

# Basic object-oriented programming concepts in Python

01219116/01219117  
Programming 2

# Objects and classes

```
class Dog:

    def __init__(self, name):
        self.name = name
        self.tricks = []

    def add_trick(self, trick):
        self.tricks.append(trick)
```

```
>>> d = Dog('Fido')
>>> e = Dog('Buddy')
>>> d.add_trick('roll over')
>>> e.add_trick('play dead')
>>> d.tricks
['roll over']
>>> e.tricks
['play dead']
```

# Everything is an object in Python

- A class is also an object.
- A method is an object.

# Class and instance variables

```
class Dog:
    kind = 'canine'

    def __init__(self, name):
        self.name = name
```

```
>>> d = Dog('Fido')
>>> e = Dog('Buddy')
>>> d.kind
'canine'
>>> e.kind
'canine'
>>> d.name
'Fido'
>>> e.name
'Buddy'
```

# Private variables

- There is no way to make variable private in Python.
- Use conventions.
  - Prefix with `_` to indicate private use
  - Prefix with `__` to let python automatically add classname prefix to avoid accident name clashes.

# Static methods

# Inheritance

# Calling parent class's methods

- Use `super()`



# Mixin

# Practice: Sokoban classes

- Player
- Box
- Exit
- Wall (maybe)
- Board

# CRC card

Class name

Responsibility (as method names)

e.g.

- move\_left, move\_right
- r, c (locations)

Collaborators