

Repaso de unidades desde la 1 a la 4

Lcdo. Diego Medardo Saavedra García. Mgtr.

2023-08-31

Section 1

Clase de Repaso: Unidades 1-4

Objetivos

- Repasar los conceptos fundamentales de las Unidades 1 a 4.
- Reforzar la comprensión de la instalación de herramientas, tipos de datos, variables y estructuras de control en Python.

Clase de Repaso: Unidad 1 - Introducción a la Programación

Objetivos

- Repasar los conceptos fundamentales de la programación.
- Familiarizarse con las diferentes formas de trabajar con Python.

Conceptos Clave

¿Qué es la programación y por qué es importante?

- La programación es el proceso de crear instrucciones para que una computadora realice una tarea específica. Es importante porque nos permite automatizar tareas, resolver problemas de manera eficiente y crear aplicaciones y sistemas.
- Diferentes formas de trabajar con Python
- Python se puede utilizar de varias formas, incluyendo:

Interactivo: Utilizando el intérprete de Python para escribir y ejecutar comandos.

Scripting: Escribiendo scripts de Python para realizar tareas específicas.

Desarrollo de aplicaciones: Creando programas completos utilizando Python.

Ejercicios Prácticos

Ejercicio 1: Crear un Saludo Personalizado

```
# Solicita al usuario su nombre
nombre = input("Ingresa tu nombre: ")

# Imprime un mensaje personalizado
print(";Hola,", nombre + "! Bienvenido a la clase de repaso.")
```


Ejercicio 2: Obtener Información del Usuario

```
# Solicita al usuario su nombre y edad
nombre = input("Ingresa tu nombre: ")
edad = input("Ingresa tu edad: ")

# Imprime un mensaje con la información del usuario
print(";Hola,", nombre + "! Tienes", edad, "años.")
```

Actividad Práctica

- Desarrolla un programa interactivo que solicite al usuario su nombre, edad y ciudad de residencia. Luego, imprime un mensaje personalizado que incluya toda la información recopilada.

Clase de Repaso: Unidad 2 - Instalación de Python y más herramientas

Objetivos

- Repasar el proceso de instalación de Python y otras herramientas esenciales.
- Familiarizarse con la configuración inicial de Git y la creación del primer “Hola, mundo” en Python.

Conceptos Clave

- Instalación de Python, Git y VS Code
- Instalar Python desde python.org.
- Instalar Git desde git-scm.com.
- Instalar Visual Studio Code desde code.visualstudio.com.

Configuración Inicial de Git

- Configurar nombre de usuario y dirección de correo electrónico con git config.

Ejercicios Prácticos

Ejercicio 1: Instalar Python y Comprobar Versión

- Abre tu terminal o línea de comandos y ejecuta los siguientes comandos:

```
python --version
```


Ejercicio 2: Instalar Git y Configurar Usuario

```
git --version  
git config --global user.name "Tu Nombre"  
git config --global user.email "tu@email.com"
```

Ejercicio 3: Primer “Hola, mundo” en Python

- Crea un archivo llamado `hola_mundo.py` y escribe el siguiente código:

```
print(";Hola, mundo!")
```

Luego, ejecuta el archivo con:

```
python hola_mundo.py
```

Actividad Práctica

- Desarrolla un programa que solicite al usuario su nombre y luego lo almacene en un archivo de texto llamado usuarios.txt. Utiliza la función `open()` para crear y escribir en el archivo.

Clase de Repaso: Unidad 3 - Introducción a Python

Objetivos

- Repasar los conceptos fundamentales de la sintaxis de Python.
- Reforzar el uso de variables, comentarios y operaciones básicas en Python.

Conceptos Clave

Indentación y Sintaxis

- La indentación es crucial en Python para definir bloques de código.
- Los bloques se crean con dos puntos (:) y se indentan con espacios o tabulaciones.

Variables y Asignación

- Las variables almacenan datos en la memoria.
- Se asigna un valor a una variable utilizando el operador de asignación =.

Comentarios

- Los comentarios son notas explicativas en el código que no se ejecutan.
- Se inician con el símbolo #.

Operaciones Básicas

- Python permite realizar operaciones aritméticas básicas como suma, resta, multiplicación y división.

Ejercicios Prácticos

Ejercicio 1: Operaciones Básicas

- Escribe un programa que solicite al usuario dos números y luego imprima la suma, resta, multiplicación y división de esos números.


```
num1 = float(input("Ingresa el primer número: "))
num2 = float(input("Ingresa el segundo número: "))

suma = num1 + num2
resta = num1 - num2
multiplicacion = num1 * num2
division = num1 / num2

print("Suma:", suma)
print("Resta:", resta)
print("Multiplicación:", multiplicacion)
print("División:", division)
```

Ejercicio 2: Conversión de Temperatura

- Escribe un programa que convierta una temperatura en grados Celsius a grados Fahrenheit.

```
celsius = float(input("Ingresa la temperatura en grados Celsius:"))  
fahrenheit = (celsius * 9/5) + 32  
print("La temperatura en grados Fahrenheit es:", fahrenheit)
```

Actividad Práctica

- Desarrolla un programa que calcule el área de un triángulo utilizando la fórmula $(\text{base} * \text{altura}) / 2$. Solicita al usuario los valores de la base y la altura.

Clase de Repaso: Unidad 4 - Tipos de Datos

Objetivos

- Repasar los conceptos fundamentales de los tipos de datos en Python.
- Reforzar el uso de strings, números, listas y diccionarios.

Conceptos Clave

Tipos de Datos

- Python tiene varios tipos de datos incorporados, como strings, números y listas.
- Cada tipo de dato tiene sus propiedades y métodos específicos.

Strings y Números

- Los strings son secuencias de caracteres y se pueden manipular con operaciones de cadena.
- Python admite números enteros y números de punto flotante para operaciones matemáticas.

Listas y Diccionarios

- Las listas son colecciones ordenadas de elementos y se definen con corchetes `[]`.
- Los diccionarios son estructuras clave-valor y se definen con llaves `{}`.

Ejercicios Prácticos

Ejercicio 1: Manipulación de Strings

- Escribe un programa que tome un string como entrada y luego imprima el string en mayúsculas y en minúsculas.


```
texto = input("Ingresa un texto: ")
mayusculas = texto.upper()
minusculas = texto.lower()

print("En mayúsculas:", mayusculas)
print("En minúsculas:", minusculas)
```

Ejercicio 2: Listas y Diccionarios

- Crea una lista de nombres y un diccionario que contenga información sobre una persona (nombre, edad, ciudad). Luego, imprime los nombres de la lista y la información del diccionario.

```
nombres = ["Ana", "Juan", "María"]  
persona = {"nombre": "Juan", "edad": 25, "ciudad": "Madrid"}  
  
print("Nombres:", nombres)  
print("Información de la persona:", persona)
```

Actividad Práctica

- Desarrolla un programa que simule una lista de compras. Permite al usuario agregar elementos a la lista y luego imprimir la lista completa.

Ejercicio Práctico: Calculadora Interactiva

- Desarrolla una calculadora interactiva en Python que permita al usuario realizar operaciones aritméticas simples. La calculadora debe ser capaz de realizar sumas, restas, multiplicaciones y divisiones. Requisitos
- El programa debe presentar un menú con las opciones de operaciones: suma, resta, multiplicación y división.
- El usuario debe seleccionar una opción ingresando el número correspondiente.
- Luego, el programa debe solicitar al usuario los números para realizar la operación.
- Finalmente, el programa debe mostrar el resultado de la operación.

Ejemplo

Calculadora Interactiva

1. Suma
2. Resta
3. Multiplicación
4. División

Selecciona una opción: 2

Ingresa el primer número: 15

Ingresa el segundo número: 7

El resultado de la resta es: 8

Sugerencias

- Utiliza una estructura de control if o elif para determinar qué operación realizar en función de la opción seleccionada.
- Puedes utilizar las funciones `float()` para convertir las entradas del usuario en números de punto flotante.
- No olvides manejar el caso de la división por cero.

Actividad Práctica

- Desarrolla el programa de la calculadora interactiva y ejecútalo para probar su funcionamiento.

Ejercicio Práctico: Calculadora Interactiva

Explicación de la Resolución:

- El programa comienza mostrando un menú de opciones al usuario, que incluye “Suma”, “Resta”, “Multiplicación” y “División”.
- El usuario selecciona una opción ingresando el número correspondiente.
- Luego, el programa solicita al usuario dos números para realizar la operación.
- Utiliza estructuras de control if y elif para determinar qué operación realizar según la opción seleccionada.
- Si la opción es “División”, el programa verifica si el segundo número es cero para evitar la división por cero.
- Finalmente, se muestra el resultado de la operación. Si el resultado es un número de punto flotante, se formatea para mostrar solo dos decimales.