para encontrar el mínimo común múltiplo (MCM) de los dos números, que se les pasarán como argumento a la función, y devuelva el MCM.

```
pythonCopy code
def mcm(a, b): if a == 0 or b == 0: return 0 else: maximo = max(a, b) while True: if maximo % a == 0 and maximo % b == 0: return maximo maximo += 1 num1 = int(input("Ingresa el primer número: ")) num2 = int(input("Ingresa el segundo número: ")) print("El MCM es:", mcm(num1, num2))
```

1. Crea una función a la que, pasándole la base y la altura, calcule y devuelva el área de un triángulo.

```
pythonCopy code
def calcular_area_triangulo(base, altura): return (base * altura) / 2 base = float(input("Ingresa la base del triángulo: ")) altura = float(input("Ingresa la altura del triángulo: ")) print("El área del triángulo es:", calcular_area_triangulo(base, altura))
```

1. Crea una función que, dado un número, verifique si un número es positivo, negativo o cero.

```
pythonCopy code
def verificar_signo(num): if num > 0: return "positivo" elif num < 0: return "negativo" else: return "cero" num = float(input("Ingresa un número: ")) print("El número es:", verificar_signo(num))
```

1. Crea una función que, dada una palabra, cuente la cantidad de letras en una palabra.

```
pythonCopy code
def contar_letras(palabra): contador = 0 for letra in palabra: if letra.isalpha(): contador += 1 return contador palabra = input("Ingresa una palabra: ") print("La cantidad de letras es:", contar_letras(palabra))
```

1. Crea una función que, dada una lista de números, convierta la lista de números a su valor absoluto.

```
pythonCopy code
def valor_absoluto(lista): for i in range(len(lista)): lista[i] = abs(lista[i]) return
lista numeros = [5, -12, 3, -8, 9] print("Lista con valores absolutos:", valor_absolut
o(numeros))
```

1. Crea una función que, dado un número, verifique si un número es primo.

```
pythonCopy code
def es_primo(numero): if numero <= 1: return False for i in range(2, numero): if numer
o % i == 0: return False return True num = int(input("Ingresa un número: ")) if es_prim
o(num): print("Es un número primo.") else: print("No es un número primo.")
```

1. Dados dos números, crea una función para encontrar el máximo común divisor (MCD) de esos dos números.

```
pythonCopy code
def mcd(a, b): while b: a, b = b, a % b return a num1 = int(input("Ingresa el primer n
úmero: ")) num2 = int(input("Ingresa el segundo número: ")) print("El MCD es:", mcd(nu
m1, num2))
```

## Ejercicios nivel medio

1. **Ejercicio:** Define una función que utilice un bucle para imprimir los primeros n números de la serie de Fibonacci.
2. **Ejercicio:** Define una función que tome un número y retorne una lista de sus divisores.
3. **Ejercicio:** Define una función que tome una lista y retorne una nueva lista con los elementos únicos de la lista original.
4. **Ejercicio:** Define una función que tome un número y retorne la suma de sus dígitos.
5. **Ejercicio:** Define una función que tome una cadena y cuente el número de vocales en la cadena.
6. **Ejercicio:** Define una función que tome una lista y un número n, y retorne los primeros n elementos de la lista.
7. **Ejercicio:** Define una función que tome una cadena y retorne la cantidad de letras mayúsculas y minúsculas en la cadena.

8. **Ejercicio:** Define una función que tome un número y retorne True si es un número perfecto, False en caso contrario. Un número perfecto es aquel que es igual a la suma de sus divisores propios positivos. Por ejemplo, 6 es un número perfecto porque sus divisores son 1, 2 y 3, y  $6 = 1 + 2 + 3$ .
9. **Ejercicio:** Define una función que reciba un número y retorne su representación en binario.
10. **Ejercicio:** Define una función que reciba dos listas y retorne la intersección de ambas (los elementos que están en las dos listas).
11. **Ejercicio:** Define una función que tome una cadena y determine si es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).
12. **Ejercicio:** Escribe un programa que imprima los números del 1 al 50, pero para múltiplos de tres imprima "Fizz" en lugar del número y para los múltiplos de cinco imprima "Buzz". Para números que son múltiplos de tanto tres como cinco imprima "FizzBuzz".
13. **Ejercicio:** Define una función que tome una lista y retorne la lista ordenada en orden ascendente.
14. **Ejercicio:** Define una función que reciba una lista de palabras y un entero n, y retorne la lista de palabras que son más largas que n.
15. **Ejercicio:** Define una función que tome un número y calcule su serie de Fibonacci.
16. **Ejercicio:** Define una función que tome una lista de números y retorne el número más grande de la lista.
17. **Ejercicio:** Define una función que reciba un número y retorne la suma de sus dígitos al cubo.
18. **Ejercicio:** Define una función que reciba una lista de números y retorne el segundo número más grande de la lista.
19. **Ejercicio:** Define una función que tome dos listas y retorne True si tienen al menos un miembro en común, de lo contrario, retorne False.
20. **Ejercicio:** Define una función que tome una lista y retorne una nueva lista con los elementos de la lista original en orden inverso.
21. **Ejercicio:** Define una función que reciba una cadena y cuente el número de dígitos y letras que contiene.

22. **Ejercicio:** Define una función que reciba una lista de números y retorne la suma acumulada de los números
23. **Ejercicio:** Define una función que encuentre el elemento más común en una lista.
24. **Ejercicio:** Define una función que tome un número y retorne un diccionario con la tabla de multiplicar de ese número del 1 al 10.
25. **Ejercicio:** Define una función que tome una cadena y retorne un diccionario con la cantidad de apariciones de cada caracter en la cadena.
26. **Ejercicio:** Define una función que tome dos listas y retorne la lista de elementos que no están en ambas listas.
27. **Ejercicio:** Define una función que tome una lista y retorne la lista sin duplicados.
28. **Ejercicio:** Define una función que reciba un número entero positivo y retorne la suma de los cuadrados de todos los números pares menores o iguales a ese número.
29. **Ejercicio:** Define una función que reciba una lista de números y retorne el promedio de los números en la lista.
30. **Ejercicio:** Define una función que reciba una lista de cadenas y retorne la cadena más larga en la lista.
31. **Ejercicio:** Define una función que reciba un número entero  $n$  y retorne una lista con los  $n$  primeros números primos.
32. **Ejercicio:** Define una función que reciba una cadena y retorne la misma cadena pero con las palabras en orden inverso.
33. **Ejercicio:** Escribe una función que reciba una lista de tuplas y retorne una lista ordenada basada en el último elemento de cada tupla.
34. **Ejercicio:** Define una función que reciba una cadena y retorne la cantidad de letras vocales en la cadena.
35. **Ejercicio:** Define una función que reciba un número entero y retorne True si es un número primo, de lo contrario retorne False.