Tabla

Descripción generada automáticamente con confianza baja

**Que es Python?**

Python es un lenguaje de programacion creado por Guido van Rossum a principios de los años 90 cuyo nombre esta inspirado en el grupo de comicos ingleses "Monty Python".

**Quien es Guido Van Rossum?**

Van Rossum nacio y crecio en los paises bajos, donde obtuvo su titulacion universitaria en matematicas y computacion por la Universidad de Amsterdam en 1982.

Creador de Python

Trabajo en

Google (Mondrian)

Dropbox

Microsoft

**Como ocurre la idea de Python?**

"Hace seis años, en diciembre de 1989, etaba buscando un proyecto de programacion como hobby que em mantuviera ocupado durante las semanas de navidad. Mi oficina estaria cerrada y no tendria mas que mi ordenador de casa a mano. Decidi escribir un interprete para el nuevo lenguaje de scripting que habia estado ideando recientemente: un descendiente de ABC que gustaria a los hackers de Unix-c. Elegi el nombre de Python para el pryecto, encontrandome en un estado de animo ligeramente irreverente[

" Me gussto la idea de un lenguaje de programacion que se llama como una serpiente sinuosa y poderosa"

**Que es Python como lenguaje de programación?**

Python es un lenguaje de programación de alto nivel, interpretado multiparadigma, de código abierto, Es uno de los lenguajes de programación mas populares del mundo, utilizado en una amplia gama de aplicaciones, desde el desarrollo web hasta la ciencia de datos.

Python se caracteriza por su sintaxis limpia y concisa, que lo hace fácil de aprender y usar. También es un lenguaje eficiente y flexible. Que permite a los programadores escribir código que sea tanto eficaz como creativo.

**Principales características de Python.**

* Alto nivel: Lo mas cercano al lenguaje humano.
* Interpretado: Se ejecuta directamente en la maquina virtual de Python y no tiene que crear un archivo ejecutable.
* Multiparadigma: imperativa o estructurada, POO y Funcional.
* Codigo abierto: Codigo fuente disponible para editar.
* Fuertemente tipado: Tipo de dato definido y no se mezcla de manera automática.

**Principales áreas de uso de Python.**

* Desarrollo web.
* Ciencia de datos.
* Machine learning.
* Robótica.
* Administración de sistemas.
  + Auditoria informática.

Tipos de datos y operadores en Python.

Numero enteros.

* Son aquellos numero spositivos o negativos que no tienen decimales.
* En Python se pueden representar mediante el tipo de dato int (entero, integer).
* El tipo de dato int puede manejar números de -2,147,483,648 a 2,147,483,647 en 32 bits.
* El tipo de dato long int en 64 bits el rango es mucho mayor.

Números reales.

* Los números reales son los que tienen decimales. En Python se expresan mediante el tipo float.
* En otros lenguajes tenemos double como en c#.
* Este formato utiliza 64 bits para almacenar cada numero, lo que significa que los números de tipo float en Python pueden representar un rango de valores muy grande (53 digitos decimales).
* En Python, 0.1 se representa como 0.100000000000000000000000055512312321369685

Operadores aritméticos

Tabla

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Booleanos.

Estos solo pueden tener dos valores: true y False. También acepta 1 o 0.

Estos valores son especialmente importantes para las expresiones condicionales y los bucles

Operadores lógicos.

Tabla

Descripción generada automáticamente

Operadores relacionales.

Los valores booleanos son además el resultado de expresiones que utilizan operadores relacionales.

Tabla

Descripción generada automáticamente

Cadenas.

Las cadenas son texto encerrado entre comillas simples o dobles, (“Cadena”) o (‘Cadena’).

También, es posible encerrar una cadena entre triples comillas (simples o dobles). De esta forma podremos escribir el texto en varias líneas.

Triple=”””Primera línea

Esto se vera en otra línea”””

Las cadenas también operadores como +, que funciona realizando una concatenación de las cadenas.

Cadena= ‘hola que’+ ‘tal’

Imprime ‘hola que tal’

Colecciones de datos en Python

Listas

* Una lista es una estructura de datos que permite almacenar y organizar datos de manera secuencial.
* Es un tipo de dato mutable (modificable).
* Las listas se definen mediante corchetes [].
* Mi\_lista = [1,2,3,’Hola’, 5.0, True]

Tuplas

* Una tupla es otra estructura de datos que se utiliza para almacenar colecciones ordenadas de elementos, al igual que las listas.
* Son inmutables.
* Para definir una tupla, se utilizan paréntesis ()
* mi\_tupla = (1,2, ‘hola’, 3.0, True)

Diccionarios

* Es una estructura de datos que permite almacenar pares clave-valor, donde cada clave debe ser única.
* Se definen utilizando llaves {} y consisten en una serie de pares clave: valor.
* Conjuntos
* Un conjunto (set) es una estructura de datos que almacena elementos únicos sin un orden especifico.
* Los conjuntos se definen utilizando llaves {} o utilizando el contructor set().
* Mi\_conjunto = set([4,5, ’mundo’, False])
* Mutables
* Aseguran de que no haya duplicados en una colección de datos.

Estructuras de control en Python

Una estructura de control en programación es un bloque de código que permite controlar el flujo de ejecución de un programa.

Que estructuras de control existen?

* If
* If-else
* Elif
* While
* For

Estructura de control condicional

* If: si se cumple una condición se ejecuta
* If-else: si se ejecuta una condición se ejecuta y si no se ejecuta algo mas
* Elif: multi condiciones

Estructura de control de repetición

* While: se utiliza para crear bucles que se ejecturan mientras una condición especifica sea verdadera.
* For se utiliza para iterar sobre una secuencia ( como una lista, tupla, cadena, etc) o un rango especifico de valores.

programación orientada a objetos.

Cosas de lo mas cotidianas como un perro o un coche pueden ser representadas con clases. Estas clases tienen diferentes características, que en el caso del perro podrían ser la edad, el nombre o la raza. Llamaremos a estas características, atributos.

Por último, pueden existir diferentes tipos de perro. Podemos tener uno que se llama Toby o el del vecino que se llama Laika. Llamaremos a estos diferentes tipos de perro objetos. Es decir, el concepto abstracto de perro es la clase, pero Toby o cualquier otro perro particular será el objeto.

Que es la poo?

La POO es un paradigma de programación que organiza el código en “Clases”. Las clases agrupan variables (“atributos”) y funciones (“métodos”) que representan entidades del mundo real.

Para que sirve la POO?

La POO facilita la organización del código, lo hace mas reutilizable y adaptable a cambios.

Que es una clase?

Una clas es una plantilla que define un conjunto de atributos y métodos que pueden ser utilizados para crear objetos.

Para crear una clase en Python, se utiliza la palabra clave class.

Que es un atributo.

Un atributo en Python es una variable que se define dentro de una clase. Los atributos se utilizan para almacenar datos que son específicos de un objeto de esa clase.

En otras palabras, los atributos son como las propiedades de un objeto. Por ejemplo, un obketo Persona podría tener atributos como nombre, edad y afición.

Que es un método?

Un método en Python es una función que se define dentro de una clase. Los métodos se utilizan para realizar acciones sobre un objeto de esa clase.

Los métodos son acciones que un objeto puede realizar.

Que es un objeto?

Un objeto es una entidad que contiene datos y comportamiento. En programación, un objeto es una instancia de una clase.

En otras palabras, los objetos son como cosas del mundo real que podemos representar en nuestro código.

Que es el encapsulamiento?

El encapsulamiento te permite controlar que partes de tu código son accesibles desde el exterior.

Se utilizan dos guiones bajos (“\_\_”) como prefijo para los nombres de atributos y métodos que se consideran privados y no deberían ser accedidos desde el exterior.

Modificadores de accesio: se pueden usar palabres clave como public, protected y private.

Los 4 pilares de la POO

1-Abstraccion

La abstracción es la capacidad de centrarse en lo importante de algo, ignorando los detalles menos relevantes.

Consiste en identificar las características esenciales de un objeto y omitir los detalles irrelevantes. Es como definir un objeto a través de sus características principales sin entrar en detalles específicos.

En programación, el termino se refiere al énfasis en el “¿Que hace? ¿Mas que en el “Como lo hace?”

2- Encapsulamiento

Es la técnica de ocultar los dertalles internos de un objeto y exponer solo lo que es necesario. Esto se logra definiendo los métodos de acceso y modificación de los datos, lo que permite proteger la integridad de los datos y l imitar el acceso externo.

3- Herencia

Permite que una clase (llamada subclase o clase derivada) herede propiedades y comportamientos de otra clase (llamada clase base o clase padre). Esto facilita la reutilización del código y la organización jerárquica de las clases.