[320] Object Oriented Programming

Yiyin Shen

Creating New Types

CLASSES AND OTHER TYPES

list

dict

movie

str

player

person
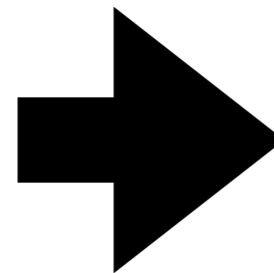
OBJECTS

Classes

dicts can represent many kinds of things

https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2021-09-21&endtime=2021-09-22

classes (today) are often a better option when all your keys are the same

```
m1 = {...}
m2 = {...}
```

create some objects
of type **dict** for **movies**

```
p1 = {}
p2 = {}
p3 = dict()
```

create some objects
of type **dict** for **people**

set some keys/values

```
p1["Fname"] = "Joseph"
p2["fname"] = "Peyman"
p3["fname"] = "Shri Shruthi"

print(type(m1))
print(type(p1))
```

```python
class Person:
    pass
```
create a Person type/class

```python
p1 = Person()
p2 = Person()
p3 = Person()
```
create some objects of type Person

set some attributes

```python
p1.Fname = "Joseph"
p2.fname = "Peyman"
p3.fname = "Shri Shruthi"

print(type(p3))
```

Objects created from classes are mutable.
Attribute names are not fixed at creation.

PythonTutor: Compare dicts to class types

Python 3.6
known limitations

```
1  p1 = {"x": 4, "y": 5}
2
3  class Coord:
4      pass
5
6  p2 = Coord()
7  p2.x = 4
8  p2.y = 5
```

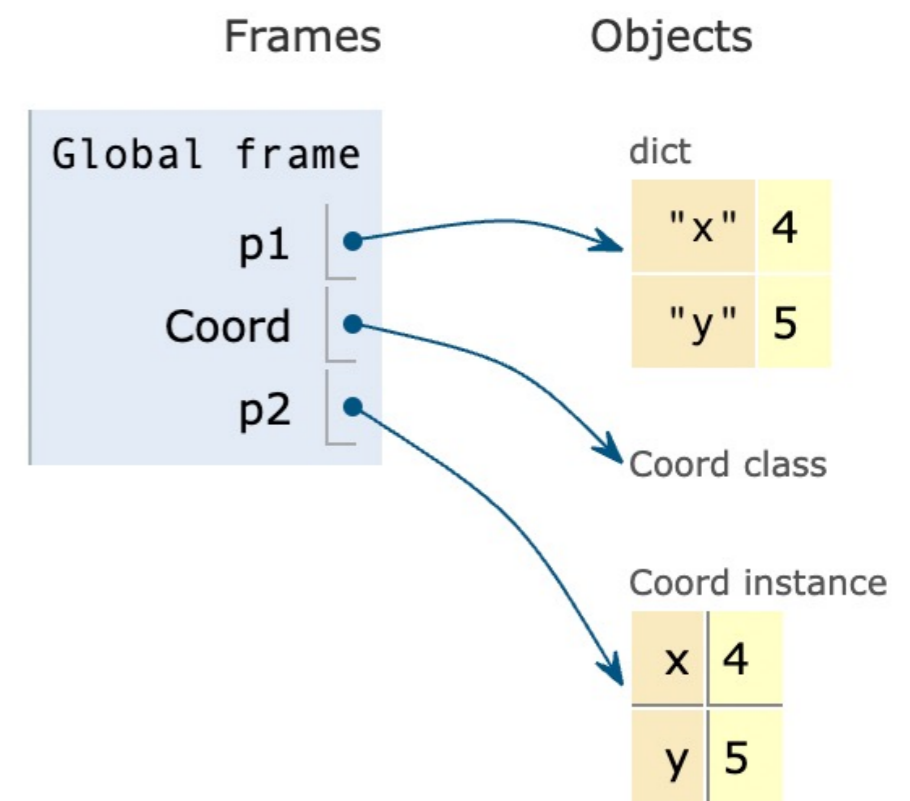Edit this code

➡ line that just executed
➡ next line to execute

<< First   < Prev   Next >   Last >>

Done running (5 steps)

Visualized with pythontutor.com

NEW: subscribe to our YouTube

Move and hide objects

Frames

Global frame
p1
Coord
p2

Objects

dict
"x" 4
"y" 5

Coord class

Coord instance
x 4
y 5

Coding Examples: Animal Classes

Principals

- objects and functions

- methods

- checking object type

- type-based dispatch

- receiver (self parameter)

- constructors