



Curso de Python

Qué es un Paradigma?

Una forma de ver el mundo.

Qué es la Programación Orientada a Objetos?

Es un nuevo paradigma de programación , es decir una nueva forma de ver la programación.

Curso de Python

Qué es un OBJETO?

Un **objeto** es una entidad que tiene un **estado** (atributos o propiedades) y un **comportamiento** (métodos o funciones). A los **objetos** en programación les asignamos **nombres** para poder referirnos a ellos y manipularlos.





ProfesorenVideo.com

Curso de Python

Qué es la programación POO?

Es una forma de programar en la cual todo se ve como un objeto del mundo real que tiene **propiedades** y tiene un **comportamiento** (lo que el objeto es capaz de hacer, es decir su función).



Curso de Python

Cómo se crea un OBJETO en programación?

Para crear un objeto se necesita de una definir previamente una CLASE.

Qué es una CLASE?

Es una plantilla en la cual se definen los atributos y las funcionalidades de los objetos que de ella se deriven.

Curso de Python

Sintaxis para definir una CLASE en Python:

```
class NombreDeClase:
```

```
    atributo_1
```

```
    atributo_2
```

```
    ...
```

```
    atributo_n
```

```
    metodo_1()
```

```
    metodo_2()
```

```
    ....
```

```
    metodo_n()
```

Usar notación CamelCase

Ejemplo de atributos:

color , tamaño, forma, peso, etc

Ejemplo de métodos:

acelera() , frena() , come() , duerme()
estudia() , trabaja() , etc



ProfesorenVideo.com

Curso de Python

Sintaxis para declarar objetos una CLASE en Python:

(1)

```
class NombreDeClase:
```

```
    atributo_1
```

```
    atributo_2
```

```
    ...
```

```
    atributo_n
```

```
    metodo_1()
```

```
    metodo_2()
```

```
    ....
```

```
    metodo_n()
```



(2)

```
objeto_1 = NombreDeClase()
```

```
objeto_2 = NombreDeClase()
```

```
objeto_3 = NombreDeClase()
```

```
objeto_4 = NombreDeClase()
```

IMPORTANTE: El proceso de crear un objeto derivado de una clase se denomina **INSTANCIACIÓN**.



ProfesorenVideo.com

Curso de Python

Sintaxis para acceder a propiedades y métodos:

(1)

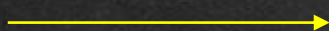
```
class NombreDeClase:
```

```
    atributo_1
```

```
    atributo_2
```

```
    metodo_1( )
```

```
    metodo_2( )
```



(2)

```
objeto_1 = NombreDeClase( )
```

```
objeto_2 = NombreDeClase( )
```

```
objeto_1.atributo_1
```

```
objeto_1.atributo_2
```

```
objeto_1.metodo_1( )
```

```
objeto_2.atributo_1
```

```
objeto_2.metodo_1( )
```

```
objeto_1.metodo_2( )
```




ProfesorenVideo.com

Curso de Python

```
class NombreDeClase:
```

```
    atributo_1
```

```
    atributo_2
```

```
    metodo_1( )
```

```
    metodo_2( )
```

```
    __init__( )
```

Qué es un método constructor?

Es un método particular que se ejecuta automáticamente cuando se instancia un objeto. Los métodos constructores deben llamarse `__init__` obligatoriamente.

Cuál es la utilidad de un constructor?

- **Inicialización de Atributos:** El constructor establece los valores iniciales de los atributos.
- **Configuración Inicial:** El constructor puede llamar a otros métodos de la clase para realizar configuraciones adicionales.
- **Flexibilidad:** Permite ejecutar cualquier lógica necesaria para preparar el objeto para su uso.



ProfesorenVideo.com

Curso de Python

Basta de teoría, mejor programemos ...

... pero después volvemos