Records in Python

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

Records are not implemented in Python.

But there is a way around this…

Records in Python

Records can be implemented in Python using **classes** - this is an advanced programming technique using an idea called **object oriented programming**

Records in Python

```
class Cats:
   def __init__(self, name, colour, age, alive):
      self.name = name
      self.colour = colour
      self.age = age
      self.alive = alive
```

This constructs the class - allows Python to understand what Cats are.

The constructor always uses `__init__` and `self` must be the first attribute. `self` is then used to create the other attributes.

Records in Python

Individual objects of the class can then be created using code such as:

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

You can then use things like:

```
print(catOne.name)
catOne.age = 8
if catOne.age > 10:
if catOne.age > catTwo.age:
```

Records in Python

And you can create an array of cats:

```
catArray = [catOne, catTwo, catThree]
```

You can then iterate over the array:

```
for aCat in catArray:
   print(aCat.name)
```

Records in Python

What attributes might a class called Cars have?

How would you create those attributes?

How would you create a car object?