UNIVERSIDAD CENTRAL DEL ECUADOR

Cuestionario de Evaluación 1 — Introducción y Fundamentos de Python

Estudiante: _____ **Fecha:** _____

Carrera: _____ **Paralelo:** _____

Objetivo de la Evaluación

Evaluar el conocimiento inicial de los estudiantes sobre la sintaxis básica, variables, tipos de datos fundamentales y la filosofía de diseño del lenguaje de programación Python mediante el análisis conceptual y de fragmentos de código.

Instrucciones

- Lea cuidadosamente cada pregunta antes de responder.
- Seleccione una única respuesta correcta marcando el casillero correspondiente con una equis (X) o un clic.
- Cada reactivo respondido correctamente equivale a 2 puntos.
- La evaluación tiene una valoración total de 20 puntos.

Cuestionario

1. Según Downey (2015), Python es un lenguaje de alto nivel. ¿Cuál es una de las principales ventajas de los lenguajes de alto nivel en comparación con los de bajo nivel?

- ☐ **A)** Son más difíciles de programar y entender.
- ☐ **B)** Tardan menos tiempo en escribirse y son más legibles.
- ☐ **C)** Se ejecutan directamente en la CPU sin traducción.
- ☐ **D)** Solo funcionan en sistemas operativos antiguos.

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

2. De acuerdo con la filosofía de diseño de Python plasmada en "The Zen of Python" por Tim Peters, ¿cuál es un principio fundamental del lenguaje?

- ☐ **A)** Lo complejo es mejor que lo complicado.
- ☐ **B)** Bello es mejor que feo.
- ☐ **C)** La legibilidad no importa si el código funciona.
- ☐ **D)** Debe haber infinitas formas de hacer lo mismo.

Referencia: Peters, T. (2004). The Zen of Python (PEP 20).

LIVEWORKSHEETS

3. Analice el siguiente fragmento de código:

```
x = "10"
y = 5
print(type(x))
```

¿Qué tipo de datos devolverá la función `type(x)`?

- ☐ **A)** `<class 'int'>`
- ☐ **B)** `<class 'float'>`
- ☐ **C)** `<class 'str'>`
- ☐ **D)** `<class 'bool'>`

4. Según Lutz (2013), una característica clave de Python es que las variables se crean dinámicamente cuando se les asigna un valor por primera vez. ¿Cuál de los siguientes es un nombre de variable válido en Python?

- ☐ **A)** `2_nombre`
- ☐ **B)** `mi-variable`
- ☐ **C)** `_nombre_estudiante`
- ☐ **D)** `class`

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

5. Observe el siguiente código de asignación múltiple:

```
a, b = 3, 7
a = b
print(a)
```

¿Cuál será el valor impreso en pantalla tras ejecutar este bloque?

- ☐ **A)** 3
- ☐ **B)** 7
- ☐ **C)** 10
- ☐ **D)** Error de sintaxis.

6. Según Sweigart (2020), la concatenación es la unión de dos cadenas de texto usando el operador aritmético de suma. ¿Cuál será el resultado del siguiente fragmento?

```
saludo = "Hola" + " " + "Mundo"
print(saludo)
```

- ☐ **A)** `HolaMundo`
- ☐ **B)** `Hola Mundo`
- ☐ **C)** `Error: No se pueden sumar textos.`
- ☐ **D)** `Hola,Mundo`

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

LIVEWORKSHEETS

7. Analice la siguiente operación matemática y determine el orden de prioridad de operadores integrado en Python:

```
resultado = 5 + 3 * 2 ** 2
print(resultado)
```

- ☐ **A)** 32
- ☐ **B)** 17
- ☐ **C)** 41
- ☐ **D)** 23

8. Según la documentación oficial de Python, la función integrada `input()` se utiliza para interactuar con el usuario. ¿Qué tipo de dato retorna siempre por defecto esta función?

- ☐ **A)** Entero (`int`)
- ☐ **B)** Booleano (`bool`)
- ☐ **C)** Cadena de texto (`str`)
- ☐ **D)** Flotante (`float`)

Referencia: Python Documentation – Built-in Functions

9. ¿Cuál de los siguientes operadores en Python se utiliza de manera específica para obtener el residuo o módulo de una división entera?

- ☐ **A)** `/`
- ☐ **B)** `//`
- ☐ **C)** `%`
- ☐ **D)** `**`

10. De acuerdo con Downey (2015), los comentarios son notas aclaratorias escritas en el programa destinadas a los desarrolladores humanos y que el intérprete ignora por completo. ¿Qué símbolo utiliza Python para iniciar un comentario de una sola línea?

- ☐ **A)** `//`
- ☐ **B)** `/*`
- ☐ **C)** `#`
- ☐ **D)** `<!--`

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

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.

Peters, T. (2004). *The Zen of Python* (PEP 20). Python Software Foundation.

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

LIVEWORKSHEETS