

Programación Orientada a

Objetos



python

02

Métodos especiales



Contenido

- Constructor `__init__`
 - Validación de datos
- Destructor `__del__`
- `__str__`
- Sobrecarga de operadores
 - Suma: `__add__`
 - Resta: `__sub__`
 - Multiplicación: `__mul__`
 - División: `__div__`
 - Igual a: `__eq__`
 - Menor que: `__lt__`
 - Mayor que: `__gt__`

Las clases en Python cuentan con múltiples métodos especiales, los cuales se encuentran entre dobles guiones bajos `__<metodo>__()`.



Constructores

`__init__()`

- El método `__init__()` es un método especial, el cual se ejecuta al momento de instanciar un objeto.
- El comportamiento de `__init__()` es muy similar a los “constructores” en otros lenguajes.
- Los argumentos que se utilizan en la definición de `__init__()` corresponden a los parámetros que se deben ingresar al instanciar un objeto.



Agregando validaciones al constructor

- El constructor inicializa al objeto, se asegura que tengan valores iniciales los atributos del objeto
- Que los valores sean valores válidos
- Parámetros por default???

```
def __init__(self, num=0, den=1):  
    if isinstance(num, int):  
        self.num = num  
    else:  
        self.num=0  
    if isinstance(den, int) and den!=0:  
        self.den = den  
    else:  
        self.den=1
```

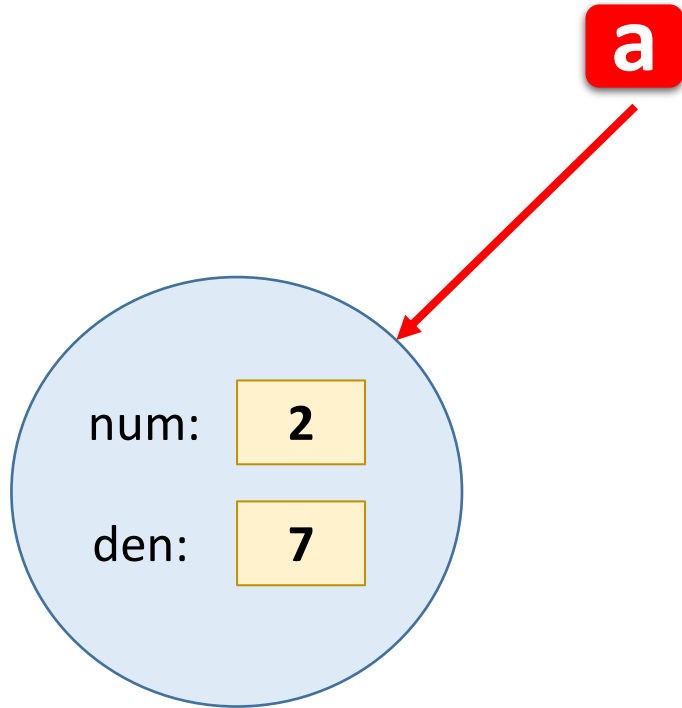


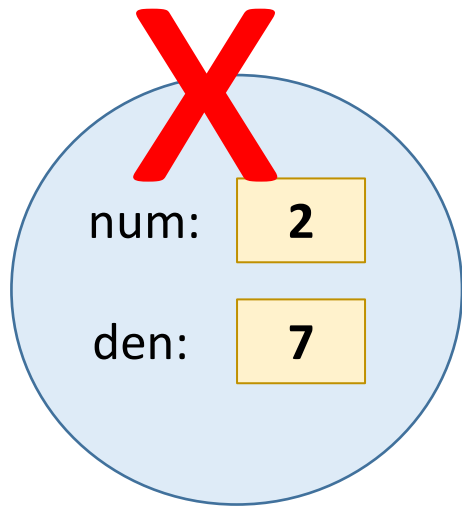
Destruyores __del__

- Es un método especial, el cual **se ejecuta al momento en el cual un objeto es descartado** por el intérprete.
- El comportamiento de __del__() es muy similar a los “destruyores” en otros lenguajes.
- Se ejecuta cuando se destruye o muere el objeto



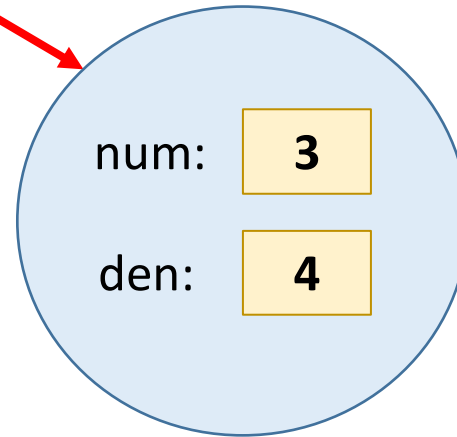
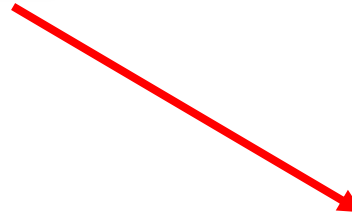
`a = Fraccion(2,7)`





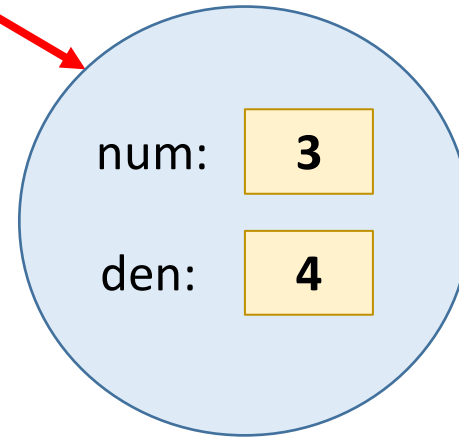
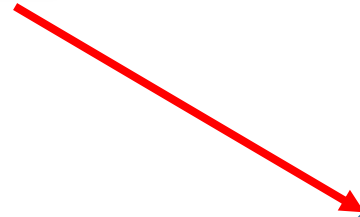
Objeto des-referenciado
Se destruye
Se ejecuta `__del__()`

a



```
a = Fraccion(2,7)  
...  
a = Fraccion(3,4)
```

a



__str__()

- El método __str__() es un método especial, el cual se ejecuta al momento en el cual un objeto se manda a mostrar, es decir es una cadena representativa de la clase, la cual puede incluir formatos personalizados de presentación del mismo.
- Regresa un string



Sobrecarga de operadores

- Suma + : `__add__`
- Resta - : `__sub__`
- Multiplicación * : `__mul__`
- División / : `__div__`
- Igual a == : `__eq__`
- Menor que < : `__lt__`
- Mayor que > : `__gt__`

```
def __mul__(self, b):  
    n = self.num * b.num  
    d = self.den * b.den  
    r = Fraccion(n,d)  
    return r
```

```
def main():  
    a = Fraccion(3,2)  
    b = Fraccion(2,5)  
    r = a*b
```

```
class Fraccion:  
    def __init__(self, num,den):  
        self.num = num  
        self.den = den  
  
    def imprime(self):  
        print("{", self.num, "/", self.den, "}")  
  
    def multiplica(self, b):  
        n = self.num * b.num  
        d = self.den * b.den  
        r = Fraccion(n,d)  
        return r
```

```
def main():  
    a = Fraccion(3,2)  
    b = Fraccion(2,5)  
    r = a.multiplica(b)  
    a.imprime()
```



Programación Orientada a

Objetos



python

02

Métodos especiales

