🧠💡 Kata: Estructuras Condicionales

📚 Descripción General

¡Bienvenido a esta kata de programación! 🎯
En esta práctica, desarrollarás tu habilidad para implementar **estructuras condicionales** en Python 🐍, aprendiendo a:

- Tomar decisiones en el flujo del programa usando if, elif y else.
- Validar datos de entrada con condiciones lógicas.
- Mejorar la interacción con el usuario mediante mensajes personalizados.

Cada ejercicio incluye:

- 🎯 **Objetivo**: Qué lograrás.
- 🔍 **Instrucciones**: Pasos claros para resolverlo.
- 🧠 **Preguntas de reflexión**: Para profundizar en el aprendizaje.

🛠️ Requisitos

Antes de comenzar, asegurate de tener lo siguiente listo:

- 💻 Python instalado (versión 3.8 o superior)
- 📝 Editor de código (recomendado: VS Code)
- 🖥️ Muchas ganas de aprender y divertirte resolviendo desafíos 💪

🔐 Ejercicio 1: Validación de contraseña

🎯 **Objetivo**: Analizar un programa existente que verifica una contraseña.
🔍 **Instrucciones**:

1. Lee el siguiente código y explica qué hace:

```
contrasena_correcta = "programacion1"

contrasena_usuario = input("Introduce la contraseña: ")

if contrasena_usuario == contrasena_correcta:

  print("Contraseña correcta. Bienvenido.")

else:

  print("Contraseña incorrecta. Intenta de nuevo.")
```

🧠 **Preguntas de reflexión**:

- ¿Qué pasa si el usuario ingresa la contraseña con mayúsculas?
- ¿Cómo mejorarías el programa para dar más intentos?

🔤 **Ejercicio 2: Identificador de vocales**

🎯 **Objetivo**: Clasificar caracteres usando condicionales.
🔍 **Instrucciones**:

1. Solicita una letra al usuario.
2. Si es una vocal (a, e, i, o, u, en mayúscula o minúscula), imprime: *"La letra ingresada es una vocal"*.
3. En otro caso, imprime: *"La letra ingresada no es una vocal"*.

🧠 **Preguntas de reflexión**:

- ¿Cómo manejarías vocales acentuadas (á, é)?
- ¿Qué estructura usarías para simplificar las comparaciones?

➕ **Ejercicio 3: Clasificador de números**

🎯 **Objetivo**: Determinar el signo de un número.
🔍 **Instrucciones**:

1. Pide un número al usuario.
2. Si es **positivo**, imprime: *"El número es positivo"*.
3. Si es **negativo**, imprime: *"El número es negativo"*.
4. Si es **cero**, imprime: *"El número es cero"*.

🧠 **Preguntas de reflexión**:

- ¿Qué ocurre si el usuario ingresa un texto?
- ¿Cómo adaptarías el código para números decimales?

⚖️ **Ejercicio 4: Comparador de números**

🎯 **Objetivo**: Comparar dos números con condicionales.
🔍 **Instrucciones**:

1. Solicita dos números al usuario.
2. Si el **primero es mayor**, imprime: *"El primer número ingresado es mayor"*.
3. Si el **primero es menor**, imprime: *"El primer número ingresado es menor"*.

4. Si **son iguales**, imprime: *"Los números ingresados son iguales"*.

🧠 **Preguntas de reflexión**:

- ¿Cómo modificarías el programa para comparar más de dos números?
- ¿Qué pasa si se ingresan valores no numéricos?

🌡️ **Ejercicio 5: Clima según temperatura**

🎯 **Objetivo**: Clasificar temperaturas en rangos.
🔍 **Instrucciones**:

1. Pide la temperatura actual en °C.
2. Si es **≤ 10°C**, imprime: *"Hace frío"*.
3. Si está **entre 10°C y 25°C**, imprime: *"Está templado"*.
4. Si es **> 25°C**, imprime: *"Hace calor"*.

🧠 **Preguntas de reflexión**:

- ¿Cómo adaptarías el programa para usar °F?
- ¿Qué considerarías para añadir más rangos (ej: "Hace mucho frío")?

📅 **Ejercicio 6: Detector de años bisiestos**

🎯 **Objetivo**: Aplicar condiciones compuestas.
🔍 **Instrucciones**:

1. Pide un año al usuario.
2. Si es **divisible por 4 pero no por 100**, o **divisible por 400**, imprime: *"Se ingresó un año bisiesto"*.
3. En otro caso, imprime: *"Se ingresó un año no bisiesto"*.

🧠 **Preguntas de reflexión**:

- ¿Por qué el año 1900 no es bisiesto?
- ¿Cómo validarías que el año sea positivo?

🔤 **Ejercicio 7: Ajustador de frases**

🎯 **Objetivo**: Manipular strings con condicionales.
🔍 **Instrucciones**:

1. Pide una frase o palabra al usuario.

2. Si **no termina en punto**, añádelo al final.
3. Imprime el resultado.

🧠 **Preguntas de reflexión**:

- ¿Cómo manejarías frases que terminan con espacios?
- ¿Qué otros caracteres de puntuación podrías considerar?

🔒 **Ejercicio 8: Validador de contraseña segura**

🎯 **Objetivo**: Implementar múltiples condiciones.
🔍 **Instrucciones**:

1. Pide al usuario que cree una contraseña.
2. Verifica que cumpla:
    - **8+ caracteres** y **≤20 caracteres**.
    - Al menos **1 mayúscula** (usa .isupper()).
    - Al menos **1 número** (usa .isdigit()).
3. Si es segura, imprime: *"¡Felicitaciones! Creaste tu contraseña."*.
4. Si no, imprime: *"La contraseña no es segura."*.

🧠 **Preguntas de reflexión**:

- ¿Cómo añadirías la regla de usar un carácter especial?
- ¿Por qué es importante limitar la longitud máxima?

💡 **Ejercicio 9: Mejorando mensajes de error**

🎯 **Objetivo**: Dar retroalimentación específica al usuario.
🔍 **Instrucciones**:

1. Basado en el **Ejercicio 8**, mejora los mensajes de error:
    - Si tiene **<8 caracteres**: *"La contraseña no es segura. Debe tener al menos 8 caracteres."*.
    - Si tiene **>20 caracteres**: *"...no más de 20 caracteres."*.
    - Si **falta mayúscula**: *"...al menos una mayúscula."*.
    - Si **falta número**: *"...al menos un número."*.

🧠 **Preguntas de reflexión**:

- ¿Cómo evitarías repetir código al verificar cada condición?
- ¿Qué ventajas tiene este enfoque para el usuario?

✂️ **Ejercicio 10: Piedra, papel o tijera**

🎯 **Objetivo**: Implementar lógica de juego con condicionales anidados.

🔍 **Instrucciones**:

1. Pide al usuario las jugadas del **Jugador 1** y **Jugador 2** (piedra, papel o tijera).
2. Usa la tabla proporcionada para determinar el resultado (ganador o empate).

Resultado	**Jugador 1**	**Jugador 2**
EMPATE	Piedra	Piedra
GANA JUGADOR 2	Piedra	Papel
GANA JUGADOR 1	Piedra	Tijera
GANA JUGADOR 1	Papel	Piedra
EMPATE	Papel	Papel
GANA JUGADOR 2	Papel	Tijera
GANA JUGADOR 2	Tijera	Piedra
GANA JUGADOR 1	Tijera	Papel
EMPATE	Tijera	Tijera

3. Imprime: *"GANA JUGADOR 1"*, *"GANA JUGADOR 2"* o *"EMPATE"*.

🧠 **Preguntas de reflexión**:

- ¿Cómo manejarías entradas inválidas (ej: "piedra" mal escrito)?
- ¿Qué estructura usarías para simplificar las comparaciones?

🌟 **Bonus**:

- **Diagrama de flujo**: Dibuja el diagrama de flujo del Ejercicio 10.