

Examen de Autoevaluación del Examen integrador

Examen teórico

Este examen es para que te prepares para la parte teórica de tu Examen Integrador.

Instrucciones:

- Contesta el examen a mano.
 - Usa Thonny para comprobar tus resultados.
1. Escribe lo que muestra Python al ejecutar cada uno de las siguientes líneas.
 - a. `print(4 + 7 / 2 - 5 * 3)`
 - b. `print(6 / 4 + 5 % 3 + 8 // 3)`
 - c. `print((5 + 11 // 3) / (2 ** 3) + 2)`
 - d. `print(2 * 3 + 12 / 4 / 6 + 6 - 5 / 2)`
 - e. `print(4 % 7 - 6 % 2 + 9 % 4)`
 2. Escribe lo que muestra Python al ejecutar cada uno de las siguientes líneas.
 - a. `print((3.0 * -2 > -10) and (5 * 4 == 15))`
 - b. `print((5 * 3.0 != 9) or (10 // 3 > 3))`
 - c. `print(not (5 != 8) or (3.0 >= 3))`
 - e. `print((3 * 4 == 12) and (5 < 2 ** 3) or (8 + 1 <= 9.0))`

3. Escribe lo que muestra Python al ejecutar las siguientes secciones de código.

a.

```
x = 9
y = 5
if x > 5 :
    print(x)
elif y < 5 :
    print(y)
else :
    print(x + y)
```

b.

```
x = 1
y = 2
if x > 5 :
    print(x)
elif y < 5 :
    print(y)
else :
    print(x + y)
```

c.

```
x = 3
y = 5
if x > 2 :
    print(x)
if y < 5 :
    print(y)
if x <= 5 and y >= 5 :
    print(x + y)
```

d.

```
x = 6
y = 4
if x > 2 :
    print(x)
if y < 4 :
    print(y)
else :
    print(x + y)
```

4. Escribe lo que muestra Python al ejecutar las siguientes secciones de código.

a. Programa 1

```
def funcion_uno(a, b) :  
    a = a + 3  
    c = b / 2  
    return a + c  
  
def main() :  
    valor1 = 5  
    valor2 = 10  
    respuesta = funcion_uno(valor1, valor2)  
    print(respuesta)  
  
main()
```

b. Programa 2

```
def uno (a, b): print(a, b)  
  
def dos(x, y):  
    uno(x, y) x = 7  
    uno(x, y) y = 2  
    uno(y, x)  
  
def main() : p = 1  
    q = 4 dos(p,q)  
    print(p, q)  
  
main()
```

c. Programa 3

```
def uno (a) :  
    a = a // 3  
    return a
```

```
def main():  
    x = 35  
    z = uno(x)  
    print(x, z)
```

```
main()
```

d. Programa 4

```
def calculo(x, y, z):  
    result = x + y * z  
    return result
```

```
def main():  
    p = 1  
    q = 2  
    r = 3  
    s = calculo(p, q, r)  
    print(s)
```

```
main()
```

5. Escribe lo que muestra Python al ejecutar cada uno de los siguientes incisos:

- a.

```
for a in range(2, 10):  
    print(a)
```
- b.

```
for b in range(10, 1, -2):  
    print(b)
```
- c.

```
for c in range(-2, 2):  
    print(c)
```
- d.

```
for d in range(-2):  
    print(d)
```
- e.

```
for e in range(0, 60, 11):  
    print(e)
```

6. Escribe lo que muestra Python al ejecutar cada uno de los siguientes incisos.

- a.

```
x = 7  
while x >= 0:  
    print(x)  
    x -= 2
```
- b.

```
y = 12  
z = 2  
while y > z:  
    print(y, z)  
    y = y - z  
    z = z + 1
```

7. Escribe una sección de código que use un estatuto while en lugar del for para cada uno de los siguientes incisos:

a.

```
for var in range (3, 15, 4):  
    print(var)
```

b.

```
for var in range (25, 5, -5):  
    print(var)
```

SOLUCIÓN

a.

```
cont = 3  
while cont < 15:  
    print(cont)  
    cont = cont + 4
```

b.

```
cont =  
25  
while cont > 5:  
    print(cont)  
    cont = cont - 5
```

8. Escribe lo que muestra Python al ejecutar cada uno de los siguientes incisos:

a) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`print(lista[3:6])`

b) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista[4] = 50`
`print(lista)`

c) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista.insert(4, 50)`
`print(lista)`

d) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista.append(50)`
`print(lista)`

e) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`print(len(lista))`

f) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`del(lista[3])`
`print(lista)`

g) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista[5] = 35`
`print(lista)`

h) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista[7] = lista[5] + lista[6]`
`print(lista)`

i) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista[4] = 2 * lista[2] - 10`
`print(lista)`

j) `lista = [10, 12, 14, 16, 18, 20, 22, 24]`
`lista[4] = 2 * lista[2] - 10`
`print(lista)`

9. Escribe lo que muestra Python al ejecutar cada uno de los siguientes incisos:

a) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`print(matriz[2][2])`

b) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`print(matriz[2][0])`

c) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`print(matriz[1])`

d) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`print(matriz[0])`

e) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`matriz.append([4, 9, 6])`
`print(matriz)`

f) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`matriz.insert(1, [4, 9, 6])`
`print(matriz)`

g) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`matriz.insert(2, [5, 7, 8])`
`print(matriz)`

h) `matriz = [[0, 2, 4], [1, 3, 5], [7, 8, 9]]`
`matriz[1][1] = matriz[2][2] - 2`
`print(matriz)`

10. Escribe lo que muestra Python al ejecutar cada uno de los siguientes incisos:

- a)

```
cadena = "Computacion"
sub_cadena = cadena[0 : 3]
print(sub_cadena)
```
- b)

```
cadena = "Computacion"
sub_cadena = cadena[ : 3]
print(sub_cadena)
```
- c)

```
cadena = "Computacion"
sub_cadena = cadena[3]
print(sub_cadena)
```
- d)

```
cadena = "Computacion"
sub_cadena = cadena[6 : ]
print(sub_cadena)
```
- e)

```
cadena = "Computacion"
sub_cadena = cadena[-6 : ]
print(sub_cadena)
```
- f)

```
cadena = "Computacion"
res = cadena.find('o')
print(res)
```
- g)

```
cadena = "Computacion"
cadena2 = cadena.replace('o', 'u')
print(cadena2)
```
- h)

```
cadena = "Computacion"
cadena2 = cadena.upper()
print(cadena2)
```
- i)

```
cadena = "Computacion"
cadena2 = cadena.lower()
print(cadena2)
```
- j)

```
cadena = "C o m p u t a c i o n"
cadena2 = cadena.split(' ')
print(cadena2)
```