



Planeación de aula.

Identificación

Grado/Grupo: 11	Área/Asignatura: Informática	Fecha : 09-11-23
Docente / C.D.A.: JAIME CABALLERO		
Sede: PRINCIPAL	Periodo Académico: CUATRO	
Eje temático : Adquirir conocimientos sobre estructuras de datos: Los estudiantes deben entender las estructuras de datos básicas, como matrices, listas enlazadas, pilas y colas, utilizando el lenguaje de programación PYTHON		
Tiempo de Ejecución: 20 DIAS		

Aprendizajes

Objetivos de aprendizajes
Explorar Matplotlib y Seaborn para crear gráficos e interpretar visualmente tus datos.
Referentes curriculares (EBC, DBA, Matriz de Referencia, Mallas de Aprendizaje)
Aplico los conocimientos en proyectos reales. Puedo comenzar con conjuntos de datos pequeños y luego trabajar en proyectos más grandes a medida que adquieras experiencia.
Evidencias de Aprendizajes / Desempeños Esperados
La programación ofrece oportunidades para la creatividad y la innovación. Los estudiantes deben ser alentados a pensar de manera creativa y a buscar soluciones innovadoras a los problemas planteados.
Recursos y materiales
Se utilizará dentro del aula proyector audiovisual, sobre el cual se estará pausando frecuentemente en caso de dudas y profundizar explicaciones.



Momentos de la clase

Inicio /exploración de saberes previos

Se realiza un repaso práctico sobre la computadora de todos los comandos dados y estudiados para seguir con el proceso de aprendizaje de la programación

Contenido / Estructuración

For anidados

Es posible **anidar** los `for`, es decir, **meter uno dentro de otro**. Esto puede ser muy útil si queremos iterar algún objeto que en cada elemento, tiene a su vez otra clase iterable. Podemos tener por ejemplo, una lista de listas, una especie de matriz.

```
lista = [[56, 34, 1],  
         [12, 4, 5],  
         [9, 4, 3]]
```

Si iteramos usando sólo un `for`, estaremos realmente accediendo a la segunda lista, pero no a los elementos individuales.

```
for i in lista:  
    print(i)  
#[56, 34, 1]  
#[12, 4, 5]  
#[9, 4, 3]
```

Si queremos acceder a cada elemento individualmente, podemos anidar dos `for`. Uno de ellos se encargará de iterar las columnas y el otro las filas.

```
for i in lista:  
    for j in i:  
        print(j)  
# Salida: 56,34,1,12,4,5,9,4,3
```

Ejemplos for

Iterando cadena al revés. Haciendo uso de `[::-1]` se puede iterar la lista desde el último al primer elemento.

```
texto = "Python"  
for i in texto[::-1]:  
    print(i) #n,o,h,t,y,P
```

Itera la cadena saltándose elementos. Con `[::-2]` vamos tomando un



elemento si y otro no.

```
texto = "Python"
for i in texto[::-2]:
    print(i) #P,t,o
```

Un ejemplo de `for` usado con **comprehensions lists**.

```
print(sum(i for i in range(10)))
# Salida: 45
```

Práctica / Transferencia

Vamos a incrementar el desarrollo de un aplicativo que diseñamos en conjunto, esta vez le agregaremos los ciclos WHILE y esperaremos la evolución del algoritmo

Descripción de la Evaluación y Valoración/cierre

Desde de sus hogares en los celulares o pc, para los que tengan, de lo contrario llevará escrito el algoritmo en sus libretas interiorizaran el contenido de tal manera que se haga suyo las estructura del mismo.