

# Orientación a objetos

# Orientación a Objetos

Python es orientado a objetos.

Soporta clases, herencia múltiple, e instanciación de clases definidas.

# Orientación a Objetos

```
>>> class Persona(object):
...     "una persona"
...
>>> Persona()
<__main__.Persona instance at 0x170eb00>
>>> Persona().__class__
<class __main__.Persona at 0x247fc18>
>>> Persona().__doc__
'una persona'
>>> class Persona(object):
...     pass
...
```

# Orientación a Objetos

```
>>> class Persona(object):  
...     """Una Persona"""  
...  
...     def __init__(self, nombre, edad):  
...         self.nombre = nombre  
...         self.edad = edad  
...  
>>> Persona("Diego Armando", 50)  
<__main__.Persona instance at 0x170ed88>
```

# Orientación a Objetos

```
>>> p = Persona("Diego Armando", 50)
<__main__.Persona instance at 0x170ed88>
>>>
>>> p = Persona("Diego Armando", 50)
>>> p.nombre
'Diego Armando'
>>> p.edad
50
>>> dir(p)
['__doc__', '__init__', '__module__', 'edad', 'nombre']
```

# Orientación a Objetos

```
>>> class Persona(object):
...     """ """
...     numero = 0
...
...     def __init__(self, nombre,edad):
...         self.__class__.numero += 1
...         self.nombre = nombre
...         self.edad = edad
...
>>> Persona('a',2 )
<__main__.Persona instance at 0x1711098>
>>> Persona('b',3 )
<__main__.Persona instance at 0x1711050>
>>> Persona.numero
2
```

# Orientación a Objetos

```
class Persona(object):  
    """ """  
  
    numero = 0  
  
    def __init__(self, nombre, edad):  
        Persona.numero += 1  
        self.nombre = nombre  
        self.edad = edad  
        self._privado = "esto es privado"  
        Self.__muy_privado = "esto es muy privado"  
  
    def _privado(self):  
        return self._privado  
  
    def muy_privado(self):  
        return self.__muy_privado
```

# Orientación a Objetos

```
>>> from persona import *
>>> Persona('a',1)
<persona.Persona instance at 0x238d758>
>>> p = Persona('a',1)
>>> p.privado
<bound method Persona.privado of <persona.Persona instance at 0x238d7a0>>
>>> p.privado()
'esto es privado'
>>> p._privado
'esto es privado'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: Persona instance has no attribute '__algo'
```



# Orientación a Objetos

```
>>> p.muy_privado
<bound method Persona.muy_privado of <persona.Persona instance at 0x238d7a0>>
>>> p.muy_privado()
'esto es muy privado'
>>> p.__muy_privado
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: Persona instance has no attribute '__muy_privado'
>>> dir(Persona)
```

# Orientación a Objetos

```
class Jugador(Persona):  
    """Un Jugador"""  
    def __init__(self, nombre, edad, titular):  
        super(Persona, self).__init__(nombre, edad)  
        self.titular = False  
  
    def juega(self):  
        self.titular = True  
        return self.titular
```

# Orientación a Objetos - Properties

```
class Persona(object):
    """ """

    def __init__(self, nombre, edad):
        self._nombre = nombre
        self._edad = edad

    @property
    def nombre(self):
        return self._nombre

    @property
    def edad(self):
        return self._edad

    @edad.setter
    def edad(self, v):
        assert isinstance(v, int)
        self._edad = v
```

# Orientación a Objetos - Properties

```
>>> p = Persona("tito", 88)
>>> p.nombre
'tito'
>>> p.nombre = "oscar"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>> p.edad
88
>>> p.edad = 100
>>> p.edad
100
>>> p.edad = "hola"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "p.py", line 20, in edad
    assert isinstance(v, int)
AssertionError
```