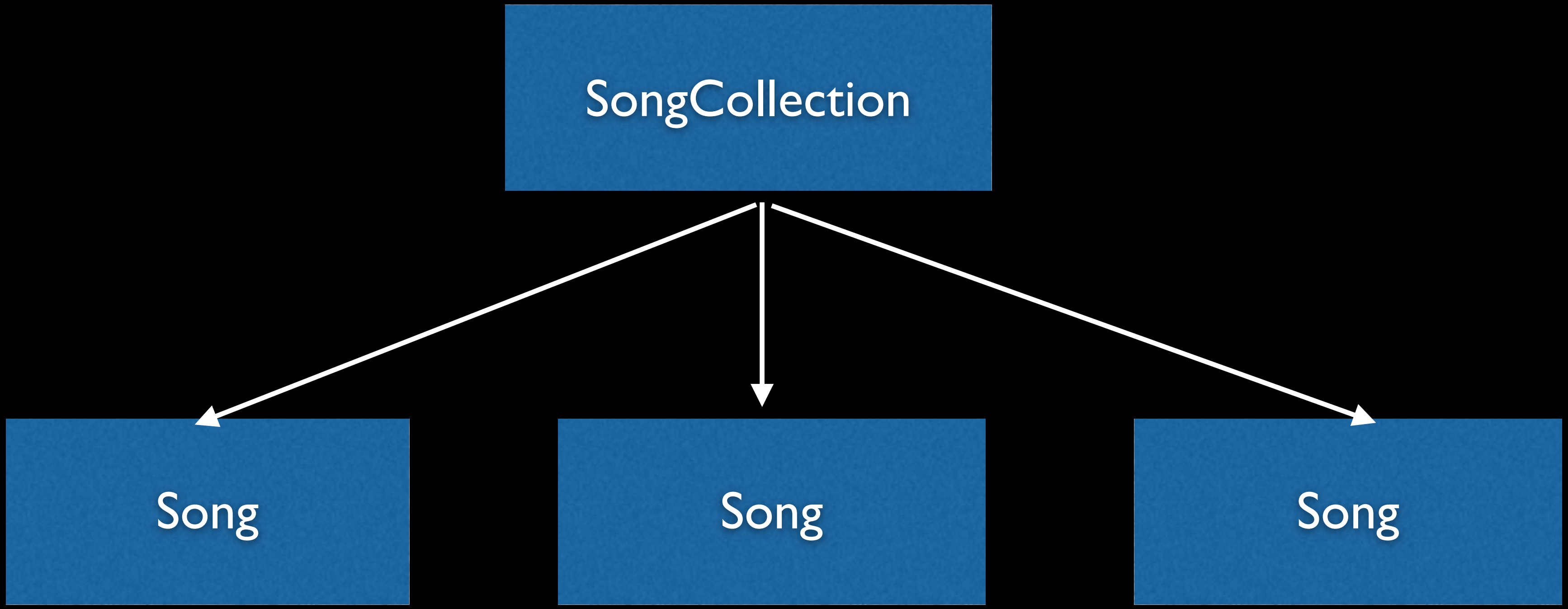
π

Python course - Day 8

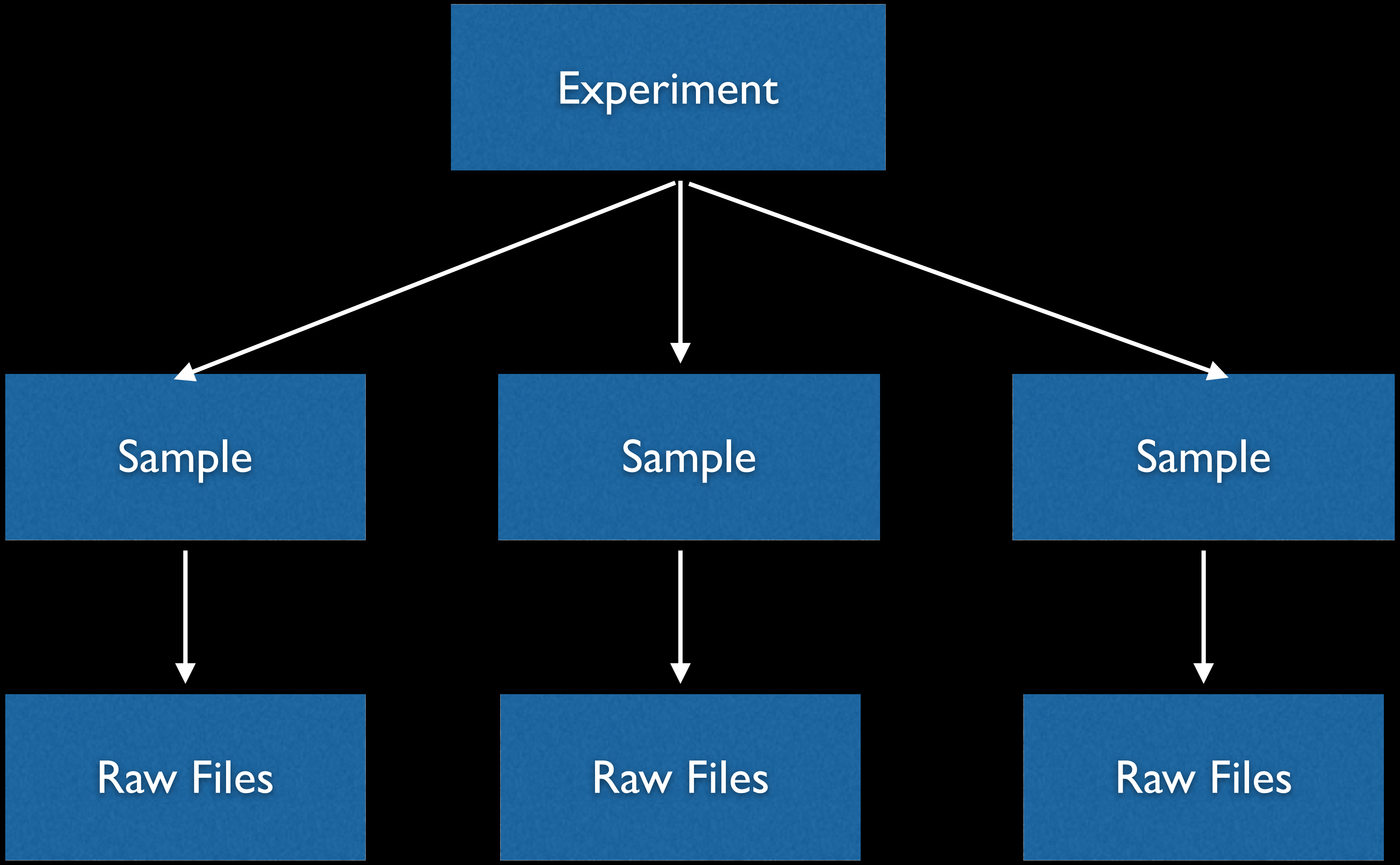
Composition

1. You have a custom object of which, one of the attributes, is another custom object
2. Contrasts with inheritance ("is a" or "has a")

Composition



Composition



Inheritance

```python
class Door(object):
   def __init__(self):
       self.status = "closed"
   def open(self):
       self.status = "open"

class SecurityDoor(Door):
    color = 'gray'
    locked = True

   def open(self):
       if self.locked: return
       return Door.open(self)

   def unlock(self):
      self.locked = False

```

Composition

```python
class Door(object):
    def __init__(self):
        self.status = "closed"
    def open(self):
        self.status = "open"

class SecurityDoor(object):
    color = 'gray'
    locked = True
    def __init__(self):
        self.door = Door()

    def open(self):
        if self.locked: return
        return self.door.open(self)

    def unlock(self):
       self.locked = False
```

Composition

Live demo

Argparse

1. Built-in module (import argparse)
2. Easier than dealing with sys.argv yourself
3. Will give you some functionality "for free"

With sys.argv

```
import sys
assert len(sys.argv == 4)

if sys.argv[1] == '-h' or sys.argv[1] == '--help':
    print "The help of this tool is the following:"

if not sys.argv[1].startswith("-"):
    print "You forgot to place a required option."

....
```

Command line tool

```
import argparse
```

Command line tool

```python
import argparse
parser = argparse.ArgumentParser()
```

Command line tool

```python
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file")
parser.add_argument("output_file")
```

Command line tool

```python
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file")
parser.add_argument("output_file")

args = parser.parse_args()
```

Command line tool

```python
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file")
parser.add_argument("output_file")

args = parser.parse_args()

print args.input_file
print args.output_file
```

Argparse

Live demo

Command line tool

```
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file", help="This is
the path of the input FASTA file.", type=str)
parser.add_argument("output_file", help="The
result of the analysis will be placed here.",
type=str)


args = parser.parse_args()

print args.input_file
print args.output_file
```

Command line tool

```
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file", help="This is the path
of the input FASTA file.", type=str)
parser.add_argument("output_file", help="The result of the
analysis will be placed here.", type=str)

parser.add_argument("-t", "--threshold", help="The scoring
threshold", type=int, default=10)

args = parser.parse_args()

print args.input_file
print args.output_file
print args.threshold
```

Command line tool

```python
import argparse
parser = argparse.ArgumentParser()

parser.add_argument("input_file", help="This is the path of the input FASTA file.", type=str)
parser.add_argument("output_file", help="The result of the analysis will be placed here.", type=str)

parser.add_argument("-t", "--threshold", help="The scoring threshold", type=int, default=10)
parser.add_argument("-f", "--fast", help="increase the speed of the program", action="store_true")

args = parser.parse_args()

print args.input_file
print args.output_file
print args.threshold
print args.fast
```

Argparse

Example of a big tool

Argparse

1. Adding mutually exclusive options
2. Easier than dealing with sys.argv yourself
3. Will give you some functionality "for free"

10 minutes break