UNIVERSIDAD CENTRAL DEL ECUADOR

Cuestionario de Evaluación — Condiciones en Python (Verdadero o Falso)

Estudiante: _____ **Fecha:** _____

Carrera: _____ **Paralelo:** _____

Objetivo de la Evaluación

Evaluar el nivel de comprensión conceptual y la capacidad de análisis sintáctico-lógico de los estudiantes respecto a las estructuras condicionales en Python, mediante la discriminación de enunciados teóricos y prácticos bajo la modalidad de Verdadero o Falso.

Instrucciones

- Lea detenidamente el enunciado de cada reactivo y la cita académica asociada.
- Determine la veracidad o falsedad de la afirmación o del resultado del código mostrado.
- Marque con una equis (X) o un clic el recuadro de **Verdadero** o **Falso** según corresponda.
- Cada respuesta correcta equivale a 2 puntos (Valoración total: 20 puntos).

Cuestionario (Verdadero o Falso)

1. Según Downey (2015), el propósito principal de una estructura condicional en un flujo de programa es permitir que la ejecución elija entre diferentes cursos de acción basándose en la evaluación de una condición.

☐ Verdadero ☐ Falso

Referencia: Downey, A. (2015). Think Python: How to Think Like a Computer Scientist.

2. De acuerdo con la documentación oficial de Python, si analizamos el código adjunto, la terminal no mostrará ninguna salida en pantalla debido a que la condición evaluada resulta falsa.

```
edad = 20

if edad >= 18:
    print("Mayor de edad")
```

☐ Verdadero ☐ Falso

Referencia: Python Documentation - if Statements

3. Según Lutz (2013), las expresiones booleanas que sirven como fundamento en las estructuras de decisión en Python pueden producir una amplia variedad de tipos de datos, incluyendo cadenas y flotantes, y no se limitan únicamente a True y False.

☐ Verdadero ☐ Falso

Referencia: Lutz, M. (2013). Learning Python (5th ed.).

LIVEWORKSHEETS

4. En el siguiente fragmento de código, para que el mensaje "Hace calor" se logre imprimir de forma efectiva, la variable `temperatura` debe cumplir obligatoriamente con la condición estricta de ser estrictamente mayor a 30 (`temperatura > 30`).

```
temperatura = 35

if temperatura > 30:
    print("Hace calor")
```

☐ Verdadero ☐ Falso

5. Van Rossum y Drake señalan el uso de operadores de comparación en Python; bajo esta sintaxis, el operador utilizado para verificar la igualdad exacta entre dos valores u objetos es el signo de un único igual (=).

☐ Verdadero ☐ Falso

Referencia: Van Rossum, G., & Drake, F. L. Python Language Reference.

6. Al evaluar el siguiente bloque condicional doble planteado bajo las directrices de Downey (2015), el programa tomará el camino del `else` e imprimirá en la consola la cadena "Reprobado".

```
nota = 8

if nota >= 7:
    print("Aprobado")
else:
    print("Reprobado")
```

☐ Verdadero ☐ Falso

Referencia: Downey, A. (2015). Think Python.

7. Según la documentación oficial de Python, la palabra reservada utilizada para ejecutar un bloque alternativo y definitivo de instrucciones cuando la condición principal del `if` (y las intermedias) resulta falsa es `else`.

☐ Verdadero ☐ Falso

Referencia: Python Documentation - Control Flow Tools

8. Basándose en las definiciones de Sweigart (2020) sobre control de flujo, la estructura sintáctica más adecuada en Python cuando se requiere evaluar secuencialmente múltiples condiciones posibles y mutuamente excluyentes es la combinación `if-elif-else`.

☐ Verdadero ☐ Falso

Referencia: Sweigart, A. (2020). Automate the Boring Stuff with Python.

9. Lutz (2013) explica las conjunciones lógicas; bajo este estándar, el operador lógico and en Python requiere estrictamente que todas y cada una de las condiciones individuales evaluadas sean verdaderas (True) para que la expresión completa devuelva True.

☐ Verdadero ☐ Falso

Referencia: Lutz, M. (2013). Learning Python.

LIVEWORKSHEETS

10. De acuerdo con la documentación oficial de Python respecto a las operaciones booleanas, el operador lógico `or` evalúa de manera inclusiva, lo que significa que devolverá True únicamente si todas las condiciones ligadas son verdaderas a la vez.

☐ Verdadero ☐ Falso

Referencia: Python Documentation - Boolean Operations

Bibliografía (APA 7.ª edición)

Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). Green Tea Press.

Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.

Sweigart, A. (2020). *Automate the Boring Stuff with Python* (2nd ed.). No Starch Press.

Van Rossum, G., & Drake, F. L. *Python Language Reference*. Python Software Foundation.

LIVEWORKSHEETS