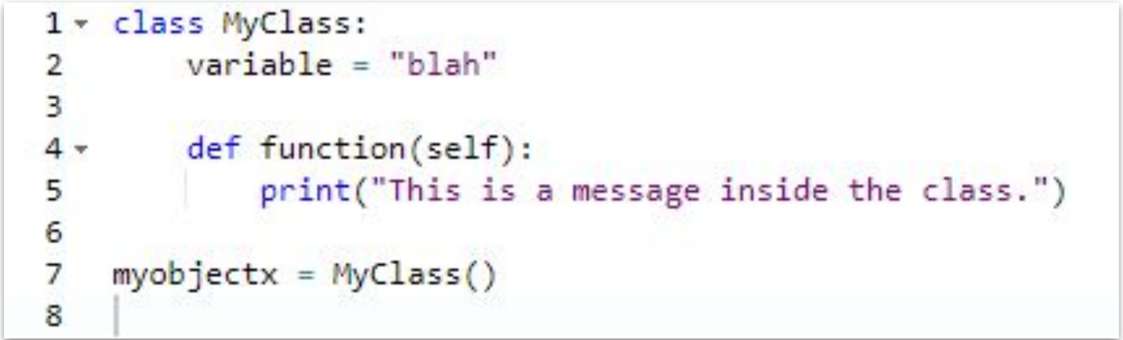
trobz

Python - Classes:
Object, Attributes and Methods

- Classes and Objects
- The self
- The __init__()
- Inheritance
- Encapsulation
- Polymorphism

Classes and Objects

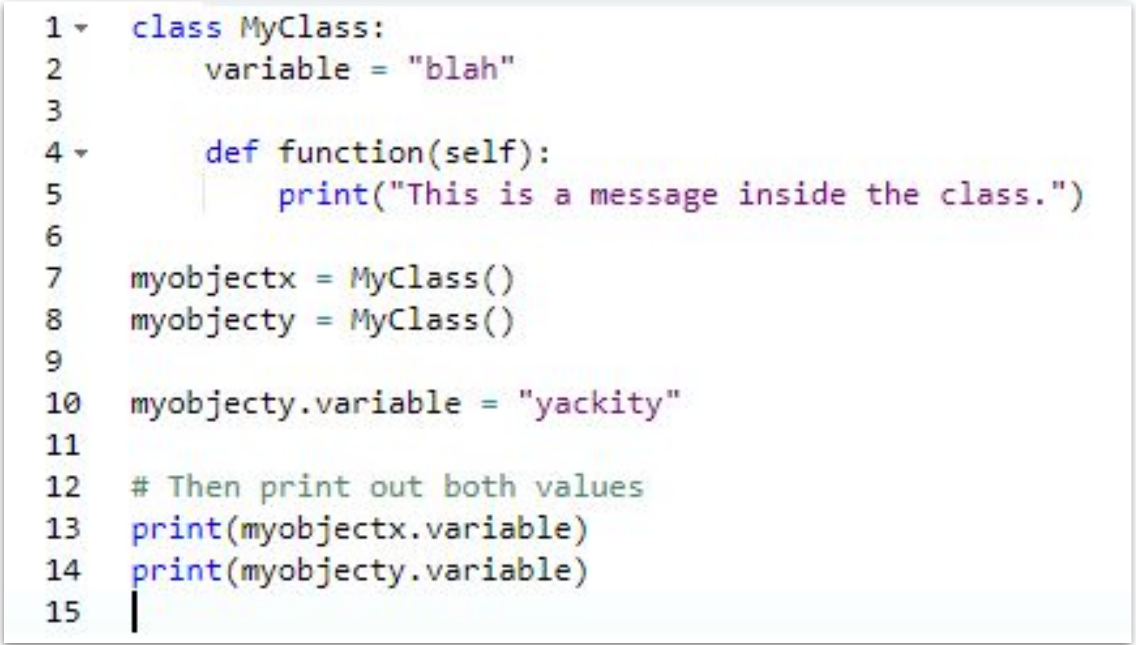
- Object:
  - Is an encapsulation of **variables** and **functions** into **a single entity**
    - Objects get their **variables** and **functions** <u>**from classes**</u>
  - You can create **multiple different objects** that are of **the same class**
    - Each object contains **independent copies** of the **variables** defined in the class
- Class:
  - Is essentially a **template** to create your objects



```
class MyClass:
    variable = "blah"

    def function(self):
        print("This is a message inside the class.")

myobjectx = MyClass()

```

Classes and Objects (2)

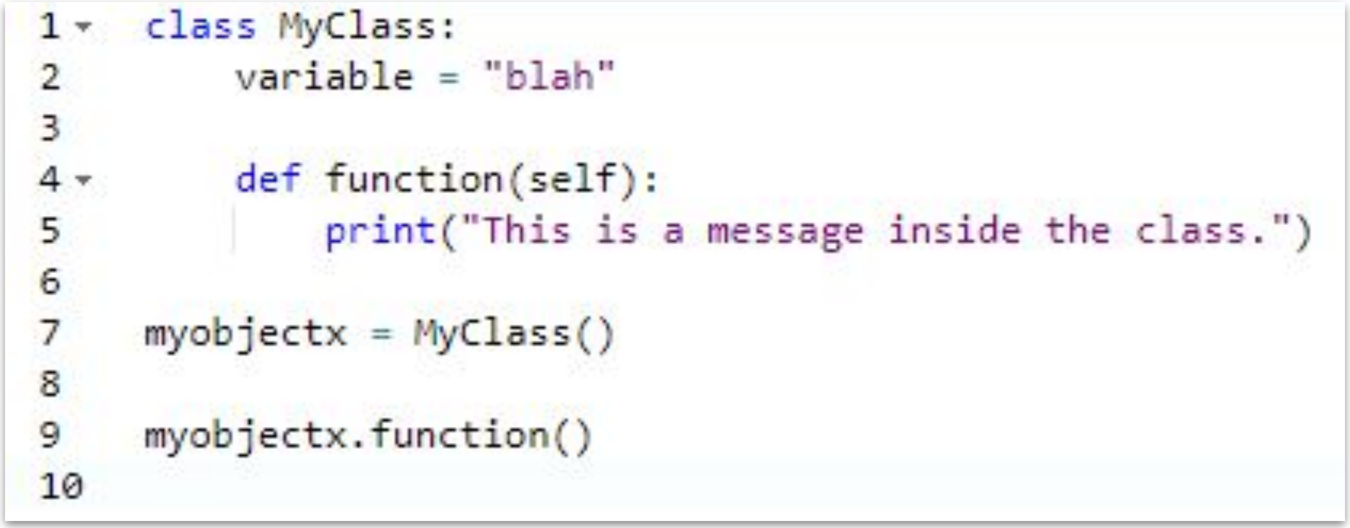
- Accessing Object Variables:



```python
class MyClass:
    variable = "blah"

    def function(self):
        print("This is a message inside the class.")

myobjectx = MyClass()
myobjecty = MyClass()

myobjecty.variable = "yackity"

# Then print out both values
print(myobjectx.variable)
print(myobjecty.variable)

```

Classes and Objects (3)

- Accessing Object Functions:



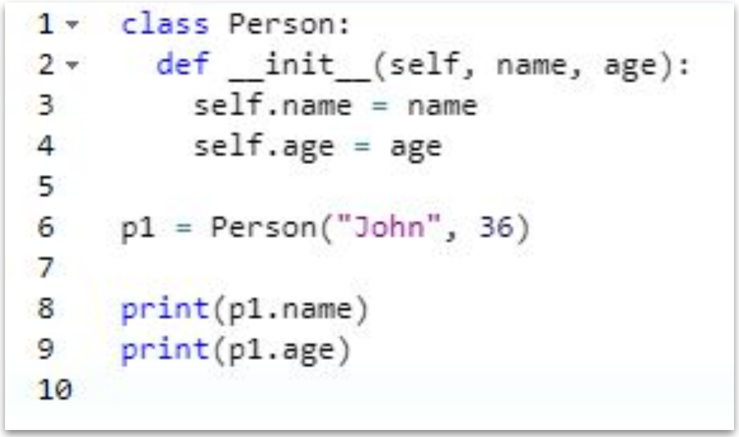
```python
class MyClass:
    variable = "blah"

    def function(self):
        print("This is a message inside the class.")

myobjectx = MyClass()

myobjectx.function()

```

The self

- **self**
  - Is an **extra first parameter** in method definition.
    - We **do not give a value for this parameter** when we call the method, Python provides it.
    - If we have a method which takes no arguments, then we still have to have one argument.
  - Is a reference to the **current instance** of the class
  - Is used to access variables that belongs to the class.
- **It does not have to be named self**
  - You can call it whatever you like, but it has to be the first parameter of any function in the class

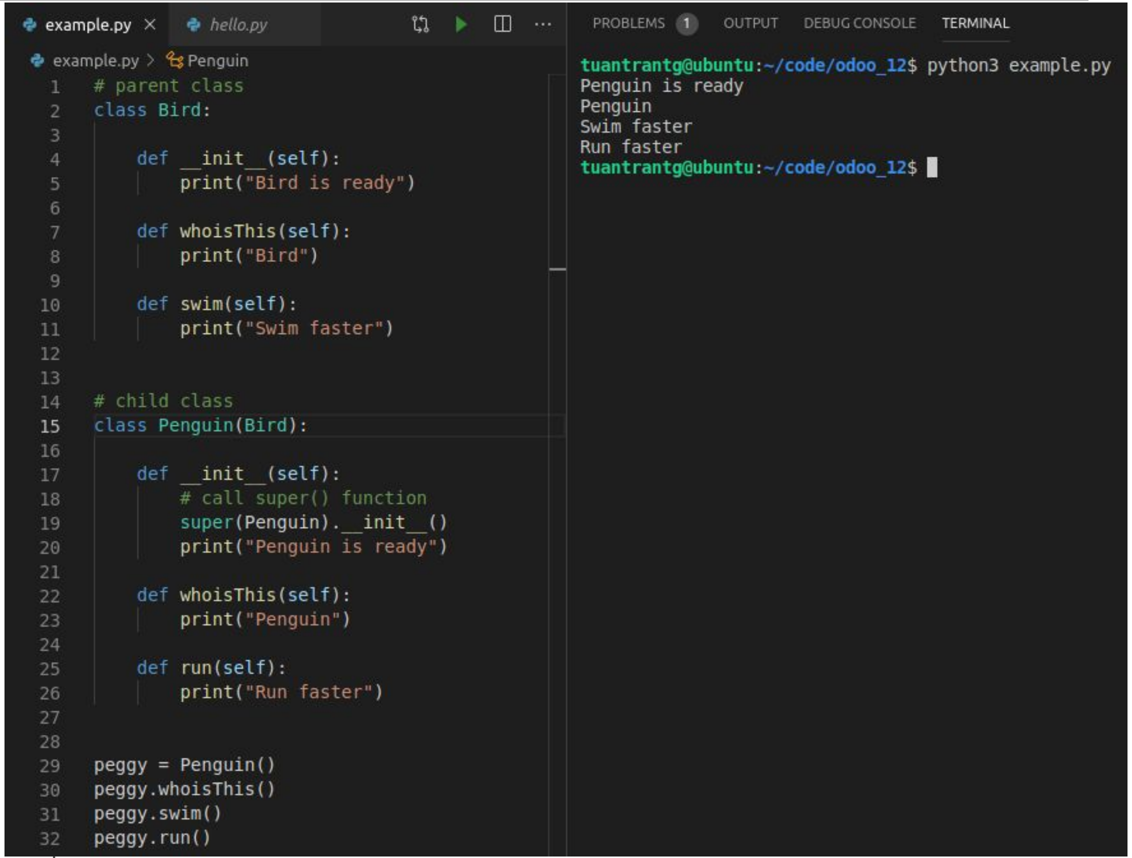
The __init__()

- **__init__()**
    - Is always executed **when the class is being initiated**
    - Is used to assign **values to object properties**, or **other operations** that are necessary to do **when the object is being created**
    - All classes have a function called **__init__()**



```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

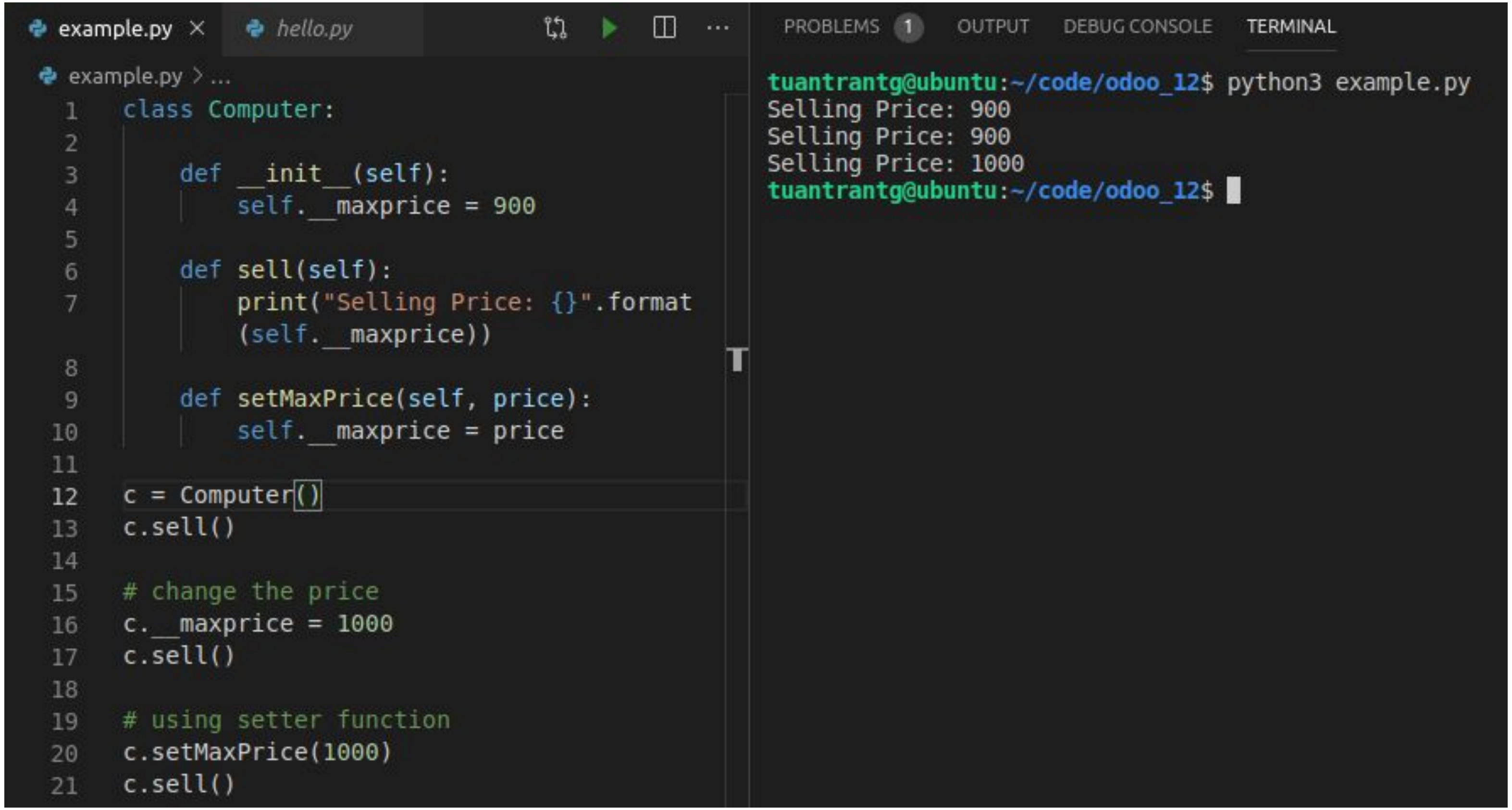
Inheritance



```python
# parent class
class Bird:

    def __init__(self):
        print("Bird is ready")

    def whoisThis(self):
        print("Bird")

    def swim(self):
        print("Swim faster")


# child class
class Penguin(Bird):

    def __init__(self):
        # call super() function
        super(Penguin).__init__()
        print("Penguin is ready")

    def whoisThis(self):
        print("Penguin")

    def run(self):
        print("Run faster")


peggy = Penguin()
peggy.whoisThis()
peggy.swim()
peggy.run()
```

```
tuantrantg@ubuntu:~/code/odoo_12$ python3 example.py
Penguin is ready
Penguin
Swim faster
Run faster
tuantrantg@ubuntu:~/code/odoo_12$
```

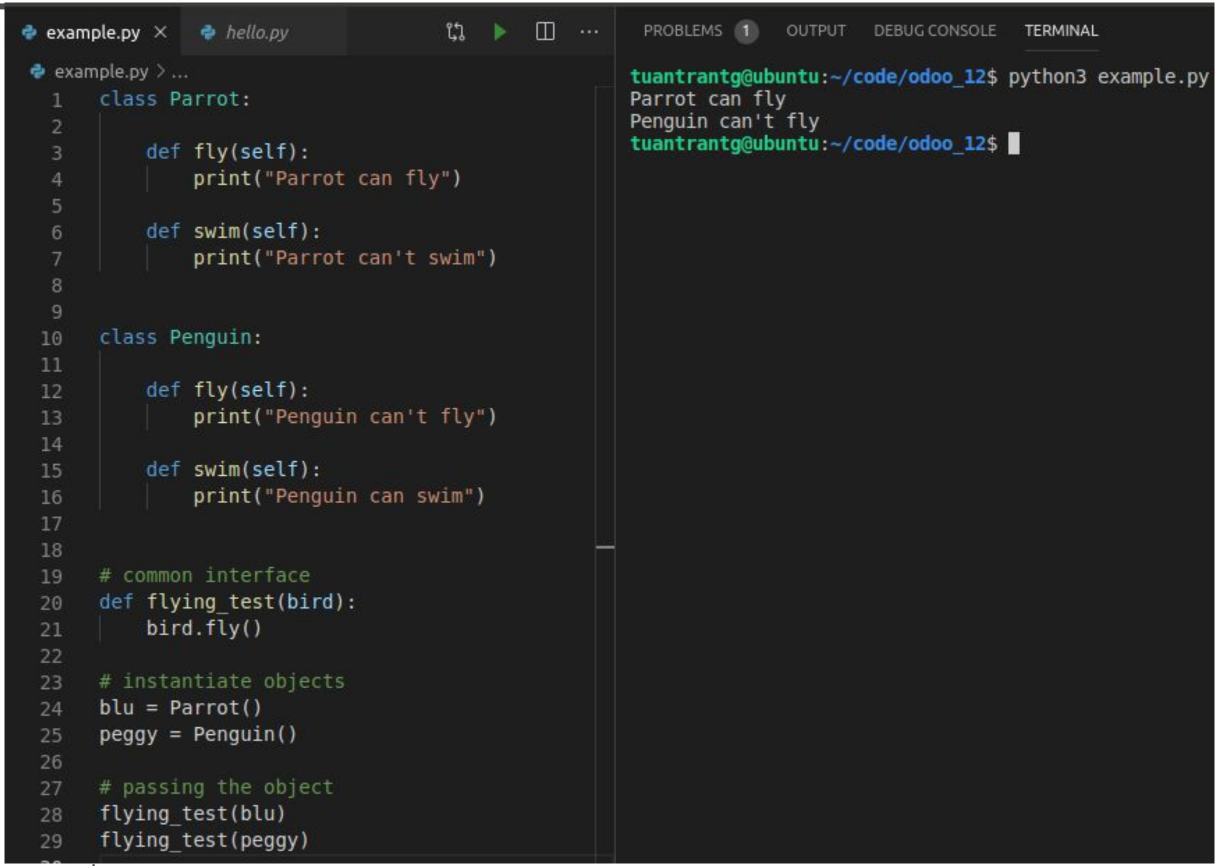
Encapsulation



```python
class Computer:

    def __init__(self):
        self.__maxprice = 900

    def sell(self):
        print("Selling Price: {}".format
        (self.__maxprice))

    def setMaxPrice(self, price):
        self.__maxprice = price

c = Computer()
c.sell()

# change the price
c.__maxprice = 1000
c.sell()

# using setter function
c.setMaxPrice(1000)
c.sell()
```

```
tuantrantg@ubuntu:~/code/odoo_12$ python3 example.py
Selling Price: 900
Selling Price: 900
Selling Price: 1000
tuantrantg@ubuntu:~/code/odoo_12$
```

Polymorphism



```python
class Parrot:

    def fly(self):
        print("Parrot can fly")

    def swim(self):
        print("Parrot can't swim")


class Penguin:

    def fly(self):
        print("Penguin can't fly")

    def swim(self):
        print("Penguin can swim")


# common interface
def flying_test(bird):
    bird.fly()

# instantiate objects
blu = Parrot()
peggy = Penguin()

# passing the object
flying_test(blu)
flying_test(peggy)
```

```
tuantrantg@ubuntu:~/code/odoo_12$ python3 example.py
Parrot can fly
Penguin can't fly
tuantrantg@ubuntu:~/code/odoo_12$
```

Q & A

Thank You!