

10 Ejercicios de Python Condicionales PDF - DIGITAL NEST

Ejercicio 1: ¿Es un número positivo, negativo o cero?

En este primer ejercicio, aprenderemos a utilizar una declaración condicional simple para determinar si un número ingresado por el usuario es positivo, negativo o cero. Veamos el código a continuación:

```
numero = float(input("Ingresa un número: "))
```

```
if numero > 0:  
    print("El número es positivo.")  
elif numero < 0:  
    print("El número es negativo.")  
else:  
    print("El número es cero.")
```

En este ejemplo, utilizamos la función `input()` para solicitar al usuario que ingrese un número. Luego, comparamos el número ingresado con cero utilizando una declaración `if`, `elif` y `else` para determinar si es positivo, negativo o cero.

Ejercicio 2: ¿Es un número par o impar?

El segundo ejercicio se centra en determinar si un número ingresado por el usuario es par o impar. A continuación, se muestra el código para este ejercicio:

```
numero = int(input("Ingresa un número: "))
```

```
if numero % 2 == 0:  
    print("El número es par.")  
else:  
    print("El número es impar.")
```

En este caso, utilizamos el operador de módulo `%` para verificar si el número ingresado es divisible por 2 sin dejar residuo. Si es así, se considera un número par; de lo contrario, se considera un número impar.

Ejercicio 3: ¿Está dentro de un rango específico?

El siguiente ejercicio consiste en determinar si un número ingresado por el usuario se encuentra dentro de un rango específico. A continuación, se muestra el código correspondiente:

```
numero = float(input("Ingresa un número: "))
```

```
if numero >= 10 and numero <= 20:  
    print("El número está dentro del rango de 10 a 20.")
```

else:

```
print("El número está fuera del rango de 10 a 20.")
```

En este ejemplo, utilizamos los operadores de comparación \geq y \leq para verificar si el número ingresado está dentro del rango de 10 a 20.

Ejercicio 4: ¿Es una vocal o una consonante?

En este ejercicio, aprenderemos a determinar si una letra ingresada por el usuario es una vocal o una consonante. Veamos el código a continuación:

```
letra = input("Ingresa una letra: ")
```

```
if letra.lower() in ['a', 'e', 'i', 'o', 'u']:
```

```
    print("La letra es una vocal.")
```

```
else:
```

```
    print("La letra es una consonante.")
```

En este caso, convertimos la letra ingresada a minúscula utilizando el método `lower()` para asegurarnos de que la comparación sea insensible a mayúsculas y minúsculas. Luego, verificamos si la letra se encuentra en la lista de vocales.

Ejercicio 5: ¿Es un triángulo válido?

El quinto ejercicio trata de determinar si las longitudes ingresadas por el usuario pueden formar los lados de un triángulo válido. Aquí tienes el código correspondiente:

```
lado1 = float(input("Ingresa la longitud del primer lado: "))
```

```
lado2 = float(input("Ingresa la longitud del segundo lado: "))
```

```
lado3 = float(input("Ingresa la longitud del tercer lado: "))
```

```
if lado1 + lado2 > lado3 and lado2 + lado3 > lado1 and lado3 + lado1 > lado2:
```

```
    print("Las longitudes ingresadas pueden formar un triángulo válido.")
```

```
else:
```

```
    print("Las longitudes ingresadas no pueden formar un triángulo válido.")
```

En este ejemplo, utilizamos las propiedades de los triángulos para determinar si las longitudes ingresadas cumplen con la desigualdad triangular.

Ejercicio 6: ¿Es un año bisiesto?

En este ejercicio, aprenderemos a determinar si un año ingresado por el usuario es un año bisiesto. A continuación, se muestra el código correspondiente:

```
anio = int(input("Ingresa un año: "))
```

```
if (anio % 4 == 0 and anio % 100 != 0) or anio % 400 == 0:
```

```
    print("El año ingresado es un año bisiesto.")
```

```
else:
```

```
print("El año ingresado no es un año bisiesto.")
```

En este caso, utilizamos una combinación de operadores de módulo y condiciones para verificar si el año ingresado es divisible por 4 pero no por 100, o si es divisible por 400.

Ejercicio 7: ¿Es un número primo?

El siguiente ejercicio trata de determinar si un número ingresado por el usuario es un [número primo](#). A continuación, se muestra el código correspondiente:

```
numero = int(input("Ingresa un número: "))

if numero > 1:
    for i in range(2, numero):
        if (numero % i) == 0:
            print("El número ingresado no es un número primo.")
            break
    else:
        print("El número ingresado es un número primo.")
else:
    print("El número ingresado no es un número primo.")
```

En este ejemplo, utilizamos un bucle for para iterar desde 2 hasta el número ingresado, y verificamos si hay algún divisor que divide al número sin dejar residuo. Si encontramos un divisor, sabemos que el número no es primo. De lo contrario, si el bucle completa todas las iteraciones sin encontrar ningún divisor, el número se considera primo.

Ejercicio 8: ¿Es una palabra palíndroma?

En este ejercicio, aprenderemos a determinar si una palabra ingresada por el usuario es [palíndroma](#), es decir, si se lee igual de izquierda a derecha y de derecha a izquierda. A continuación, se muestra el código correspondiente:

```
palabra = input("Ingresa una palabra: ")

if palabra.lower() == palabra.lower()[::-1]:
    print("La palabra ingresada es palíndroma.")
else:
    print("La palabra ingresada no es palíndroma.")
```

En este caso, convertimos la palabra ingresada a minúsculas utilizando el método lower() para asegurarnos de que la comparación sea insensible a mayúsculas y minúsculas. Luego, utilizamos la técnica de rebanado de cadenas [::-1] para invertir la palabra y comparamos si es igual a la palabra original.

Ejercicio 9: ¿Es una contraseña segura?

El siguiente ejercicio trata de determinar si una contraseña ingresada por el usuario cumple con ciertos requisitos de seguridad. A continuación, se muestra el código correspondiente:

```
contrasena = input("Ingresa una contraseña: ")

longitud_suficiente = len(contrasena) >= 8
tiene_mayusculas = any(c.isupper() for c in contrasena)
tiene_minusculas = any(c.islower() for c in contrasena)
tiene_digitos = any(c.isdigit() for c in contrasena)
tiene_caracteres_especiales = any(c in "!@#$%^&*()-_+[]\|;:\",<.>/?`~" for c in
contrasena)

if longitud_suficiente and tiene_mayusculas and tiene_minusculas and tiene_digitos
and tiene_caracteres_especiales:
    print("La contraseña ingresada es segura.")
else:
    print("La contraseña ingresada no cumple con los requisitos de seguridad.")
```

En este ejemplo, verificamos varias condiciones para determinar si la contraseña cumple con los requisitos de seguridad. Estas condiciones incluyen una longitud mínima de 8 caracteres, al menos una letra mayúscula, al menos una letra minúscula, al menos un dígito y al menos un carácter especial.

Ejercicio 10: ¿Qué día de la semana es?

El último ejercicio trata de determinar qué día de la semana corresponde a una fecha ingresada por el usuario. A continuación, se muestra el código correspondiente:

```
import datetime

fecha = input("Ingresa una fecha (formato: DD/MM/AAAA): ")

try:
    dia, mes, anio = map(int, fecha.split('/'))
    fecha = datetime.date(anio, mes, dia)
    dia_semana = fecha.strftime("%A")
    print("El día de la semana correspondiente a la fecha ingresada es:", dia_semana)
except:
    print("La fecha ingresada no es válida.")
```

En este ejemplo, utilizamos el módulo `datetime` para trabajar con fechas. Primero, convertimos la fecha ingresada por el usuario en un objeto de fecha utilizando el método `datetime.date()`. Luego, utilizamos el método `strftime("%A")` para obtener el nombre del día de la semana correspondiente a esa fecha.

ENCUENTRA MAS EJERCICIOS DE PYTHON EN [ESTE ENLACE](#)