“Examen de autoevaluación: Examen integrador”

Parte 2: Fundamentos de Python, funciones y estructuras de datos

---

Este examen es para que te prepares para la parte 2 de tu Examen Integrador. Al final se encuentran las respuestas.

Instrucciones:

- Contesta el examen a mano sin ver la sección de las respuestas.
- Revisa tu examen usando las respuestas que se encuentran al final.
- Para cada una de las respuestas que tengas equivocada usa un bloque de código de Google Colaboratory para probar los estatutos y encontrar la razón por la que te equivocaste.

1. 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)
```

2. 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()
```

3. **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)
```

4. **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
```

5. **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)
```

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

a)
```
str = "Computacion"
print(str[-6 : -3])
```

b)
```
str = "Computacion"
print(str[3 : 8])
```

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

d)
```
str = "Computacion"
print(str[-3 : ])
```

e)
```
str = "Computacion"
print(str[5 : ])
```

f)
```
str = "Computacion"
print(str[ : 5])
```

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

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

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

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

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

7. 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]
print (lista[-7 : -2])

c) lista = [10, 12, 14, 16, 18, 20, 22, 24]
print (lista[-5 :])

d) lista = [10, 12, 14, 16, 18, 20, 22, 24]
print (lista[: -5])

e) lista = [10, 12, 14, 16, 18, 20, 22, 24]
print (lista[: 5])

f) lista = [10, 12, 14, 16, 18, 20, 22, 24]
print (lista[5 :])

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

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

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

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

Respuestas a los ejercicios

Usa esta sección para revisar tus respuestas.

Respuestas al problema 1

a. 9

b. 2

c. 3
   8

d. 6
   10

Respuestas al problema 2

a. Programa 1
   13.0

b. Programa 2
   1 4
   7 4
   2 7
   1 4

c. Programa 3
   35 11

d. Programa 4
   7

Respuestas al problema 3

a.
   2
   3
   4
   5
   6
   7
   8
   9

b.
   10
   8
   6
   4
   2

c.

-2
-1
0
1

d.

El programa no muestra nada en la pantalla.

e.

0
11
22
33
44
55

Respuestas al problema 4

a.

7
5
3
1

b.

12 2
10 3
7 4

Respuestas al problema 5

a.

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

b.

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

Respuestas al problema 6

a. tac
b. putac
c. Computac
d. ion
e. tacion
f. Compu
g. 1
h. Cumputaciun
i. COMPUTACION
j. Computación
k. ['C', 'o', 'm', 'p', 'u', 't', 'a', 'c', 'i', 'o', 'n']

Respuestas al problema 7

a. [16, 18, 20]
b. [12, 14, 16, 18, 20]
c. [16, 18, 20, 22, 24]
d. [10, 12, 14]
e. [10, 12, 14, 16, 18]
f. [20, 22, 24]
g. [10, 12, 14, 16, 50, 20, 22, 24]
h. [10, 12, 14, 16, 50, 20, 22, 24]
i. [10, 12, 14, 16, 50, 18, 20, 22, 24]
j. [10, 12, 14, 16, 18, 20, 22, 24, 50]

8. 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"

```
res = cadena.find('o')
print(res)
```

d) cadena = "Computacion"

```
cadena2 = cadena.replace('o', 'u')
print(cadena2)
```

e) cadena = "Computacion"

```
cadena2 = cadena.upper()
print(cadena2)
```

f) cadena = "Computacion"

```
cadena2 = cadena.lower()
print(cadena2)
```

g) cadena = "C o m p u t a c i o n"

```
cadena2 = cadena.split(' ')
print(cadena2)
```

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

a) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
lista.insert(4, 50)
print(lista)
```

b) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
lista.append(50)
print(lista)
```

c) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
print(len(lista))
```

d) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
del lista[3]
print(lista)
```

e) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
lista[7] = lista[5] + lista[6]
print(lista)
```

f) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
lista[4] = 2 * lista[2] - 10
print(lista)
```

g) lista = [10, 12, 14, 16, 18, 20, 22, 24]

```
acum = 0
for num in lista:
    if (num % 2 == 0):
        acum = acum + num * 2
print(acum)
```

10. Modos de acceso de un archivo de texto:

Modo de acceso	Descripción
r (read)	Abre un archivo para leer únicamente.
w (write)	Abre un archivo para escribir únicamente, reemplazando el contenido actual del archivo o creándolo si no existe.
a (append)	Abre un archivo para añadir únicamente, manteniendo el contenido actual y añadiendo los datos al final del archivo.
w+	Abre un archivo para escribir y leer, el archivo se crea.
r+	Abre un archivo para leer y escribir, el archivo debe existir.
a+	Abre un archivo para añadir y leer, el archivo debe existir.

11. Python permite leer un archivo de texto de las siguientes formas:

- **file.read ()** Lee todo el archivo
- **file.read (1)** Lee un caracter
- **file.readline ()** Lee línea por línea
- **file.readlines ()** Obtiene una lista con todas las líneas del archivo.

12. Python permite escribir en un archivo de texto de las siguientes formas:

- Escribe un texto en un archivo:

file.write ("Agregar contenido al archivo")

- Escribe una serie de líneas leyéndolas desde una lista:

lineas = ["Hola a todos\n", "Hasta luego\n“]
file.writelines (lineas)

13. ¿Cuál es la sintaxis correcta para abrir un archivo de texto?

14. ¿Cuál es la sintaxis correcta para cerrar un archivo de texto?

15. ¿Cuál es la sintaxis correcta de un diccionario?

16. ¿Cómo recorro los elementos de un diccionario con el ciclo for?