Python Dictionaries and Objects

Dictionaries

- Hash maps
- Map keys -> values
- Like a real dictionary… word -> definition

Mapping name to age

- Answer the question: What is Leah's age?
- Dictionary where keys are names (Strings) and values are ages (integers)

Mapping name to age

```
ages = {}
ages[“Leah”] = 22
ages[“Taibo”] = 21
print ages
>> {'Leah': 22, 'Taibo': 21}

print ages[“Leah”]
22

ages[“Leah”] = 23
print ages[“Leah”] + ages[“Taibo”]
43
```

Dictionaries

- Keys and values can be other data types

Objects

- A standard way to organize information
- Holds similar information about a single “thing” in one place
- Variables and procedures
- In fact, all the data structures you've learned are also objects (lists, strings, dictionaries)
- Object examples:
  - Football tournament
  - Race car

Defining a Class

```
class Car():
    def __init__(self):
        self.wheels = 4

myCar = Car()
print myCar.wheels ?
```

Defining a Class

```
class Car():
    def __init__(self):
        self.wheels = 4

myCar = Car()
yourCar = Car()
print myCar.wheels ?
print yourCar.wheels ?
myCar.wheels = 3
print myCar.wheels ?
print yourCar.wheels ?
```

Constructor with parameters

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color
```

Constructor with parameters

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color

myCar = Car(“red”)
print myCar.color ?
```

Adding Procedures

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color

    def lose_a_wheel(self):
        self.wheels = self.wheels – 1

myCar = Car(“red”)
print myCar.wheels ?
myCar.lose_a_wheel()
print myCar.wheels ?
myCar.lose_a_wheel()
print myCar.wheels ?
```

Procedure with parameters

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color

    def lose_a_wheel(self):
        self.wheels = self.wheels – 1

    def lose_wheels(self, num_wheels_lost):
        self.wheels = self.wheels – num_wheels_lost

myCar = Car(“green”)
myCar.lose_wheels(3)
print myCar.wheels ?
```

Procedure to print output

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color

    def printColor(self):
        print “My color is”, self.color

myCar = Car(“green”)
myCar.printColor()
```

Inheritance

```
class Car():
    def __init__(self, color):
        self.wheels = 4
        self.color = color

class Taxi(Car):
    def price_per_ride(self):
        return 10

myTaxi = Taxi("yellow")
print myTaxi.color
print myTaxi.price_per_ride()
```

Object examples in your projects

Restaurant

- Name
- List of menu items
- add_menu_item()

Painting

- Artist
- Sold/unsold
- price
- set_price()

Lab 4: Objects and dictionaries

- Complete all exercises
- If you're stuck, don't leave! Ask me for help.
- Show us once you're done with problem 2.
- Sample code from lecture on page 2

Lab notes

init has 2 underscores on each side…
def _ _init_ _(self):

Defining a function:
def function_name(parameters):
    statements

For example:
def add(a,b):
    return a+b