

Para saber mais: herança menos verbosa

Temos o seguinte código:

```
class Conta(object):
    def __init__(self, titular, saldo):
        self.titular = titular
        self.saldo = saldo

    def calcular_imposto(self):
        self.saldo = self.saldo * 0.10
        return self.saldo

class ContaCorrente(Conta):
    def __init__(self, titular, saldo):
        super(ContaCorrente, self).__init__(titular, saldo)

    def calcular_imposto(self):
        return super(ContaCorrente, self).calcular_imposto() + 20
```

Veja que a classe filha possui um construtor (**init**) idêntico ao do pai. Nesse caso, quando a classe filha receber os mesmos parâmetros do construtor pai, não precisamos declarar a a função **init** na classe filha. Vejamos como fica:

```
class Conta(object):
    def __init__(self, titular, saldo):
        self.titular = titular
        self.saldo = saldo

    def calcular_imposto(self):
        self.saldo = self.saldo * 0.10
        return self.saldo

class ContaCorrente(Conta):
    def calcular_imposto(self):
        return super(ContaCorrente, self).calcular_imposto() + 20
```

No entanto, se a classe filha recebe um parâmetro extra em seu construtor, precisamos invocar o construtor do pai como no exemplo abaixo:

```
class Conta(object):
    def __init__(self, titular, saldo):
        self.titular = titular
```

```
self.saldo = saldo
```

```
def calcular_imposto(self):  
    self.saldo = self.saldo * 0.10  
    return self.saldo
```

```
class ContaCorrente(Conta):
```

```
    def __init__(self, titular, saldo, bonus):  
        super(ContaCorrente, self).__init__(titular, saldo)  
        self.bonus = bonus
```

```
    def calcular_imposto(self):  
        return super(ContaCorrente, self).calcular_imposto() + self.bonus
```

```
cc = ContaCorrente('xxx', 2000, 50);  
print cc.calcular_imposto()
```