

Python (lenguaje de programación)

1. Que es una clase abstracta?

- a) ☐ Una clase abstracta es el nombre de cualquier clase desde la que puede instanciar un objeto.
- b) ☐ Las clases abstractas deben redefinirse cada vez que se crea una instancia de un objeto a partir de ellas.
- c) ☐ Las clases abstractas deben heredar de las clases concretas.
- d) ☐ Una clase abstracta existe solo para que otras clases "concretas" puedan heredar de la clase abstracta.

2. ¿Qué sucede cuando usa la función incorporada `any()` en una lista?

- a) ☐ La función `any()` devolverá aleatoriamente cualquier elemento de la lista.
- b) ☐ La función `any()` devuelve `True` si algún elemento de la lista se evalúa como `True`. De lo contrario, devuelve `False`.
- c) ☐ La función `any()` toma como argumentos la lista a verificar dentro y el elemento a verificar. Si "cualquiera" de los elementos de la lista coincide con el elemento a comprobar, la función devuelve `True`.
- d) ☐ La función `any()` devuelve un valor booleano que responde a la pregunta "¿Hay algún elemento en esta lista?"

ejemplo

```
if any([True, False, False, False]) == True:  
    print('Yes, there is True')  
>>> 'Yes, there is True'
```

3. ¿Qué son los atributos?

- a) ☐ Los atributos son una versión larga de una declaración `if/else`, que se usa cuando se prueba la igualdad entre objetos.
- b) ☐ Los atributos son una forma de contener datos o describir un estado para una clase o una instancia de una clase.
- c) ☐ Los atributos son cadenas que describen las características de una clase.
- d) ☐ Los argumentos de función se denominan "atributos" en el contexto de métodos de clase y métodos de instancia.

4. ¿Cuál es el término para describir este código?

recuento, fruta, precio = (2, 'manzana', 3,5)

- a) ☐ asignación de tupla
- b) ☐ desempaquetado de tuplas
- c) ☐ coincidencia de tuplas
- d) ☐ duplicación de tuplas

5. ¿Cuál es uno de los usos más comunes de la biblioteca sys de Python?

- a) ☐ para capturar los argumentos de la línea de comandos proporcionados en el tiempo de ejecución de un archivo
- b) ☐ para conectar varios sistemas, como conectar una interfaz web, un servicio API, una base de datos y una aplicación móvil
- c) ☐ para tomar una instantánea de todos los paquetes y bibliotecas en su entorno virtual
- d) ☐ para escanear la salud de su ecosistema de Python mientras está dentro de un entorno virtual

6. ¿Cuál es la forma correcta de escribir un doctest?

- a) ☐

```
def sum(a, b):  
    """  
    sum(4, 3)  
    7  
  
    sum(-4, 5)  
    1  
    """  
    return a + b
```

- b) ☐

```
def sum(a, b):  
    """  
    >>> sum(4, 3)  
    7  
  
    >>> sum(-4, 5)  
    1  
    """
```

```
return a + b
```

c) ☐

```
def sum(a, b):  
    """  
    # >>> sum(4, 3)  
    # 7  
  
    # >>> sum(-4, 5)  
    # 1  
    """  
    return a + b
```

d) ☐

```
def sum(a, b):  
    ###  
    >>> sum(4, 3)  
    7  
  
    >>> sum(-4, 5)  
    1  
    ###  
    return a + b
```

7. ¿Cuál de estas NO es una característica de las tuplas nombradas?

- a) ☐ Puede asignar un nombre a cada uno de los miembros de `namedtuple` y referirse a ellos de esa manera, de manera similar a como accedería a las claves en el `dictionary`.
- b) ☐ Cada miembro de un objeto de tupla con nombre se puede indexar directamente, como en una `tuple` normal.
- c) ☐ Las `namedtuples` son tan eficientes en memoria como las `tuple` regulares.
- d) ☐ No se necesita importar para usar `namedtuples` porque están disponibles en la biblioteca estándar.

8. ¿Qué afirmación NO describe el concepto de encapsulación de la programación orientada a objetos?

- a) ☐ Protege los datos de interferencias externas.
- b) ☐ Se encapsula una clase principal y ningún dato de la clase principal pasa a la clase secundaria.

- c) ☐ Mantiene los datos y los métodos que pueden manipular esos datos en un solo lugar.
- d) ☐ Solo permite cambiar los datos por métodos.

9. ¿Cuál es el propósito de una declaración if/else?

- a) ☐ Le dice a la computadora qué fragmento de código ejecutar si las instrucciones que codificaste son incorrectas.
- b) ☐ Ejecuta un fragmento de código si todas las importaciones fueron exitosas y otro fragmento de código si las importaciones no fueron exitosas.
- c) ☐ Ejecuta un trozo de código si una condición es verdadera, pero un trozo de código diferente si la condición es falsa.
- d) ☐ Le dice a la computadora qué fragmento de código ejecutar si hay suficiente memoria para manejarlo, y qué fragmento de código ejecutar si no hay suficiente memoria para manejarlo.

10. ¿Qué hace la función integrada map()?

- a) ☐ Crea una ruta desde múltiples valores en un iterable a un solo valor.
- b) ☐ Aplica una función a cada elemento en un iterable y devuelve el valor de esa función.
- c) ☐ Convierte un tipo de valor complejo en tipos de valor más simples.
- d) ☐ Crea un mapeo entre dos elementos diferentes de diferentes iterables.

11. ¿Cuál es el propósito de la instrucción pass en Python?

- a) ☐ Se utiliza para omitir la declaración yield de un generador y devolver un valor de Ninguno.
- b) ☐ Es una operación nula utilizada principalmente como marcador de posición en funciones, clases, etc.
- c) ☐ Se utiliza para pasar el control de un bloque de instrucciones a otro.
- d) ☐ Se utiliza para omitir el resto de un ciclo while o for y volver al inicio del ciclo.

12. ¿Cuál es el término que se usa para describir los elementos que se pueden pasar a una función?

- a) ☐ argumentos
- b) ☐ paradigmas
- c) ☐ atributos
- d) ☐ decoradores

13. ¿Cuándo deja de iterar un bucle for?

- a) ☐ cuando encuentra un bucle infinito
- b) ☐ cuando encuentra una declaración if/else que contiene una palabra clave break
- c) ☐ cuando ha evaluado cada elemento en el iterable en el que está trabajando o se encuentra una palabra clave de interrupción
- d) ☐ cuando el tiempo de ejecución del ciclo excede $O(n^2)$

14. ¿Qué sucede cuando usa la función integrada all() en una lista?

- a) ☐ La función all() devuelve un valor booleano que responde a la pregunta "¿Son iguales todos los elementos de esta lista?"
- b) ☐ La función all() devuelve True si todos los elementos de la lista se pueden convertir en cadenas. De lo contrario, devuelve False.
- c) ☐ La función all() devolverá todos los valores de la lista.
- d) ☐ La función all() devuelve True si todos los elementos de la lista se evalúan como True. De lo contrario, devuelve False.

15. Elija la representación correcta de doctest para la función en Python.

- a) ☐ :

```
def sum(a, b):  
    # a = 1  
    # b = 2  
    # sum(a, b) = 3  
  
    return a + b
```

b) ☐ :

```
def sum(a, b):  
    """  
    a = 1  
    b = 2  
    sum(a, b) = 3  
    """  
  
    return a + b
```

c) ☐ :

```
def sum(a, b):  
    """  
    >>> a = 1  
    >>> b = 2  
    >>> sum(a, b)  
    3  
    """  
  
    return a + b
```

d) ☐ :

```
def sum(a, b):  
    '''  
    a = 1  
    b = 2  
    sum(a, b) = 3  
    '''  
  
    return a + b
```

16. ¿Qué hace el método `__init__()` de una clase?

- a) ☐ Hace que las clases se reconozcan entre sí si se define más de una clase en un solo archivo de código.
- b) ☐ Se incluye para preservar la compatibilidad con versiones anteriores de Python 3 a Python 2, pero ya no necesita usarse en Python 3.
- c) ☐ Es un método que actúa como constructor y se llama automáticamente cada vez que se crea un nuevo objeto a partir de una clase. Determina el estado inicial de un objeto nuevo.
- d) ☐ Inicializa cualquier importación que haya incluido en la parte superior de su archivo.

17. ¿Qué afirmación sobre los métodos de clase es verdadera?

- a) ☐ Un método de clase es una función regular que pertenece a una clase, pero debe devolver Ninguno.
- b) ☐ Un método de clase puede modificar el estado de la clase, pero no puede modificar directamente el estado de una instancia que hereda de esa clase.
- c) ☐ Un método de clase es similar a una función regular, pero un método de clase no toma ningún argumento.
- d) ☐ Un método de clase contiene todos los datos de una clase en particular.

18. ¿Cuál es la complejidad del tiempo de ejecución de buscar un elemento en un árbol de búsqueda binaria?

- a) ☐ El tiempo de ejecución para buscar en un árbol de búsqueda binario es $O(1)$ porque cada nodo actúa como una clave, similar a un diccionario.
- b) ☐ El tiempo de ejecución para buscar en un árbol de búsqueda binario es $O(n!)$ porque cada nodo debe compararse con todos los demás nodos.
- c) ☐ El tiempo de ejecución para buscar en un árbol de búsqueda binaria es generalmente $O(h)$, donde h es la altura del árbol.
- d) ☐ El tiempo de ejecución para buscar en un árbol de búsqueda binaria es $O(n)$ porque se deben visitar todos los nodos del árbol.

19. ¿Qué es una función lambda?

- a) ☐ cualquier función que haga uso de constantes científicas o matemáticas, a menudo representadas por letras griegas en la escritura académica
- b) ☐ una función que se ejecuta cuando se usan decoradores
- c) ☐ cualquier función cuya definición esté contenida en cinco líneas de código o menos
- d) ☐ una función pequeña y anónima que puede tomar cualquier cantidad de argumentos pero solo tiene una expresión para evaluar

20. ¿Cuál es la diferencia entre atributos de clase y atributos de instancia?

- a) ☐ Los atributos de instancia se pueden cambiar, pero los atributos de clase no se pueden cambiar
- b) ☐ Los atributos de clase son compartidos por todas las instancias de la clase. Los atributos de instancia pueden ser único solo para esa instancia
- c) ☐ No hay diferencia entre atributos de clase y atributos de instancia
- d) ☐ Los atributos de clase pertenecen solo a la clase, no a la instancia de esa clase. Los atributos de instancia se comparten entre todas las instancias de una clase

21. ¿Qué opción es el ejemplo más sintácticamente correcto de la bifurcación condicional?

- a) ☐ :

```
num_people = 5

if num_people > 10:
    print("There is a lot of people in the pool.")
elif num_people > 4:
    print("There are some people in the pool.")
else:
    print("There is no one in the pool.")
```

- b) ☐ :

```
num_people = 5

if num_people > 10:
    print("There is a lot of people in the pool.")
if num_people > 4:
    print("There are some people in the pool.")
else:
    print("There is no one in the pool.")
```

- c) ☐ :

```
num_people = 5

if num_people > 10;
    print("There is a lot of people in the pool.")
elif num_people > 4;
    print("There are some people in the pool.")
else;
    print("There is no one in the pool.")
```


d) ☐ :

```
if num_people > 10;
    print("There is a lot of people in the pool.")
if num_people > 4;
    print("There are some people in the pool.")
else;
    print("There is no one in the pool.")
```

22. ¿Cómo funciona defaultdict?

- a) ☐ defaultdict creará automáticamente un diccionario para usted que tiene claves que son los números enteros 0-10.
- b) ☐ defaultdict obliga a un diccionario a aceptar solo claves que sean de los tipos especificados cuando creó el defaultdict (como cadenas o números enteros).
- c) ☐ Si intenta leer desde un defaultdict con una clave inexistente, se creará un nuevo par clave-valor predeterminado para usted en lugar de generar un KeyError.
- d) ☐ defaultdict almacena una copia de un diccionario en la memoria que puede usar de forma predeterminada si el original se modifica involuntariamente.

23. ¿Qué devolverá este fragmento de código?

```
import numpy as np
np.ones([1,2,3,4,5])
```

- a) ☐ Devuelve una matriz de 5x5; cada fila tendrá los valores 1,2,3,4,5.
- b) ☐ Devuelve un array con los valores 1,2,3,4,5
- c) ☐ Devuelve cinco matrices cuadradas diferentes llenas de unos. El primero es 1x1, el segundo 2x2, y así sucesivamente hasta 5x5
- d) ☐ Devuelve una matriz de 5 dimensiones de tamaño 1x2x3x4x5 llena de 1s.

24. ¿Cuál es la sintaxis correcta para llamar a un método de instancia en una clase llamada Juego?

- a) ☐ my_game = Game(self) self.my_game.roll_dice()
- b) ☐ my_game = Game() self.my_game.roll_dice()
- c) ☐ my_game = Game() my_game.roll_dice()
- d) ☐ my_game = Game(self) my_game.roll_dice(self)

25. ¿Qué comando creará una lista del 10 al 1? Ejemplo:

[10,9,8,7,6,5,4,3,2,1]

- a) ☐ reversed(list(range(1,11)))
- b) ☐ list(reversed(range(1,10)))
- c) ☐ list(range(10,1,-1))
- d) ☐ list(reversed(range(1,11)))

26. En Python, ¿cómo puede el compilador identificar el bloque interno de un bucle for?

- a) ☐ debido al nivel de sangría después del ciclo for
- b) ☐ debido a la palabra clave end al final del bucle for
- c) ☐ porque el bloque está entre corchetes ({})
- d) ☐ debido al espacio en blanco al final del cuerpo del bucle for

27. Suponga que tiene una variable llamada vector de tipo np.array con 10.000 elementos. ¿Cómo puedes convertir vector en una variable llamada matriz con dimensiones 100x100?

- a) ☐ matrix = (vector.shape = (100,100))
- b) ☐ matrix = vector.to_matrix(100,100)
- c) ☐ matrix = matrix(vector,100,100)
- d) ☐ matrix = vector.reshape(100, 100)

28. La similitud de Jaccard es una fórmula que te dice qué tan similares son dos conjuntos. Se define como la cardinalidad de la intersección dividida por la cardinalidad de la unión. ¿Qué opción es una implementación precisa en Python?

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- a) ☐ def jaccard(a, b): return len (a | b) / len (a & b)

Completa las respuestas en 60 minutos y envía tus respuestas al reclutador

- b) ☐ `def jaccard(a, b): return len (a & b) / len (a | b)`
- c) ☐ `def jaccard(a, b): return len (a && b) / len (a || b)`
- d) ☐ `def jaccard(a, b): return a.intersection(b) / a.union(b)`

29. ¿Qué necesita hacer para instalar paquetes adicionales en Python?

- a) ☐ Use un compilador de C como gcc o clang.
- b) ☐ Usa un administrador de paquetes como pip o conda.
- c) ☐ Usa un IDE como Notepad++ o Idle.
- d) ☐ Use un administrador de paquetes como NPM o NuGet.

30. Cuando usa NumPy en Python, ¿cómo verifica la dimensionalidad (número y longitud de las dimensiones) de un objeto llamado my_object?

- a) ☐ `my_object.get_shape()`
- b) ☐ `my_object.shape`
- c) ☐ `my_object.dim()`
- d) ☐ `len(my_object)`

31. Suponga que desea volver a verificar si dos matrices se pueden multiplicar usando NumPy para fines de depuración. ¿Cómo completaría este fragmento de código llenando los espacios en blanco con las variables apropiadas?

```
import numpy as np

def can_matrices_be_multiplied (matrix1, matrix2):
    rowsMat1, columnsMat1 = matrix1.shape
    rowsMat2, columnsMat2 = matrix2.shape

    if _____ == _____ :
        print('The matrices can be multiplied!')
        return True
    else:
        return False
```

- a) ☐ `columnsMat1; rowsMat1;`
- b) ☐ `columnsMat1; rowsMat2;`
- c) ☐ `columnsMat1; columnsMat2;`
- d) ☐ `columnsMat2; rowsMat1;`

32. ¿Qué fragmento de código imprimirá Mi nombre es Joffrey, hijo de Robert?

a) ☐ :

```
class Father():
    name = 'Robert'

class Person(Father):
    def __init__(self, name):
        self.fathername = super.name
        self.name = name

    def introduce(self):
        print("My name is", self.name, "son of", self.fathername)

king = Person("Joffrey")
king.introduce()
```

b) ☐ :

```
class Father():
    name = 'Robert'

class Person(Father):
    def __init__(self, name):
        self.fathername = self.name
        self.name = name

    def introduce(self):
        print("My name is", self.name, "son of", self.fathername)

king = Person("Joffrey")
king.introduce()
```

c) ☐ :

```
class Father():
    name = 'Robert'

class Person(Father):
    def __init__(self, name):
        self.name = name

    def introduce(self):
        print("My name is", self.name, "son of", super.name)

king = Person("Jofrey")
```

```
rey.presentar()
```

d) ☐ :

```
class Padre():
    nombre = 'Roberto'
```

```
Persona de clase (Padre):
    def __init__(uno mismo, nombre):
        self.nombre = nombre

    def presentar (auto):
        print("Mi nombre es", self.name, "hijo de", base.name)
```

```
rey = Persona("Joffrey")
rey.presentar()
```

33. ¿El proceso de decapado en Python incluye?

- a) ☐ conversión de una jerarquía de objetos de Python en un flujo de bytes
- b) ☐ conversión de una tabla de datos en una lista
- c) ☐ conversión de un flujo de bytes en la jerarquía de objetos de Python
- d) ☐ conversión de una lista en una tabla de datos

Explicación: el decapado es el proceso de esterilización de un objeto de Python, es decir, la conversión de un flujo de bytes en una jerarquía de objetos de Python. El reverso de este proceso se conoce como decapado.

34. ¿Cuál es la salida del siguiente programa?

```
print("programming".center())
```

- a) ☐ cr
- b) ☐ programación
- c) ☐ El error dice TypeError: el centro esperaba al menos 1 argumento, obtuvo 0
- d) ☐ Ninguna de las anteriores

35. Considere el siguiente fragmento de código que usa decoradores para calcular el tiempo de ejecución de la función `execution_fn`:

```
import functools
import time

def timer(MISSING_ARG_1):
    @functools.wraps(func)
    def wrapper(*args, **kwargs):
        start_time = time.perf_counter()
        rval = func(*args, **kwargs)
        end_time = time.perf_counter()
        duration = end_time - start_time
        print(f"Executed in {duration:.4f} seconds")
        return MISSING_ARG_2
    return MISSING_ARG_3

@timer
def execution_fn():
    for i in range(3):
        time.sleep(1)

execution_fn()
```

¿Cuáles de las siguientes opciones son los argumentos que faltan?

a) ☐ :

MISSING_ARG_1 = wrapper

MISSING_ARG_2 = rval

MISSING_ARG_3 = func

b) ☐ :

MISSING_ARG_1 = func

MISSING_ARG_2 = rval

MISSING_ARG_3 = wrapper

c) ☐ :

MISSING_ARG_1 is empty

MISSING_ARG_2 = rval

MISSING_ARG_3 = wrapper

d) ☐ :

MISSING_ARG_1 is empty

Completa las respuestas en 60 minutos y envía tus respuestas al reclutador

```
MISSING_ARG_2 = rval
```

```
MISSING_ARG_3 = func
```

36. Los puntos de datos en pyplot se llaman

- a) ☐ punteros
- b) ☐ puntos
- c) ☐ marcadores
- d) ☐ ninguno de esos