

# ②

## Pythonのクラスと継承

---

DjangoでModelを作成するにあたって  
必要なクラスの概念についておさらいします

# Pythonのクラス

- コンピュータ・サイエンス概論Iで学習した「クラス」を覚えていますか？
  - 覚えていなければ、第12-1回を復習しよう！
- Djangoでも、Pythonのクラスを使用します
  - 特に、モデルを作成する際に、クラスを作成する必要があります
  - さらに、今まで学習していない「継承」、「クラス変数」という仕組みも使用します

# Pythonのクラスとインスタンスのおさらい

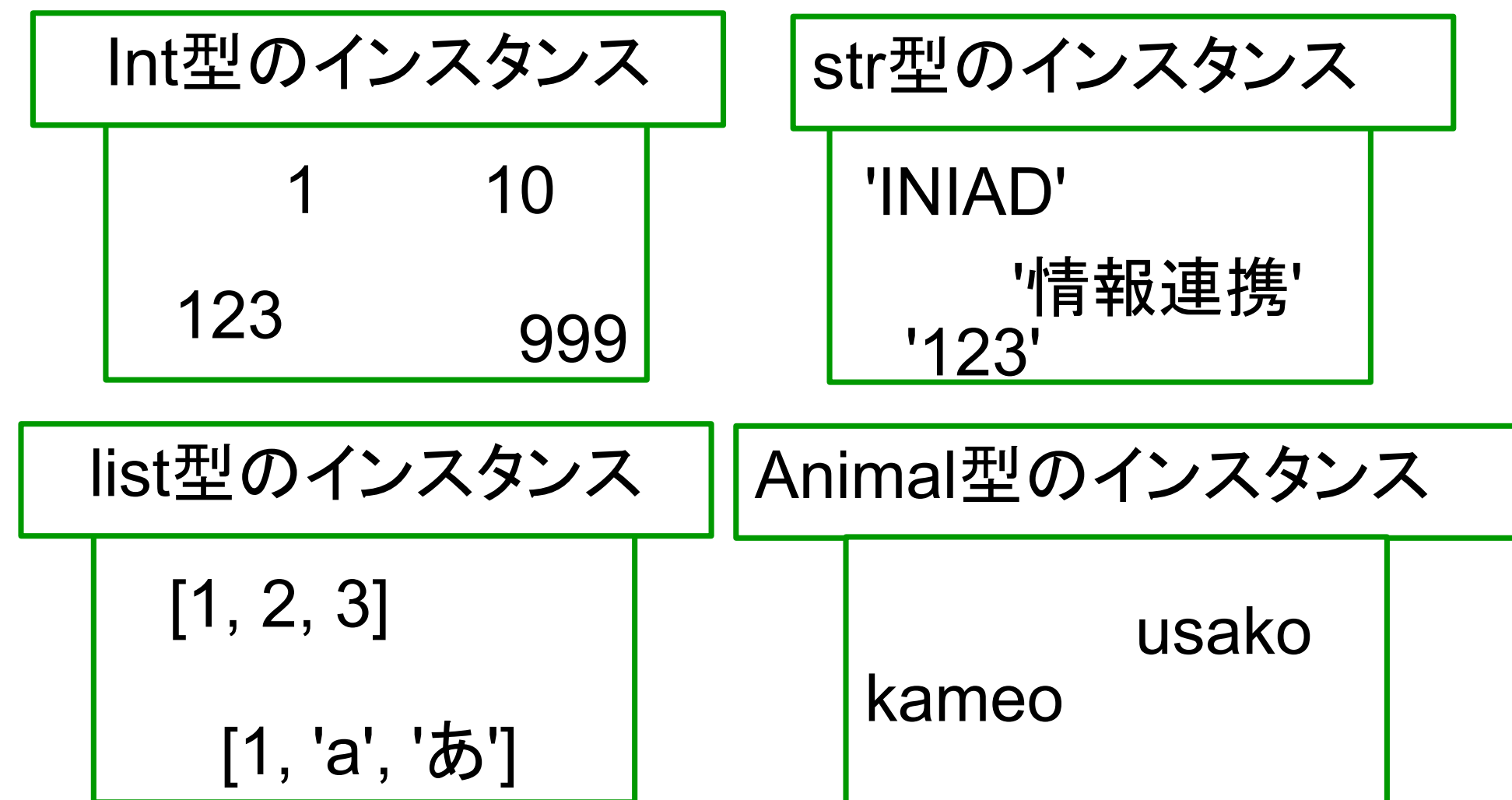
## ● クラスとは

- データを抽象化したもの
- インスタンスの設計図
- データ型と同じ意味
- 例
  - int (整数)
  - str (文字列)
  - list (リスト)
  - Animal (動物)



## ● インスタンスとは

- データの具体例
- あるクラスの具体的な個体
- データの値と同じ意味
- 例



# クラスの定義と、メソッド、属性、クラス変数

- クラスは「class クラス名」と書いて定義します

- 右の例は動物を表すクラス
- 関数と区別するため、クラス名の1文字目は大文字

Animalの日本語表記は「動物」

- クラス変数

- クラスに共通する性質、クラス自体の性質
- 例: クラスAnimal の日本語の表記「動物」

クラスAnimalの定義

```
class Animal:  
    japanese_class_name = '動物'
```

クラス変数

- 属性

- あるクラスの各インスタンスに固有の性質
- 例: タロー、ハナコ、等動物の各個体の名前

属性

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

- メソッド

- あるクラスの各インスタンスに共通する操作
- 例: 動物の「食べる」

メソッド

```
def eat(self):  
    print('ムシヤムシヤ')
```

# クラスの継承

- Pythonのクラスは、別のクラスを、「継承」することができます
  - 継承元のクラスを親クラス（スーパークラス）、  
継承先のクラスを子クラス（サブクラス）と呼びます
- 子クラスのインスタンスは、親クラスの属性やメソッドを利用できる
  - 親から属性・メソッドを引き継ぐ！
- 継承するには、クラスの定義でクラス名の後ろに(親クラス)とかく

親クラス

継承の例 animal.py

```
class Animal:
    japanese_class_name = '動物'

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

    def eat(self):
        print('ムシャムシャ')

class Dog(Animal):
    japanese_class_name = '犬'

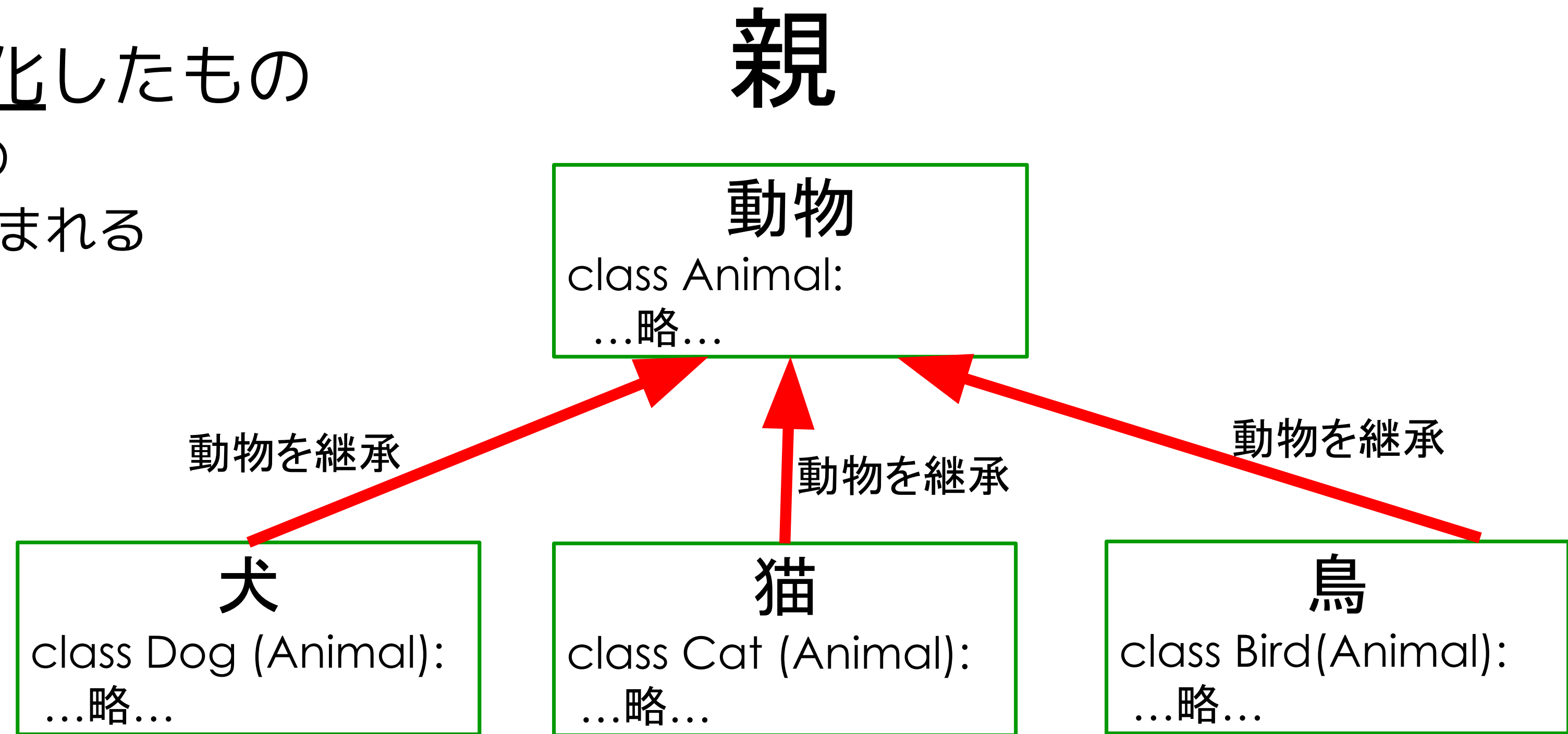
    def bark(self):
        print('ワンワン')
```

子クラス

# 親クラスと子クラスの関係

- 親クラスは、子クラスを汎化したもの

- より抽象的・一般的にしたもの
- 例: 動物には、犬、猫、鳥が含まれる



- 子クラスは、親クラスを特化したもの

- より具体的にしたもの
- 例: 犬、猫、鳥の方が、生き物の種別について「動物」より詳しい言い方



# 継承の機能

- 子クラスのインスタンスは、親クラスのメソッドも利用できる

```
from animal import Dog, Bird

my_dog = Dog('タロウ')
my_dog.eat() # ムシャムシャ と表示
my_dog.bark() # ワンワン と表示

my_bird = Bird('チャッピー')
my_bird.eat() # ムシャムシャ と表示
```

ムシャムシャ  
ワンワン  
ムシャムシャ

親クラスのメソッドeatを呼び出せる

子クラスの定義には、eatメソッドは無い

継承の例 animal.py

```
class Animal:
    japanese_class_name = '動物'

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

    def eat(self):
        print('ムシャムシャ')

class Dog(Animal):
    japanese_class_name = '犬'

    def bark(self):
        print('ワンワン')

class Bird(Animal):
    japanese_class_name = '鳥'
```

親クラス

子クラス

子クラス

- 共通する機能を親クラスにまとめると便利

継承にはその他にもいろいろな機能や意味がある。調べてみよう。