educación IT
CODE YOUR FUTURE

Introducción a Python

Módulo 4

educación IT
CODE YOUR FUTURE

Breve introducción al Paradigma de Objetos

Paradigmas

Todo paradigma decide plantear algún nuevo **concepto abstracto** de forma de pensar o resolver un problema.

Hace mucho tiempo, se estudiaba la computación como una rama de la matemática y se especializaba en ello. Por eso el concepto de paradigma tiene que ver con el pensamiento científico aplicado a las ciencias duras.

Existen varios paradigmas, y con ellos tenemos ventajas a la hora de escribir código. En los programas pueden interactuar varios al unísono. No es obligatorio hacer un desarrollo solo con con uno, o con varios, eso depende del programador.

Con el tiempo uno no se fija que paradigma está aplicando para resolver un problema, se convierte en algo inherente. Pero siempre es bueno conocerlos y entender cómo funcionan.

En este curso, solo los nombramos, pero a medida que avances en tu estudio, veras esto reflejado en código y lenguaje.

Tipos de paradigmas

- **Paradigma Imperativo**
  Los programas se componen de una secuencia claramente definida de instrucciones que se **ejecutan en orden.**
- **Paradigma Declarativo**
  Se piden los resultados esperados **sin mostrar explícitamente** los pasos para alcanzarlos.
- **Paradigma Lógico**
  La mayoría de los lenguajes de programación se basan en la teoría lógica de primer orden. La idea sería **"aplicar reglas lógicas para inferir“.**
- **Paradigma Funcional**
  En este paradigma las **funciones pueden enlazarse entre ellas, utilizarse como parámetro de otra función.** Esto no solo permite reutilizar código, sino que al enlazar una función con otra, puede generar una más compleja.

Un paradigma más: POO

El paradigma de **programación orientada a objetos (POO)** plantea que todo sistema o proceso informático puede modelarse (pensarlo) como "objetos" cotidianos. Es decir, programar bajo este paradigma, implica que nuestros programas, deberán ser pensados de forma especial y diferente.

Los objetos con características similares, se dice que fueron creados por la misma clase (por la misma fábrica, para traerlo a algo más tangible).

Los objetos son entidades que combinan características (variables) y métodos (funciones).

Las fábricas que crean estos objetos pueden ser desarrolladas por el programador o emplear ya existentes.

Ejemplo POO

Este es un ejemplo para comprender simbólicamente este paradigma.

Resulta que tenemos una *fábrica* de pelotas que se usa para crear una "pelota", que son los *objetos* del tipo que se fabrican en ese lugar. Entonces nuestra pelota se dice que es una "*instancia de la fábrica de pelotas*".

Esta pelota tiene ciertos atributos, un color (variable color), un diámetro (variable diámetro), etc. Pero no solo tiene atributos nuestra pelota, también tiene "métodos" (funciones).

Nuestra pelota tiene el método "picar", con el cual puede brincar.

La fábrica no solo crea nuestra pelota, la que charlamos en el ejemplo anterior. Puede fabricar otras pelotas con otros atributos (otro color, otro diámetro, etc). Pero por lo general con los mismos métodos. Las pelotas no son todas iguales (atributos diferentes), pero por lo general, todas las pelotas pican.

Usaremos POO para la aplicación de escritorio

Vamos a usar el paradigma de orientación a objetos para poder **armar aplicaciones de escritorio.**

Esto no quiere decir que solo se use este paradigma para crear aplicaciones gráficas. Muchas veces se suele confundir o asociar POO solo para esto.

Este paradigma se utiliza en muchas otras situaciones en programación, y muchas veces

las mismas no tienen nada que ver con la creación de aplicaciones de escritorio.

Hacemos esta aclaración para que lo tengan en cuenta y no se genere una idea equivocada.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!