

Curso: Programación Orientada a Objetos

Tema:

Definición de Clases, Objetos y Atributos



Logro de aprendizaje

Al finalizar la sesión, el/la estudiante define clases, objetos y atributos de instancia utilizando el paradigma de programación orientada a objetos en el entorno del lenguaje de programación Python.





¿Qué entiende por clase en programación?



Instancia



Definición de clases, objetos y atributos públicos



¿Qué es una clase y cómo se crean?



Las clases son plantillas que permiten empaquetar datos y funcionalidad para crear objetos.

Las clases en Python se crean mediante la palabra reservada “**class**”, seguida del nombre de la clase y tiene la siguiente forma:

```
class nombre_de_la_clase:
```

Ejemplo:

```
class Estudiante:
```



¿Qué es un objeto y cómo se crean?



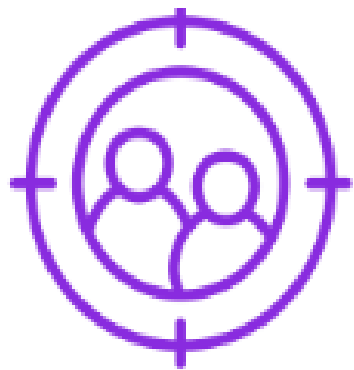
Los objetos son ejemplares particulares de clases que poseen atributos de instancia y métodos.

Para crear objetos en Python se sigue la siguiente forma:

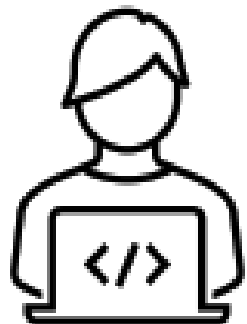
```
Nombre_del_objeto = nombre_de_la_clase()
```

Ejemplo:

```
estudiante_01 = Estudiante()
```



¿Qué son los atributos públicos?



Son aquellos que son accesibles para los usuarios de la clase, sus métodos y otra clase que herede de ella.

Ejemplo: Crear la clase Estudiante, el objeto e1 de tipo estudiante y los atributos públicos código, nombre, nota_01, nota_02 y asignarle valores fijos y luego visualizar los datos fijos en pantalla.



¡ Solución del ejemplo !

Creamos la clase estudiante

```
class Estudiante:  
    pass
```

Creamos el objeto de tipo estudiante

```
e1 = Estudiante()
```

Asignamos datos fijos

```
e1.codigo = 123  
e1.nombre = "Pedro Périco de los Palotes"  
e1.nota_01 = 18  
e1.nota_02 = 20
```

Para Visualizar en pantalla

```
print("Código: ",e1.codigo)  
print("Nombre: ",e1.nombre)  
print("Nota 01: ",e1.nota_01)  
print("Nota 02: ",e1.nota_02)
```

¡ Codificando!



Creación de Objetos Sin Argumentos

```
class Persona:  
    def __init__(self):  
        self.nombre = 'Juan'  
        self.apellido = 'Perez'  
        self.edad = 28
```

```
persona1 = Persona()  
print(persona1.nombre)  
print(persona1.apellido)  
print(persona1.edad)
```

Creación de Objetos con Argumentos

```
class Persona:  
    def __init__(self,nombre,apellido,edad):  
        self.nombre = nombre  
        self.apellido = apellido  
        self.edad = edad
```

```
persona1 = Persona('Juan' , 'Perez' , '28')  
print(persona1.nombre)  
print(persona1.apellido)  
print(persona1.edad)
```

```
persona2 = Persona('Karla' , 'Lopez' , '25')  
print(persona2.nombre)  
print(persona2.apellido)  
print(persona2.edad)
```

```
class Persona:  
    def __init__(self,nombre,apellido,edad):  
        self.nombre = nombre  
        self.apellido = apellido  
        self.edad = edad
```

```
persona1 = Persona('Juan' , 'Perez' , '28')  
print(f'Objeto Persona1: {persona1.nombre}  
{persona1.apellido} {persona1.edad}')
```

```
persona2 = Persona('Karla' , 'Lopez' , '25')  
print(f' Objeto Persona 2: {persona2.nombre}  
{persona2.apellido} {persona2.edad}')
```

Métodos de Instancia

```
class Persona:
```

```
    def __init__(self,nombre,apellido,edad):
```

```
        self.nombre = nombre
```

```
        self.apellido = apellido
```

```
        self.edad = edad
```

```
    def mostrar_detalle(self):
```

```
        print(f'Objeto Persona1: {persona1.nombre} {persona1.apellido}{persona1.edad}')
```

```
persona1 = Persona('Juan' , 'Perez' , '28')
```

```
Persona1.mostrar_detalle()
```

```
persona2 = Persona('Karla' , 'Lopez' , '25')
```

```
Persona2.mostrar_detalle()
```

Ejercicios

1. Implementar una clase llamada Persona que tendrá como atributo (variable) su nombre y dos métodos (funciones), uno de dichos métodos inicializara el atributo nombre y el siguiente método mostrara en la pantalla el contenido del mismo.
Definir dos objetos de la clase Persona.
2. Implementar una clase llamada Alumno que tenga como atributos su nombre y su nota. Definir los métodos para inicializar sus atributos, imprimirlos y mostrar un mensaje si esta aprobado (nota mayor o igual a 11)
Definir dos objetos de la clase alumno
3. Confeccionar una clase que permita cargar el nombre y la edad de una persona. Mostrar los datos cargados. Imprimir un mensaje si es mayor de edad ($\text{edad} \geq 18$)
4. Desarrollar un programa que cargue los datos de un triangulo por teclado e implemente los siguientes métodos: inicializar los atributos, imprimir el valor del lado mayor y otro método que muestre si es equilatero o no. El nombre de la clase llamarla triangulo