



Certificación PCAP

Actividad Final: Todas las lecciones

Actividad 1

ACTIVIDAD TODAS LAS LECCIONES

Objetivos

- | Profundizar en los conceptos vistos en todas las lecciones del curso

Contenido correspondiente a todas las lecciones

Actividad relacionada con todas las lecciones:

Con esta actividad final se pretende trabajar en los diferentes conceptos vistos en esta asignatura. En esta actividad debes enviar para cada pregunta la opción u opciones correctas, así como una breve explicación de tu elección.

Es importante que no te olvides de enviar la breve explicación de tu elección ya que si no lo envías se te valorará la pregunta como incorrecta. Además, indicar que en el caso de preguntas con varias opciones posibles sólo se corregirá la pregunta como correcta si se han indicado todas las opciones correctas y no hay ninguna incorrecta.

Pregunta 1:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
class MiClase:
    var = 0
    def __funcion(self):
        MiClase.var += 1
        return MiClase.var

o = MiClase()
o._MiClase__funcion()
print(o._MiClase__funcion())
```

- A) 0
- B) 1
- C) 2
- D) 3
- E) Al ejecutarse dará un error.

Pregunta 2:

¿Cuál o cuáles de las siguientes líneas de código funcionarán sin problemas cuando se sustituya el comentario por ella dentro del método `inc()` para que la salida del fragment de código sea igual a 3? (una o varias opciones)

```
class MiClase:
    var = 0

    def __init__(self):
        MiClase.var += 1
        self.prop = MiClase.var

    def get(self):
        return self.prop

    def put(self, val):
        self.prop = val

    def inc(self, val):
        # Inserta aquí tu código

o = MiClase()
o.inc(2)
print(o.get())
```

- A) `put(self.prop + val)`
- B) `self.put(self.get() + val)`
- C) `self.put(get() + val)`
- D) `self.put(self.prop + val)`
- E) `self.put(prop + val)`

Pregunta 3:

¿Cuál o cuáles de las siguientes expresiones devolverá `True`? (una o varias opciones)

- A) `' T ' in 'Toledo'`
- B) `'eo' in 'Toledo'`
- C) `'a' not in 'ABC'`
- D) `'True' == 1`
- E) `False == 0`
- F) `46 == 46.0`

Pregunta 4:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
def mi_funcion(x, y=1):  
    if x == 1:  
        return 1  
    return x + mi_funcion(x - y)  
  
print(mi_funcion(2))
```

- A) 12
- B) 21
- C) None
- D) 3
- E) Lanzará una excepción

Pregunta 5:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
def mi_funcion(n):  
    for i in range(1, n + 1, 2):  
        yield i  
  
for i in mi_funcion(4):  
    print(i, end='')
```

- A) 12
- B) 13
- C) 123
- D) 1234
- E) Lanzará una excepción

Pregunta 6:

Si necesitas capturar y manejar dos excepciones diferentes llamadas Ex1 y Ex2 en una única rama except, puedes escribirlo de la siguiente manera o maneras: (una o varias opciones)

- A) except Ex1 Ex2:
- B) except (Ex1, Ex2):
- C) except Ex1, Ex2:
- D) except Ex1+Ex2:

Pregunta 7:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
i = -5
while i < 0:
    i = i // 2
    if i % 3 == 0:
        break
else:
    i -= 4
print(i)
```

- A) -3
- B) -7
- C) -2
- D) El bucle nunca terminaría

Pregunta 8:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
lst1 = [1,2,3,4]
lst2 = lst1[-3:-1]
lst2[0] = -lst2[1]
print(lst1, lst2)
```

- A) [-3, 3] [-3, 3]
- B) [1, 2, 3, 4] [3, 3]
- C) [1, 2, 3, 4] [2, 3]
- D) [1, 2, 3, 4] [3, 3, 4]
- E) [1, 2, 3, 4] [-3, 3]
- F) Lanzará una excepción

Pregunta 9:

Un constructor de una clase superclase: (una o varias opciones)

- A) Puede devolver cualquier valor con la sentencia return.
- B) No puede ser invocado directamente dentro de la clase.
- C) Puede ser invocado directamente por cualquiera de sus subclases.
- D) Puede ser invocado directamente por cualquiera de sus superclases.

Pregunta 10:

Dada la siguiente función, selecciona la línea o líneas de código que al ejecutarse posteriormente no terminarán en excepción: (una o varias opciones)

```
def mi_funcion(a, b=0):  
    c = a * b  
  
# Línea de código
```

- A) x = mi_funcion(b=1)
- B) mi_funcion(b=1)
- C) x = mi_funcion(a=0)
- D) mi_funcion(a=0)
- E) x = mi_funcion(b=1, 0)
- F) mi_funcion(b=1, 0)
- G) x = mi_funcion(1)
- H) mi_funcion(1)

Pregunta 11:

Si necesitas leer un sólo carácter desde un stream llamado s, ¿qué instrucción utilizarías? (1 opción)

- A) ch = read (s, 1)
- B) ch= s.input (1)
- C) ch= input (s, 1)
- D) ch= s.read (1)

Pregunta 12:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
class A:
    def a(self):
        print("A", end='')
    def b(self):
        self.a()

class B(A):
    def a(self):
        print("B", end='')
    def do(self):
        self.b()

class C(A):
    def a(self):
        print("C", end='')
    def do(self):
        self.b()

B().do()
C().do()
```

- A) BB
- B) CC
- C) AA
- D) BC

Pregunta 13:

¿Cuántos asteriscos se mostrarán por pantalla al ejecutar el siguiente código? (1 opción)

```
s = '*****'
s = s - s[1:2]
print(s)
```

- A) Cinco
- B) Ninguno, el código dará error
- C) Cuatro
- D) Dos

Pregunta 14:

¿Cuál o cuáles de las siguientes acciones se puede realizar después de declarar la variable `mi_variable` de la siguiente forma? (una o varias opciones)

```
mi_variable = ()
```

- A) `mi_variable[:]`
- B) `mi_variable.append(0)`
- C) `mi_variable[0]`
- D) `mi_variable[0] = 5`
- E) `del mi_variable`

Pregunta 15:

¿Qué se puede deducir de la siguiente línea? (una o varias)

```
x = a.b.c.f()
```

- A) Que antes de dicha línea hace falta poner `import a.b.c`
- B) Que `f()` se define en un subpaquete `c` de un subpaquete `b` de un paquete `a`
- C) Que antes de dicha línea hace falta poner `from a.b.c import f`
- D) La línea es incorrecta
- E) El nombre de la función que se está invocando se llama `a.b.c.f()`

Pregunta 16:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
lst = [x + 1 for x in range(5) if x < 3]
lst = list(map(lambda x: x ** 1 ** 2, lst))
print(lst)
```

- A) [1, 4, 9]
- B) [0, 1, 4]
- C) [1, 2, 3]
- D) [1, 1, 2]

Pregunta 17:

¿Cuántos asteriscos se van a mostrar por pantalla al ejecutar el siguiente código? (1 opción)

```
def mi_funcion(x):
    if x <= 0:
        print("*", end='')
    elif x >= 2:
        print("*" * 2, end='')
    else:
        print("*" * 1, end='')

for i in range(1:4):
    mi_funcion(i)
```

- A) 4
- B) 5
- C) 3
- D) El código dará error

Pregunta 18:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
x = "Maria"
y = "maria"
if x == y:
    print("IGUAL")
elif x < y:
    print("MENOR")
else:
    print("MAYOR")
```

- A) MENOR
- B) IGUAL
- C) MAYOR
- D) No mostrará nada

Pregunta 19:

¿Cuántos asteriscos se van a mostrar por pantalla al ejecutar el siguiente código? (1 opción)

```
def o(p):
    def q(a):
        return '*' + '*' * p
    return q

r = o(2)
s = o(3)
try:
    print(r(2), end='')
except:
    pass
else:
    print('*', end='')

try:
    print(s(), end='')
except:
    pass
else:
    print('*', end='')
```

- A) No mostrará nada
- B) tres
- C) cuatro
- D) cinco
- E) nueve

Pregunta 20:

¿Cuál será la salida por pantalla al ejecutar el siguiente código? (1 opción)

```
class I:
    def __init__(self):
        self.s = 'abc'
        self.i = 0

    def __iter__(self):
        return self

    def __next__(self):
        if self.i == len(self.s):
            raise StopIteration
        v = self.s[self.i]
        self.i += 1
        return v

for x in I():
    print(x, end='')
```

- A) cba
- B) 210
- C) 012
- D) abc