

Plan de aprendizaje personalizado para aprender a programar en Python para desarrollar aplicaciones que usen inteligencia artificial y creen automatismos de mejora en negocios.

Componente 1: Introducción a Python

Descripción del contenido a aprender:

En este componente aprenderás los conceptos básicos de Python, como las variables, los tipos de datos, las operaciones, las estructuras de control, las funciones y los módulos. También aprenderás a usar el entorno de desarrollo integrado (IDE) de Python y a ejecutar tus primeros programas.

Estimación del esfuerzo de tiempo requerido:

Este componente te llevará aproximadamente 10 horas de estudio.

Cinco recursos recomendados para adquirir esta habilidad:

- Curso interactivo de [Python en Codecademy](#)
- Tutorial de [Python en W3Schools](#)
- Libro “Python para todos” de Raúl González Duque
- Vídeo “[Python Tutorial for Beginners](#)” de Programming with Mosh
- Vídeo “[Aprende Python desde cero](#)” de Platzi

Tres ejercicios prácticos para aplicar y aprender la destreza:

- Escribe un programa que pida al usuario su nombre y edad y le salude.
- Escribe un programa que calcule el área de un círculo dado su radio.
- Escribe un programa que pida al usuario dos números y muestre el resultado de sumarlos, restarlos, multiplicarlos y dividirlos.

Componente 2: Estructuras de datos y algoritmos en Python

Descripción del contenido a aprender:

En este componente aprenderás a usar las estructuras de datos más comunes en Python, como las listas, las tuplas, los diccionarios y los conjuntos. También aprenderás a implementar algunos algoritmos básicos, como la búsqueda binaria, la ordenación por selección y el recorrido de árboles.

Estimación del esfuerzo de tiempo requerido:

Este componente te llevará aproximadamente 15 horas de estudio.

Cinco recursos recomendados para adquirir esta habilidad:

- Curso interactivo de [Estructuras de Datos en Python en Codecademy](#)
- Libro “[Problem Solving with Algorithms and Data Structures using Python](#)” de Bradley N. Miller y David L. Ranum
- Vídeo “[Data Structures and Algorithms in Python](#)” de CS Dojo
- Vídeo “[Estructuras de Datos en Python](#)” de Platzi
- Artículo “[Python Data Structures Tutorial](#)” de DataCamp

Componente 3: Programación orientada a objetos en Python

Descripción del contenido a aprender:

En este componente aprenderás los principios de la programación orientada a objetos (POO) en Python, como las clases, los objetos, los métodos, los atributos, la herencia, el polimorfismo y la encapsulación. También aprenderás a diseñar y modelar tus propias clases y a usar algunas de las clases más útiles del módulo estándar de Python.

Estimación del esfuerzo de tiempo requerido:

Este componente te llevará aproximadamente 10 horas de estudio.

Cinco recursos recomendados para adquirir esta habilidad:

- Curso interactivo de [POO en Python en Codecademy](#)
- Libro “Python 3 [Object-Oriented Programming](#)” de Dusty Phillips
- Vídeo “[Object-oriented Programming in Python](#)” de Programming with Mosh
- Vídeo “[Programación Orientada a Objetos en Python](#)” de Platzi
- Artículo “[Object Oriented Programming in Python](#)” de Real Python

Tres ejercicios prácticos para aplicar y aprender la destreza:

- Escribe una clase llamada Persona que tenga como atributos el nombre, la edad y el género. Define un método llamado saludar que imprima un mensaje de bienvenida con el nombre de la persona. Crea dos objetos de la clase Persona y haz que se saluden.
- Escribe una clase llamada Animal que tenga como atributos el nombre, el tipo y el sonido. Define un método llamado hacer_sonido que imprima el sonido del animal. Escribe dos clases hijas de Animal llamadas Perro y Gato que hereden los atributos y el método de la clase padre y que añadan un atributo propio llamado raza. Crea un objeto de cada clase hija y haz que hagan sonido.
- Escribe una clase llamada Calculadora que tenga como atributo un número. Define cuatro métodos llamados sumar, restar, multiplicar y dividir que reciban otro número como parámetro y que devuelvan el resultado de la operación correspondiente con el atributo. Crea un objeto de la clase Calculadora e invoca los cuatro métodos con diferentes números.

Componente 4: Introducción a la inteligencia artificial con Python

Descripción del contenido a aprender:

En este componente aprenderás los conceptos fundamentales de la inteligencia artificial (IA), como las definiciones, las aplicaciones, los tipos, los agentes y los problemas. También aprenderás a usar algunas de las librerías más populares de Python para IA, como NumPy, Pandas, Matplotlib, Scikit-learn y TensorFlow.

Estimación del esfuerzo de tiempo requerido:

Este componente te llevará aproximadamente 20 horas de estudio.

Cinco recursos recomendados para adquirir esta habilidad:

- Curso interactivo de [IA con Python en Codecademy](#)
- Libro “[Artificial Intelligence with Python](#)” de Prateek Joshi
- Vídeo “[Artificial Intelligence Tutorial for Beginners](#)” de Simplilearn
- Vídeo “[Introducción a la Inteligencia Artificial con Python](#)” de Platzi

Componente 5: Desarrollo de aplicaciones con Python

Descripción del contenido a aprender:

En este componente aprenderás a desarrollar aplicaciones con Python que usen inteligencia artificial y creen automatismos de mejora en negocios. Aprenderás a usar frameworks como Flask y Django para crear interfaces web, a usar bases de datos como SQLite y MongoDB para almacenar y manipular datos, y a usar APIs como Google Cloud Vision y IBM Watson para integrar servicios de IA en tus aplicaciones.

Estimación del esfuerzo de tiempo requerido:

Este componente te llevará aproximadamente 25 horas de estudio.

Cinco recursos recomendados para adquirir esta habilidad:

- Curso interactivo de [Desarrollo Web con Python en Codecademy](#)
- Libro [“Python Web Development with Django”](#) de Jeff Forcier, Paul Bissex y Wesley Chun
- Vídeo [“Build a Python App that Tracks Amazon Prices”](#) de Dev Ed
- Vídeo [“Desarrollo Web con Python y Django”](#) de Platzi
- Artículo [“How to Build a Face Recognition App in Python”](#) de DataCamp

Tres ejercicios prácticos para aplicar y aprender la destreza:

- Crea una aplicación web con Flask que muestre un formulario para introducir el nombre de un producto y que devuelva el precio más bajo encontrado en Amazon. Usa la librería BeautifulSoup para hacer web scraping.
- Crea una aplicación web con Django que permita registrar y consultar los datos de los clientes de una empresa. Usa la librería SQLite para crear una base de datos relacional.
- Crea una aplicación web con Flask que permita subir una imagen y que devuelva el texto que contiene. Usa la API de Google Cloud Vision para hacer reconocimiento óptico de caracteres.

Este es el final del plan de aprendizaje personalizado que he creado para ti. Espero que te haya gustado y que te sirva para alcanzar tu objetivo. Te deseo mucho éxito y diversión en tu camino. 👍