Records

Records are another example of a **data structure.**

The other data structure you need to know about are **Arrays** (in Python **Lists**)

Data structures allow more than one data item to be stored in a single variable

Records

Records are not used directly in Python.

They are essentially a way of storing database style information about an object (a thing).

Pseudocode uses them…

Records

Imagine we had a group of cats. Our cats might be represented using:

```
RECORD Cats
   name     : string
   colour   : string
   age      : integer
   alive    : Boolean
ENDRECORD
```

Note: you have to specify the data type

This would work really well as a database record with four fields - the primary key would be the name so only one cat could have each name

Records

We could then create a record for a cat:

```
catOne <- Cats(“Tiddles”, “Black”, 7, True)
```

And another cat:

```
catTwo <- Cats(“Clive”, “White”, 17, False)
```

And so on.

This assumes Cats all have the same attributes

Records

This would create a set of Cats that we could deal with like this:

```
IF catOne.alive = True
   OUTPUT catOne.name
   OUTPUT catOne.age
ELSE
   OUTPUT catOne.name + “is dead”
ENDIF
```

This uses dot notation (e.g. catOne.Name)
The Name comes from the record definition on slide 3

Records

You could even put your set of cats into an array like this:

```
catsArray <- [catOne, catTwo, catThree]
```