

IADES INSTITUTO ARGENTINO DE EDUCACIÓN SECUNDARIA Y SUPERIOR

Carrera Analista de Sistemas - 2do Año 1er Cuatrimestre
Parcial de Taller de Programación II
Temas: Fundamentos de programación en Python y Git / Github
Prof. Federico I. Poliseno

18 de junio de 2024

NOMBRE Y APELLIDO:
DNI:

1.

A. ¿Qué es Git y para qué se utiliza?

B. Describe brevemente lo que hacen los siguientes comandos de Git:

git init:
git commit:
git add:
git clone:

C. Describe brevemente "los tres estados" y cuál sería el flujo de trabajo básico en Git.

2. Define PEP 8 y describe al menos dos de sus recomendaciones.

--

3. Dada la siguiente lista con enteros num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] definir una función evaluar_lista que reciba num como argumento y que imprima en pantalla si un número es par o impar.

¿Cómo modificarías el script para que la función nos devuelva sólo los números impares agregando sólo dos líneas de código a la función anterior?

4. Explicar qué sucede durante la ejecución del siguiente fragmento de código, detallando cada uno de los recursos aplicados:

```
inventario = {}

num_productos = int(input("Ingresa la cantidad de productos en el inventario a cargar: "))

for _ in range(num_productos):
    nombre = input("Ingresa el nombre del producto: ")
    cantidad = int(input(f"Ingresa la cantidad de {nombre}: "))
    inventario[nombre] = cantidad
```


5. Dado el diccionario 'estudiantes' donde las claves son los nombres de los estudiantes y los valores son tuplas que contienen (edad, calificación):

```
estudiantes = {
    "Juan": (21, 85),
    "Ana": (22, 92),
    "Luis": (20, 78),
    "Maria": (23, 88)
}
```

- A. Imprime el nombre del estudiante con la mayor calificación.
- B. Calcula la edad promedio de los estudiantes.

6. Importando el módulo random de Python:

- A. Generar una lista de diez números enteros aleatorios en 1 y 100. **Utilizar: randint**
- B. Ordenar la lista en orden descendente. **Utilizar: sort**
- C. Imprimir el número más bajo y el más alto **utilizando los subíndices** de la lista ordenada.

7.

A. Definir recursividad.

B. Explicar paso a paso qué sucede al invocar la siguiente función con 2 como argumento.

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)
```

8. Completar que se mostraría en pantalla al ejecutar, teniendo en cuenta que cada ítem condiciona a los siguientes:

notas = [40, 100, 60, 80, 50, 70, 60]

- A. print(len(notas))
- B. print(max(notas))
- C. print(min(notas))
- D. print(sum(notas))
- E. print(sum(notas)/len(notas))
- F. print(notas.pop())
- G. print(notas)
- H. print(notas.count(60))
- I. print(notas.index(80))
- J. print(notas[1] % notas[-1])

1	2	3	4	5	6	7	8
/10pts	/5pts	/20pts	/20pts	/10pts	/10pts	/20pts	/5pts