



Curso de Entrenamiento Para la Certificación PYTHON



Niveles de Certificación

ENTRY

ASSOCIATE

PROFESSIONAL



PCEP-30-xx
Certified Entry-Level Python
Programmer Certification



PCAP-31-xx
Certified Associate in Python
Programming Certification



PCPP-32-1xx
Certified Professional in Python
Programming 1 Certification



PCPP-32-2xx
Certified Professional in Python
Programming 2 Certification

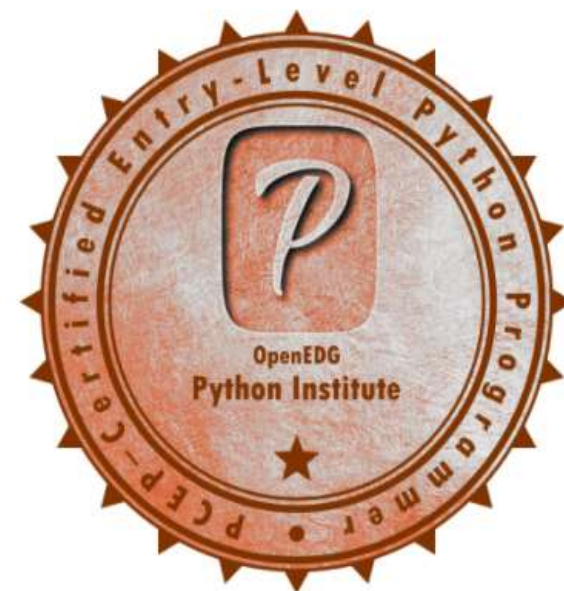


Available via **OpenEDG Testing Service**

Available via **Authorized Pearson VUE Testing Centers / OnVUE online proctoring**

PCEP: Información del examen

- **Nombre del examen:** PCEP - Programador de Python de nivel de entrada certificado
- **Código de examen/Versión de examen:** PCEP-30-01 (Estado: Retirada: 31 de diciembre de 2022) | [PCEP-30-02](#) (Estado: [Activo](#))
- **Nivel de examen:** Entrada
- **Certificaciones asociadas:** PCAP – Asociado certificado en programación Python ([PCAP-31-03](#) , PCAP-31-02, PCAP-31-01)
- **Requisitos previos:** ninguno
- **Duración:** PCEP-30-01: 45 minutos (examen) + aprox. 5 minutos (Acuerdo de Confidencialidad/Tutorial) | PCEP-30-02: 40 minutos (examen) + aprox. 5 minutos (acuerdo de confidencialidad/tutorial)
- **Número de preguntas:** 30
- **Formato:** preguntas de selección única y selección múltiple, arrastrar y soltar, llenar espacios, ordenar | Pitón 3.x
- **Puntaje de aprobación:** 70%
- **Idioma:** Inglés
- **Cursos alineados:** [PCAP: Fundamentos de programación en Python](#) (Cisco Networking Academy, dirigido por un instructor: módulos 1 a 4), [Python Essentials 1](#) : plataforma educativa OpenEDG, autoinscripción/autoaprendizaje
- **Precio del examen completo:** 59 USD (una sola toma), 76,70 USD (con una repetición) | [Compra de comprobante de examen](#)
- **Canal de entrega:** [Servicio de pruebas OpenEDG](#)
- **Políticas de prueba:** haga clic [aquí](#) para ver las políticas de prueba
- **Plan de estudios del examen:** Haga clic [aquí](#) para ver el plan de



Objetivos del examen

El candidato de la prueba debe demostrar el conocimiento suficiente de los siguientes conceptos:

1. Los fundamentos de la programación informática, es decir, cómo funciona la computadora, cómo se ejecuta el programa, cómo se define y construye el lenguaje de programación, cuál es la diferencia entre compilación e interpretación, qué es Python, cómo se posiciona entre otros lenguajes de programación , y qué distingue a las diferentes versiones de Python;

2. Los métodos básicos de formato y salida de datos ofrecidos por Python, junto con los principales tipos de datos y operadores numéricos, sus relaciones mutuas y enlaces; el concepto de variables y convenciones de nombres de variables; el operador de asignación, las reglas que rigen la construcción de expresiones; la introducción y conversión de datos;

3. Valores booleanos para comparar valores de diferencia y controlar las rutas de ejecución utilizando las instrucciones *if* y *if-else* ; la utilización de bucles (*while* y *for*) y cómo controlar su comportamiento usando las instrucciones *break* y *continue* ; la diferencia entre operaciones lógicas y bit a bit; el concepto de listas y procesamiento de listas, incluida la iteración proporcionada por el bucle *for* y el corte; la idea de matrices multidimensionales;

4. La definición y el uso de funciones: su razón de ser, propósito, convenciones y trampas; el concepto de pasar argumentos de diferentes maneras y establecer sus valores predeterminados, junto con los mecanismos para devolver los resultados de la función; problemas con el alcance del nombre; nuevos agregados de datos: tuplas y diccionarios, y su papel en el procesamiento de datos.

El alcance de la certificación incluye las siguientes áreas temáticas:

1. Programación informática y fundamentos de Python (18 %)
2. Flujo de control: bloques y bucles condicionales (28 %)
3. Recopilaciones de datos: listas, tuplas, diccionarios y cadenas (26 %)
4. Funciones y Excepciones (28%)

Información adicional:

- **Página de información** : [Página de certificación PCEP](#)
- **Canal de entrega** : [servicio de prueba OpenEDG](#)
- **Versión del examen** : PCEP-30-02 (no supervisado)
- **Políticas de examen** : [Página de políticas de prueba de PCEP](#)
- **Nota importante** : el cupón se puede usar para un solo intento (cupón de un solo disparo)
- **Estado del examen** : activo



The image displays a Python Institute PCEP-30-02 Certificate and Exam Voucher. The certificate is a white document with a blue header featuring the Python Institute logo and the text "PYTHON INSTITUTE". The main body of the certificate is white with a large, circular, gold-colored seal in the center. The seal contains the Python logo (a stylized 'P' inside a circle) and the text "PCEP-Certified Entry-Level Python Programmer". Below the seal, it says "OpenEDG Python Institute". The certificate is partially covered by a blue banner that reads "Exam". In the top right corner, there is a green diagonal banner with the text "OEDG". At the bottom, a blue banner displays the text "PCEP-30-02 (Single Voucher)" and the price "\$59.00".

PCEP-30-02 (Single Voucher) **\$59.00**

El alcance de la certificación incluye las siguientes áreas temáticas:

1. Programación informática y fundamentos de Python (18 %)
2. Flujo de control: bloques y bucles condicionales (28 %)
3. Recopilaciones de datos: listas, tuplas, diccionarios y cadenas (26 %)
4. Funciones y Excepciones (28%)

Información adicional:

- **Página de información** : [Página de certificación PCEP](#)
- **Canal de entrega** : [servicio de prueba OpenEDG](#)
- **Versión del examen** : PCEP-30-02 (no supervisado)
- **Políticas de examen** : [Página de políticas de prueba de PCEP](#)
- **Nota importante** : el cupón incluye **una** opción de repetición.
El candidato debe esperar 7 días antes de que se le permita volver a tomar un examen reprobado (vale con una repetición)
- **Estado del examen** : activo



Los Practice Tests incluyen **todos los temas y objetivos que cubre el examen PCEP-30-0x** . La experiencia de evaluación, incluida la calificación y la progresión de la prueba, **simula el examen real** , lo que permite a los candidatos a la prueba tener una buena idea del tipo de preguntas que aparecerán en el examen, así como ayudarlos a comprender cómo se califican y ponderan a lo largo del examen. secciones.

- **Canal de entrega** : [Servicio de pruebas de OpenEDG](#) (haga clic en *Práctica* e ingrese el código para activar el Compendio de pruebas de práctica)
- **Alineación con el examen** : PCEP: programador Python de nivel de entrada certificado (PCEP-30-02)
- **Información adicional** : el límite de lanzamiento se aplica a ciertos recursos.



The image shows two practice test booklets. The one in the foreground is blue and white, labeled 'PCEP-30-02 Practice Test'. It features a circular seal with the text 'PCEP-Certified Entry-Level Python Programmer' around the perimeter. In the center of the seal is a large 'P' logo, with 'OpenEDG Python Institute' and a star below it. The Python logo is also visible on the booklet. A green banner in the top right corner of the image area says 'OEDG'. A green ribbon at the bottom right of the image area says 'PracticeTest'.

PCEP-30-02 Practice Test	\$29.00
--------------------------	---------

Bloque de examen n.º 1: programación informática y fundamentos de Python (18 %)

Objetivos que cubre el bloque (7 ítems de examen)

- **PCEP 1.1 Comprender términos y definiciones fundamentales**

interpretación y el intérprete, compilación y el compilador, léxico, sintaxis y semántica

- **PCEP 1.2 Comprender la lógica y la estructura de Python Palabras**

clave, instrucciones, sangrado, comentarios

- **PCEP 1.3 Introduce literales y variables en el código y usa diferentes sistemas de numeración**

Booleanos, enteros, números de punto flotante, notación científica, cadenas, sistema numérico binario, octal, decimal y hexadecimal, variables, convenciones de nomenclatura, implementación de recomendaciones PEP-8

- **PCEP 1.4 Elegir operadores y tipos de datos adecuados al problema**

operadores numéricos: `** * / % //` `+` `-`, operadores de cadenas: `*` `+`, operadores de asignaciones y atajos, operadores: unarios y binarios, prioridades y vinculación, operadores bit a bit: `~ & ^ | << >>`, operadores booleanos: `not and or`, expresiones booleanas, operadores relacionales (`== != > >= < <=`), la precisión de los números de coma flotante, conversión de tipos

- **PCEP 1.5 Realizar operaciones de consola de entrada/salida**

`print()`, funciones `input()`, parámetros de palabras clave `sep=` y `end=`, funciones `int()` y `float()`

Bloque de examen n.º 2: flujo de control: bloques y bucles condicionales (29 %)

Objetivos que cubre el bloque (8 ítems de examen)

- **PCEP 2.1 Tome decisiones y bifurque el flujo con las** sentencias condicionales de la instrucción if: if, if-else, if-elif, if-elif-else, sentencias condicionales múltiples, sentencias condicionales anidadas
- **PCEP 2.2 Realiza diferentes tipos de iteraciones de** la instrucción pass, creando bucles con while, for, range() e in; iterar a través de secuencias, expandir bucles con while-else y for-else, anidar bucles y sentencias condicionales, controlar la ejecución de bucles con break y continue

Bloque de examen n.º 3: Colecciones de datos: tuplas, diccionarios, listas y cadenas (25 %)

Objetivos que cubre el bloque (7 ítems de examen)

- **PCEP 3.1 Recopile y procese datos utilizando listas** que construyen vectores, indexación y división, la función len(), métodos básicos de lista (append(), insert(), index()) y funciones (len(), sorted(), etc.), la instrucción del; iterando a través de listas con el bucle for, inicializando bucles; operadores in y not in, listas de comprensión; copia y clonación, listas en listas: matrices y cubos
- **PCEP 3.2 Recolectar y procesar datos usando tuplas** tuplas: indexación, rebanado, construcción, inmutabilidad; tuplas vs. listas: similitudes y diferencias, listas dentro de tuplas y tuplas dentro de listas
- **PCEP 3.3 Recolectar y procesar datos usando diccionarios** diccionarios: construir, indexar, agregar y quitar claves; iterando a través de diccionarios y sus claves y valores, comprobando la existencia de claves; Métodos keys(), items() y valores()
- **PCEP 3.4 Operar con cadenas** construcción de cadenas, indexación, rebanado, inmutabilidad; escapar usando el carácter \; comillas y apóstrofes dentro de cadenas, cadenas de varias líneas, funciones y métodos básicos de cadenas

Bloque de examen n.º 4: funciones y excepciones (28 %)

Objetivos que cubre el bloque (8 ítems de examen)

- **PCEP 4.1 Descompone el código usando funciones**

que definen e invocan funciones y generadores definidos por el usuario; la palabra clave de retorno, resultados devueltos, la palabra clave Ninguno, recursividad

- **PCEP 4.2 Organizar la interacción entre la función y su entorno**

parámetros vs argumentos; paso de argumentos posicionales, de palabras clave y mixtos; valores de parámetros predeterminados, alcances de nombres, ocultación de nombres (sombreado), la palabra clave global

- **Jerarquía de excepciones incorporadas de PCEP 4.3 Python**

BaseException, Exception, SystemExit, KeyboardInterrupt, abstractiveExceptions, ArithmeticError, LookupError junto con IndexError y KeyError; Excepciones TypeError y ValueError, la excepción AssertionError junto con la palabra clave assert

- **PCEP 4.4 Conceptos básicos del manejo de excepciones de Python Try**

-except, try-except Exception, ordenar las ramas de excepción, propagar excepciones a través de los límites de la función; delegar responsabilidad para el manejo de excepciones

Última actualización: 2 de febrero de 2022

Alineado con PCEP 30-02

PCAP: información del examen

- **Nombre de la credencial:** PCAP – Asociado certificado en programación Python
- **Código de examen/versión actual del examen:** PCAP-31-02 (Pearson VUE, retirada), PCAP-31-03 (supervisión en línea de Pearson VUE/OnVUE)
- **Versiones de examen retiradas :** PCAP-31-01 (retirada), PCAP-NP-31-02 (retirada)
- **Nivel de examen:** Asociado
- **Certificaciones asociadas:** [PCPP | Profesional certificado en programación Python \(PCPP-32-101 y PCPP-32-201\)](#) A partir de noviembre de 2020 (PCPP1)
- **Requisitos previos:** ninguno
- **Duración:** 65 minutos (examen) + 10 minutos (Acuerdo de Confidencialidad/Tutorial)
- **Número de preguntas:** 40
- **Formato:** Preguntas de opción única y opción múltiple | Pitón 3.x
- **Puntaje de aprobación:** 70% (PCAP-31-02, PCAP-31-03)
- **Idioma:** Inglés
- **Cursos alineados:** [PCAP: Fundamentos de programación en Python](#) (Cisco Networking Academy, dirigido por un instructor), [Python Essentials - Parte 1](#) , [Python Essentials - Parte 2](#) (Edube Interactive - OpenEDG Education Platform, autoinscripción/autoaprendizaje)
- **Precio del examen completo:** USD 295 ([consultar descuentos disponibles](#))
- **Canal de entrega:** [Pearson VUE Testing Centers](#) (PCAP-31-02, PCAP-31-03), [supervisión en línea OnVUE](#) de Pearson VUE (PCAP-31-03)



Objetivos del examen

El candidato de la prueba debe demostrar el conocimiento suficiente de los siguientes conceptos:

1. Los fundamentos de la programación informática, es decir, cómo funciona la computadora, cómo se ejecuta el programa, cómo se define y construye el lenguaje de programación, cuál es la diferencia entre compilación e interpretación, qué es Python, cómo se posiciona entre otros lenguajes de programación , y qué distingue a las diferentes versiones de Python;

2. Los métodos básicos de formato y salida de datos ofrecidos por Python, junto con los principales tipos de datos y operadores numéricos, sus relaciones mutuas y enlaces; el concepto de variables y convenciones de nombres de variables; el operador de asignación, las reglas que rigen la construcción de expresiones; la introducción y conversión de datos;

3. Valores booleanos para comparar valores de diferencia y controlar las rutas de ejecución utilizando las instrucciones *if* y *if-else* ; la utilización de bucles (*while* y *for*) y cómo controlar su comportamiento usando las instrucciones *break* y *continue* ; la diferencia entre operaciones lógicas y bit a bit; el concepto de listas y procesamiento de listas, incluida la iteración proporcionada por el bucle *for* y el corte; la idea de matrices multidimensionales;

4. La definición y el uso de funciones: su razón de ser, propósito, convenciones y trampas; el concepto de pasar argumentos de diferentes maneras y establecer sus valores predeterminados, junto con los mecanismos para devolver los resultados de la función; problemas con el alcance del nombre; nuevos agregados de datos: tuplas y diccionarios, y su papel en el procesamiento de datos;

5. Módulos de Python: su razón de ser, función, cómo importarlos de diferentes maneras y presentar el contenido de algunos módulos estándar provistos por Python; la forma en que se acoplan los módulos para formar paquetes; el concepto de excepción y la implementación de excepciones de Python, incluida la instrucción *try-except* , con sus aplicaciones, y la instrucción *raise* ; cadenas y sus métodos específicos, junto con sus similitudes y diferencias en comparación con las listas;

6. Los fundamentos de OOP (Programación Orientada a Objetos) y la forma en que se adoptan en Python, mostrando la diferencia entre OOP y el enfoque procedimental clásico; las características objetivas estándar: herencia, abstracción, encapsulación y polimorfismo, junto con problemas específicos de Python, como variables de instancia frente a clase, y la implementación de la herencia de Python; naturaleza objetiva de las excepciones; Los generadores de Python (la instrucción de *rendimiento*) y los cierres (la palabra clave *lambda*); los medios que los desarrolladores de Python pueden usar para procesar (crear, leer y escribir) archivos.

PCAP: la certificación de asociado certificado en programación

Python es una credencial profesional que mide su capacidad para realizar tareas de codificación relacionadas con los conceptos básicos de programación en el lenguaje Python y las nociones y técnicas fundamentales utilizadas en la programación orientada a objetos.

- Información: [Página de Certificación PCAP](#)
- Canales de entrega: [Centros de pruebas de Pearson VUE](#) (PCAP-31-02/PCAP-31-03) / Supervisión en línea de OnVUE (PCAP-31-03)
- Versión del examen: PCAP-31-02/PCAP-31-03 (supervisado)



The image displays a Python Institute PCAP (Professional Certified Associate in Python Programming) certificate. The certificate is white with a blue border and features the Python Institute logo, the text 'CERTIFICATE of proficiency in the python programming language', and a circular seal that reads 'PCAP - Certified Associate in Python Programming'. Below the seal, it says 'OpenEDG Python Institute' and '★★★'. To the right of the certificate, a blue banner reads 'Exam + Retake'. In the top right corner, a blue banner reads 'PVUE OnVUE'. At the bottom, a blue banner displays the exam code 'PCAP-31-03' and the price '\$295.00'.

PVUE OnVUE

Exam + Retake

PCAP-31-03 **\$295.00**

PCAP-31-03 Vale de examen

PCAP: la certificación de asociado certificado en programación

Python es una credencial profesional que mide su capacidad para realizar tareas de codificación relacionadas con los conceptos básicos de programación en el lenguaje Python y las nociones y técnicas fundamentales utilizadas en la programación orientada a objetos.

PCAP-31-03 Prueba de práctica

El Compendio de estudio presenta **materiales de preparación para el examen y recursos útiles** en forma de *páginas de estudio* , *preguntas de la sección del examen y pruebas de práctica* , que conforman un paquete sólido de ayudas para el aprendizaje y la práctica para los candidatos que desean desarrollar su confianza antes de realizar los exámenes.

Los Practice Tests incluyen **todos los temas y objetivos cubiertos por el examen PCAP-31-03** . La experiencia de evaluación, incluida la calificación y la progresión de la prueba, **simula el examen real** , lo que permite a los candidatos a la prueba tener una buena idea del tipo de preguntas que aparecerán en el examen, así como ayudarlos a comprender cómo se califican y ponderan a lo largo del examen. secciones.



PCAP-31-03 Examen + Prueba de práctica \$ 319.00

Exam block #1: Modules and Packages (12%)

Objectives covered by the block (6 items)

- import variants; advanced qualifying for nested modules
- `dir()`; `sys.path` variable
- `math`: `ceil()`, `floor()`, `trunc()`, `factorial()`, `hypot()`, `sqrt()`;
`random`: `random()`, `seed()`, `choice()`, `sample()`
- `platform`: `platform()`, `machine()`, `processor()`, `system()`,
`version()`, `python_implementation()`,
`python_version_tuple()`
- `idea`, `__pycache__`, `__name__`, public variables, `__init__.py`
- searching for modules/packages; nested packages vs directory tree

Exam block #2: Exceptions (14%)

Objectives covered by the block (5 items)

- `except`, `except:-except`; `except:-else:`, `except (e1,e2)`
- the hierarchy of exceptions
- `raise`, `raise ex`, `assert`
- event classes, `except E as e`, `arg` property
- self-defined exceptions, defining and using

Exam block #3: Strings (18%)

Objectives covered by the block (8 items)

- ASCII, UNICODE, UTF-8, codepoints, escape sequences
- `ord()`, `chr()`, literals
- indexing, slicing, immutability
- iterating through,
- concatenating, multiplying, comparing (against strings and numbers)
- `in`, `not in`
- `.isxxx()`, `.join()`, `.split()`
- `.sort()`, `sorted()`, `.index()`, `.find()`, `.rfind()`

Exam block #4: Object-Oriented Programming (34%)

Objectives covered by the block (12 items)

- ideas: class, object, property, method, encapsulation, inheritance, grammar vs class, superclass, subclass
- instance vs class variables: declaring, initializing
- `__dict__` property (objects vs classes)
- private components (instance vs classes), name mangling
- methods: declaring, using, self parameter
- introspection: `hasattr()` (objects vs classes), `__name__`, `__module__`, `__bases__` properties
- inheritance: single, multiple, `isinstance()`, overriding, not is and is operators
- inheritance: single, multiple, `isinstance()`, overriding, not is and is operators
- constructors: declaring and invoking
- polymorphism
- `__name__`, `__module__`, `__bases__` properties, `__str__()` method
- multiple inheritance, diamonds

Exam block #5: Miscellaneous (List Comprehensions, Lambdas, Closures, and I/O Operations) (22%)

Objectives covered by the block (9 items)

- list comprehension: if operator, using list comprehensions
- lambdas: defining and using lambdas, self-defined functions taking lambda as arguments; map(), filter();
- closures: meaning, defining, and using closures
- I/O Operations: I/O modes, predefined streams, handles; text/binary modes
open(), errno and its values; close()
.read(), .write(), .readline(); readlines() (along with bytearray())

Last updated: September 17, 2020

Aligned with PCAP-31-03 (Pearson VUE/OnVUE Online
Proctoring from Pearson VUE)



NUMBER: OPEN 2020 10 276

Date of issue: **October 12, 2020**
Payment method: **Credit/Debit Card**
PO: **OPEN-PO-AFYX-NJ1N-EXUF**

No.	Product / Service	Unit	Qty	Unit Net price	VAT rate	Net value	Gross value
1	PCAP - Certified Associate in Python Programming (PCAP-31-02/PCAP-31-03)	items	1	88.50	NP*	88.50	88.50
Total (USD)						88.50	88.50

Manos a la preparación



Question 1

The following expression

```
2 ** 3 ** 2 ** 1
```

is:

- A. invalid
- B. equal to 16
- C. equal to 16.0
- D. equal to 512
- E. equal to 64
- F. equal to 128.0

Question 2

If you want to build a string that reads:

```
Peter's sister's name's "Anna"
```

which of the following literals would you use? (Select all that apply)

- A. `"Peter's sister's name's \"Anna\""`
- B. `'Peter\'s sister\'s name\'s \"Anna\"'`
- C. `"Peter's sister's name's "Anna""`
- D. `'Peter's sister's name's "Anna"'`

Question 2

If you want to build a string that reads:

```
Peter's sister's name's "Anna"
```

which of the following literals would you use? (Select all that apply)

- A. `"Peter's sister's name's \"Anna\""`
- B. `'Peter\'s sister\'s name\'s \"Anna\"'`
- C. `"Peter's sister's name's "Anna""`
- D. `'Peter's sister's name's "Anna"'`

Question 3

What is the expected output of the following snippet?

```
i = 250
while len(str(i)) > 72:
    i *= 2
else:
    i //= 2
print(i)
```

- A. 125
- B. 250
- C. 72
- D. 500

Question 4

What snippet would you insert in the line indicated below:

```
n = 0
while n < 4:
    n += 1
    # insert your code here
```

to print the following string to the monitor after the loop finishes its execution:

```
1 2 3 4
```

- A. `print(n)`
- B. `print(n, sep=" ")`
- C. `print(n, end=" ")`
- D. `print(n, " ")`

Question 5

What is the value type returned after executing the following snippet?

```
x = 0
y = 2
z = len("Python")
x = y > z
print(x)
```

- A. int
- B. float
- C. str
- D. bool
- E. NoneType

Question 6

What will the final value of the `Val` variable be when the following snippet finishes its execution?

```
Val = 1  
Val2 = 0  
Val = Val ^ Val2  
Val2 = Val ^ Val2  
Val = Val ^ Val2
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. The code is erroneous

Question 7

Which line can be used instead of the comment to cause the snippet to produce the following expected output? (Select all that apply)

Code:

```
z, y, x = 2, 1, 0
x, z = z, y
y = y - z
# put line here
print(x, y, z)
```

Expected output:

```
0, 1, 2
```

Question 8

What is the expected output of the following snippet?

```
a = 0
b = a ** 0
if b < a + 1:
    c = 1
elif b == 1:
    c = 2
else:
    c = 3
print(a + b + c)
```

- A. 1
- B. 2
- C. 3
- D. The code is erroneous

Question 9

How many stars (*) does the following snippet print?

```
i = 10
while i > 0 :
    i -= 3
    print("*")
    if i <= 3:
        break
else:
    print("*")
```

- A. three
- B. two
- C. one
- D. The code is erroneous

Question 9

How many stars (*) does the following snippet print?

```
i = 10
while i > 0 :
    i -= 3
    print("*")
    if i <= 3:
        break
else:
    print("*")
```

- A. three
- B. two
- C. one
- D. The code is erroneous

Question 18

How many stars (*) will the following snippet send to the console?

```
lst = [[c for c in range(r)] for r in range(3)]  
for x in lst:  
    for y in x:  
        if y < 2:  
            print('*', end='')
```

- A. One
- B. Two
- C. Three
- D. Four
- E. The program will cause a runtime exception/error