

ALGORITMIA

Juan Carlos Molina Lozano
Docente

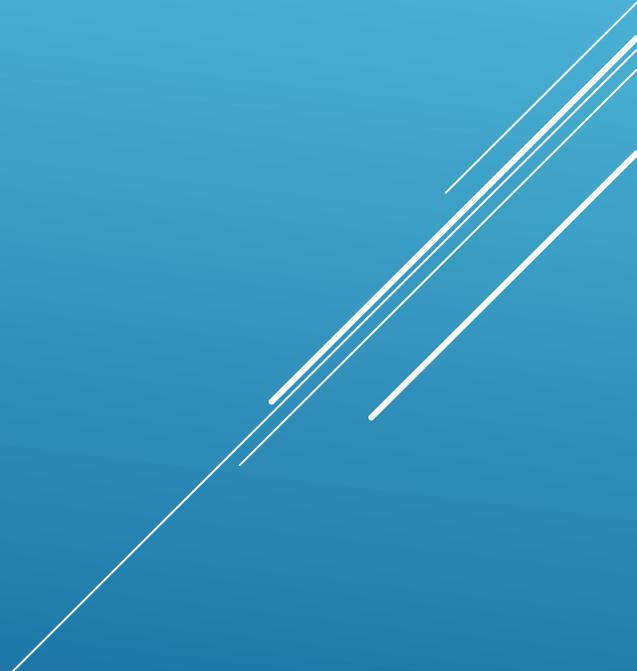
OBJETIVOS DE LA CLASE

1. Comprender la importancia de Python y sus aplicaciones en diferentes campos.
 2. Aprender la sintaxis básica del lenguaje y sus principales estructuras.
 3. Desarrollar habilidades para escribir programas en Python aplicando estructuras de control y condicionales.
 4. Fomentar la resolución de problemas mediante ejercicios prácticos y proyectos sencillos.
 5. Introducir la importancia de buenas prácticas en programación.
- 

METODOLOGÍA

1. Se aplicará un enfoque constructivista donde los estudiantes aprenderán a través de la exploración y la resolución de problemas.
2. Se fomentará el aprendizaje activo mediante el desarrollo de ejercicios prácticos y proyectos sencillos.
3. Se utilizarán ejemplos del mundo real para contextualizar el uso de Python.
4. Se promoverá el trabajo colaborativo a través de discusiones y resolución de problemas en grupo.
5. Se incentivará la autoevaluación y el aprendizaje basado en errores para fortalecer la comprensión del lenguaje.

CONTENIDO

1. Introducción a Python.
 2. Características de Python.
 3. IDE (Integrated Development Environment) para Python.
 4. Sintaxis.
 5. Comandos.
 6. Conclusiones.
 7. Ejercicios.
- 

INTRODUCCIÓN A PYTHON

- Python es un lenguaje de programación interpretado, de alto nivel y con tipado dinámico.
 - Creado por Guido van Rossum en 1991.
 - Se destaca por su sintaxis simple y legible, lo que lo hace ideal para principiantes y expertos.
 - Usado en desarrollo web, ciencia de datos, inteligencia artificial, automatización de procesos, ciberseguridad, videojuegos y más.
- 

CARACTERISTICAS DE PYTHON

- Diseñado para ser fácil de aprender y usar, inspirado en el lenguaje ABC.
 - Compatible con múltiples plataformas (Windows, MacOS, Linux, móviles y web).
 - Gran comunidad y extensa librería estándar que facilita el desarrollo.
 - Lenguaje de código abierto con mejoras constantes.
 - Permite integración con otros lenguajes como C, C++, Java, y .NET.
- 

VENTAJAS DE PYTHON

- **Sintaxis clara y legible:** La sintaxis de Python es simple y fácil de entender, lo que acelera el desarrollo y facilita la colaboración entre equipos.
- **Gran comunidad y soporte:** Python cuenta con una comunidad activa de desarrolladores que contribuyen con bibliotecas, tutoriales y resuelven dudas en foros. Esto proporciona un valioso respaldo y facilita el aprendizaje y la resolución de problemas.
- **Amplia biblioteca estándar:** Python incluye una biblioteca estándar extensa que cubre diversas áreas, desde manipulación de archivos hasta desarrollo web.
- **Versatilidad y portabilidad:** Python es un lenguaje multiplataforma, lo que significa que el código escrito en Python puede ejecutarse en diferentes sistemas operativos sin modificaciones significativas.
- **Desarrollo rápido de prototipos:** La facilidad de escritura y la sintaxis concisa hacen que Python sea ideal para el desarrollo rápido de prototipos.
- **Integración fácil con otros lenguajes:** Python puede integrarse fácilmente con otros lenguajes, como C y C++.

DESVENTAJAS DE PYTHON

- Velocidad de ejecución: Python, al ser un lenguaje interpretado, puede ser más lento en comparación con lenguajes compilados, como C++ o Java, esta es una de las mayores desventajas del lenguaje python. Esto puede ser una limitación en aplicaciones que requieren un rendimiento extremadamente rápido, como desarrollo de juegos o cómputo intensivo.
- Gestión de memoria automática: Aunque la gestión automática de memoria es una ventaja para muchos desarrolladores, puede ser una desventaja en aplicaciones que requieren un control preciso de los recursos de memoria.
- No es ideal para desarrollo móvil: Aunque existen frameworks como Kivy o BeeWare, Python no es considerado el lenguaje principal para el desarrollo móvil.
- Problemas de comprensión asíncrona: Aunque Python ha mejorado en la gestión de operaciones asíncronas, algunos desarrolladores aún encuentran desafíos al trabajar con código asíncrono, especialmente en comparación con lenguajes diseñados específicamente para manejar tareas concurrentes.
- Interpretación y ejecución: El hecho de que Python sea interpretado puede ser una desventaja en términos de velocidad de ejecución en comparación con lenguajes compilados. Sin embargo, esta desventaja se ve mitigada por las implementaciones Just-In-Time (JIT) como PyPy.
- Problemas de integración con C/C++: Aunque Python se puede integrar con C y C++, puede haber desafíos y complejidades asociadas con la interacción entre estos lenguajes, especialmente en proyectos más grandes.

IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) PARA PYTHON

Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Algunos IDE recomendados para programar en Python:

- Visual Studio Code (VS Code) – Ligero, personalizable y con muchas extensiones útiles para Python.
- PyCharm – IDE profesional con herramientas avanzadas para desarrollo en Python.
- Jupyter Notebook – Ideal para análisis de datos y aprendizaje interactivo.
- IDLE – Sencillo y viene por defecto con la instalación de Python.
- Thonny – Recomendado para principiantes, con una interfaz intuitiva.
- Spyder – Orientado a ciencia de datos y análisis numérico.
- Google Colab

SINTAXIS BÁSICA DE PYTHON

- Variables y tipos de datos: **int, float, str, bool, list, dict, tuple, set**.
- Operadores: aritméticos (**+, -, *, /, %, //, ****), lógicos (**and, or, not**), comparación (**==, !=, <, >, <=, >=**).
- Estructuras de control: condicionales (**if, elif, else**), bucles (**for, while**).
- Definición de funciones: uso de **def nombre_funcion():**.

Ejemplo:

```
nombre = "Juan"  
edad = 25  
print(f"Hola, mi nombre es {nombre} y tengo {edad} años.")
```

COMANDOS Y CARACTERÍSTICAS CLAVES

Entrada y salida de datos:

```
nombre = input("Introduce tu nombre: ")
print("Hola, " + nombre)
```

Manipulación de listas y diccionarios:

```
lista = [1, 2, 3, 4]
lista.append(5)
print(lista)
```

Uso de módulos:

```
import math
print(math.sqrt(16))
```

Excepciones:

```
try:
    x = int(input("Introduce un número: "))
    print(10 / x)
except ZeroDivisionError:
    print("No se puede dividir por cero")
```

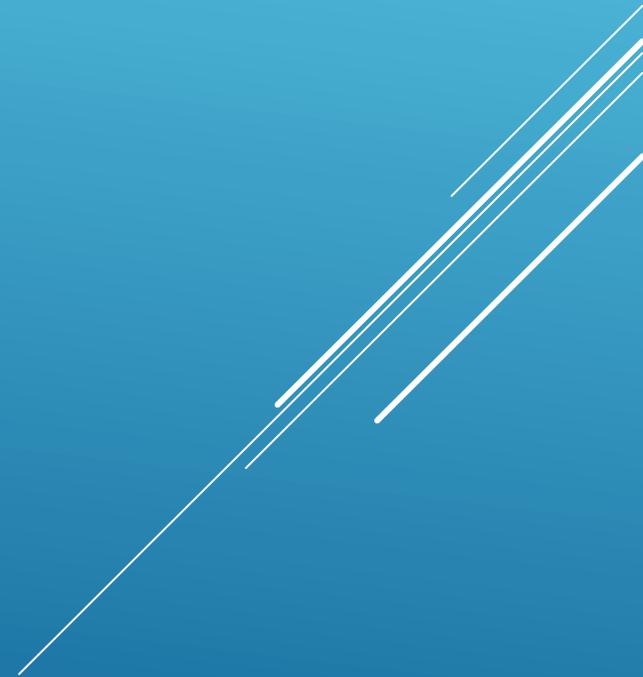
CONCLUSIONES

Python es un lenguaje versátil y poderoso que facilita el aprendizaje de la programación.

Aplicaciones en diversas áreas del conocimiento.

Es importante practicar con ejercicios y proyectos reales.

Preguntas y comentarios.



EJERCICIOS

1. Determinar si un número es positivo, negativo
2. Determine un programa para saber si una persona es mayor de edad o no.
3. Desarrollar un programa que permita si un estudiante aprobó la nota o no.
4. Número par o impar: Escribe un programa que pida un número y determine si es par o impar.
5. Calculadora de descuentos: Si el total de la compra es mayor a \$100, aplicar un 10% de descuento y mostrar el total final.
6. Determinar el mayor de dos números: Pide al usuario dos números y muestra cual es el mayor.
7. Escribir un programa que pida al usuario su nombre y lo salude con un mensaje personalizado.
8. Solicitar dos números al usuario y mostrar su suma.
9. Escribir un programa que reciba tres números y determine cuál es el mayor de los tres.

RESUMEN

1. **Introducción a Python.**
 2. **Características de Python.**
 3. **IDE (Integrated Development Environment) para Python.**
 4. **Sintaxis.**
 5. **Comandos.**
 6. **Conclusiones.**
 7. **Ejercicios.**
- 