



**Curso Introducción a la
programación con Python3**

**Programación
orientada a objetos**


OpenWebinars

Introducción a la Programación orientada a Objetos

```
import math
class punto():
    """ Representación de un punto en el
    plano, los atributos son x e y
    que representan los valores de las
    coordenadas cartesianas."""

    def __init__(self,x=0,y=0):
        self.x=x
        self.y=y

    def mostrar(self):
        return str(self.x)+":"+str(self.y)

    def distancia(self, otro):
        """ Devuelve la distancia entre ambos puntos. """
        dx = self.x - otro.x
        dy = self.y - otro.y
        return math.sqrt((dx*dx + dy*dy))
```

Llamamos **clase** a la representación abstracta de un concepto. Las clases se componen de **atributos** y **métodos**.

Los **atributos** definen las características propias del objeto y modifican su estado.

Los **métodos** son bloques de código (o funciones) de una clase que se utilizan para definir el comportamiento de los objetos.

El **constructor** `__init__`, que nos permite inicializar los atributos de objetos. Este método se llama cada vez que se crea una nueva instancia de la clase (objeto).

Todos los métodos de cualquier clase, recibe como primer parámetro a la instancia sobre la que está trabajando. (**self**).

Introducción a la Programación orientada a Objetos

```
>>> punto1=punto()  
>>> punto2=punto(4,5)  
>>> print(punto1.distancia(punto2))  
6.4031242374328485
```

Para crear un objeto, utilizamos el nombre de la clase enviando como parámetro los valores que va a recibir el constructor.

```
>>> punto2.x  
4  
>>> punto2.x = 7  
>>> punto2.x  
7
```

Podemos acceder y modificar los atributos de objeto.