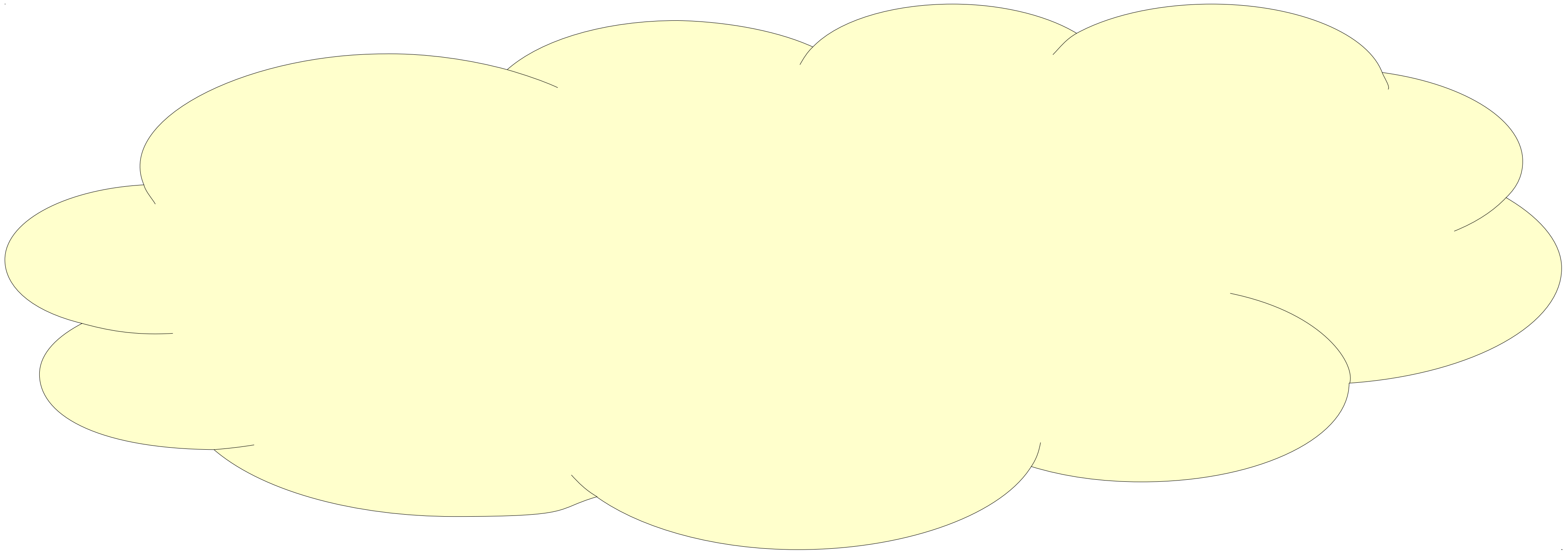
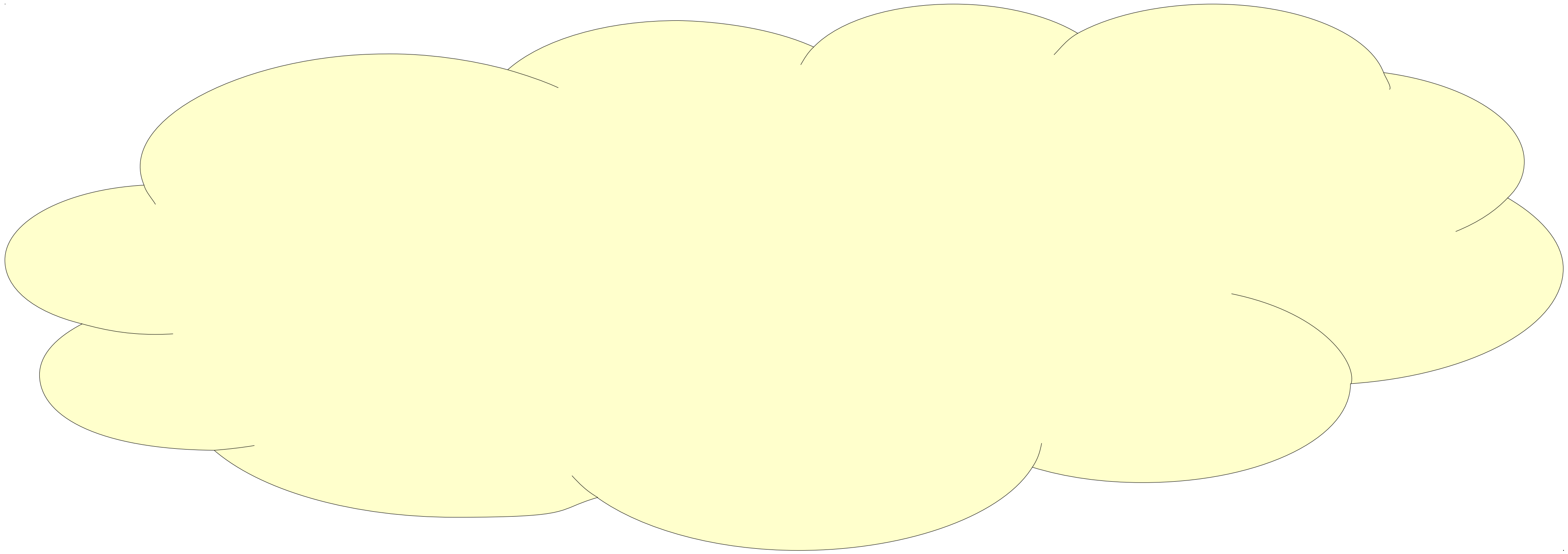
HashMap

Not All Data is Linear

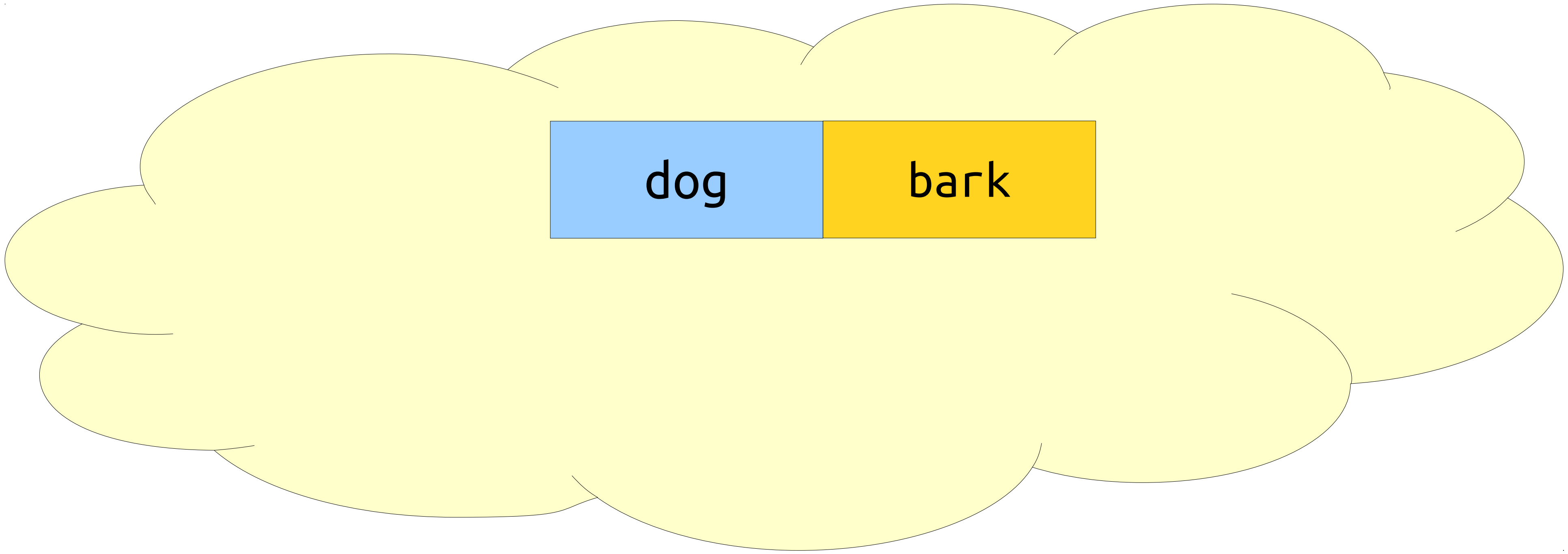
```java
HashMap<String, String> animals =
     new HashMap<String, String>();
```



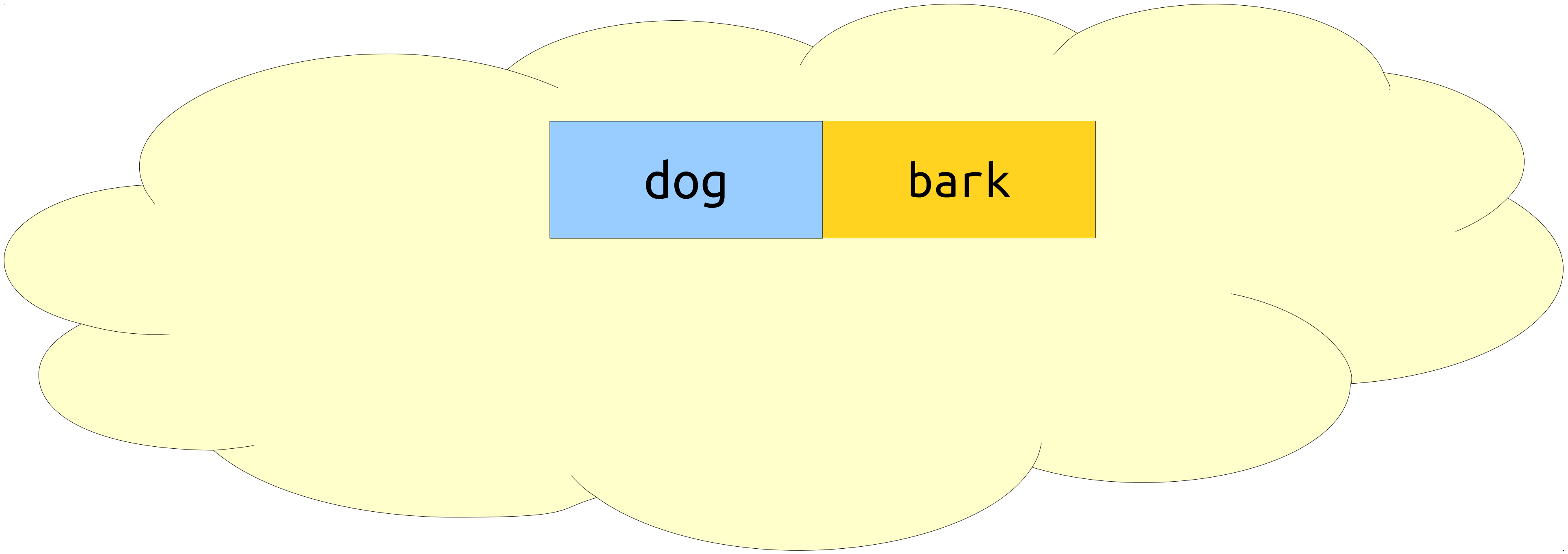
```
HashMap<String, String> animals =
     new HashMap<String, String>();
```



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
```



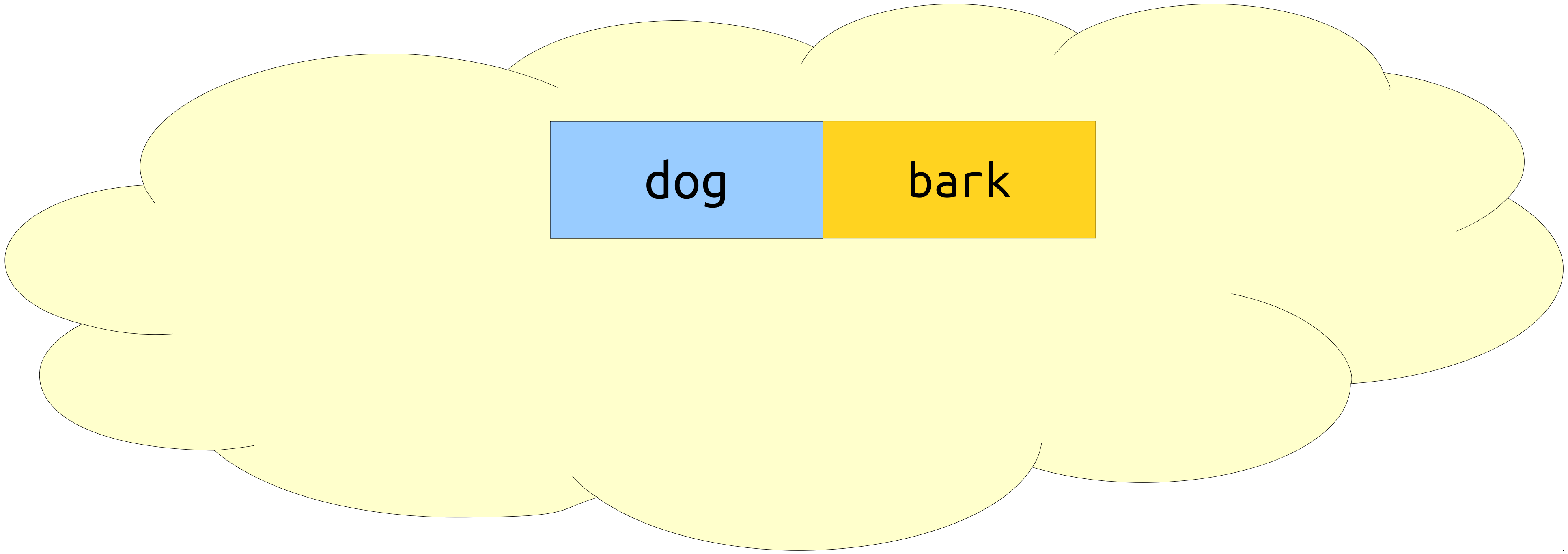
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
```



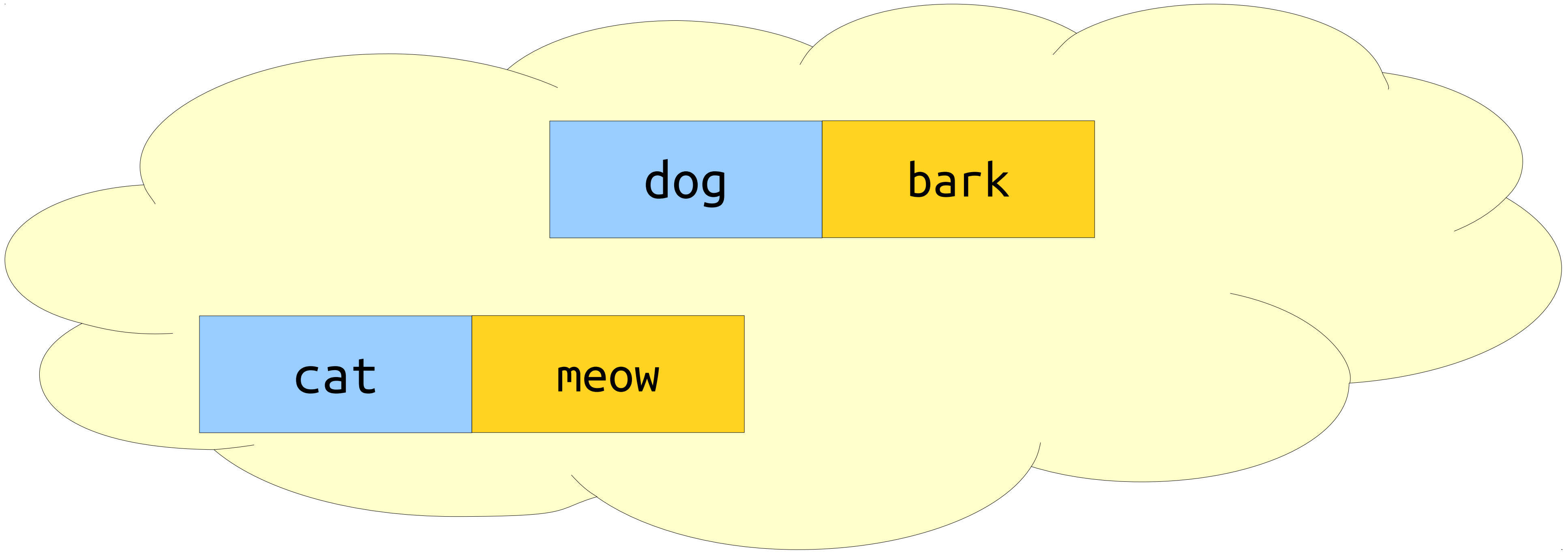
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
```

To add a key/value pair to a **HashMap**, use the syntax

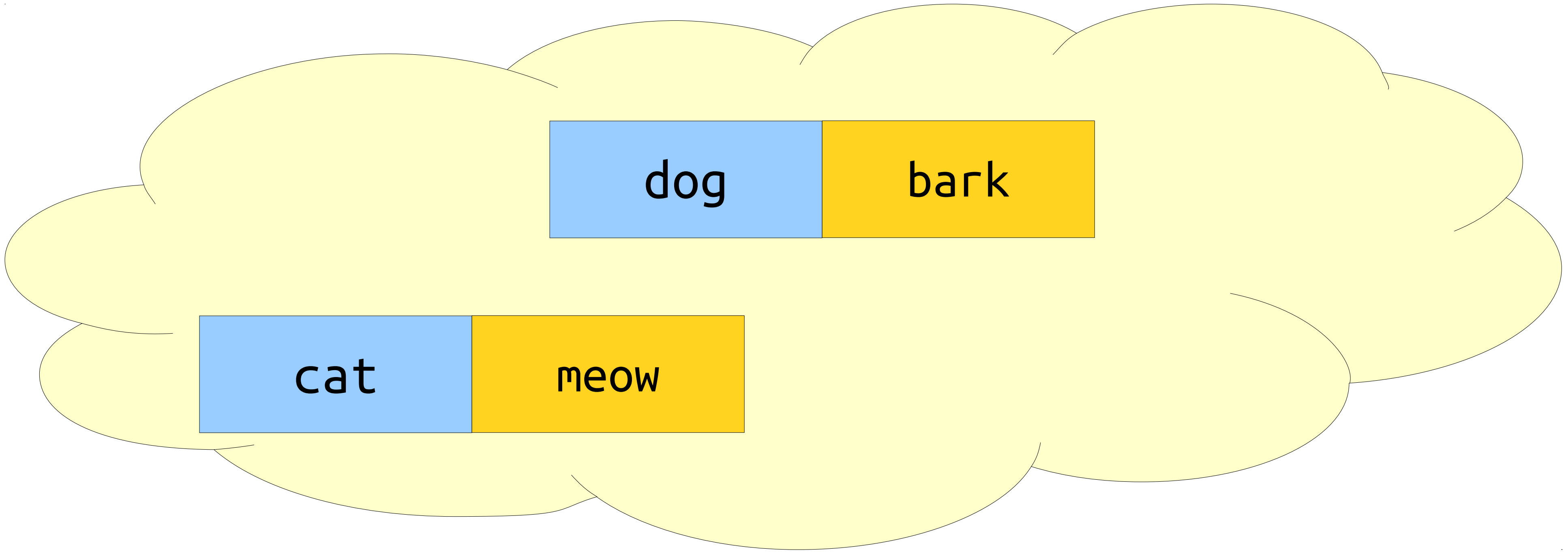
***map*.put(*key*, *value*)**



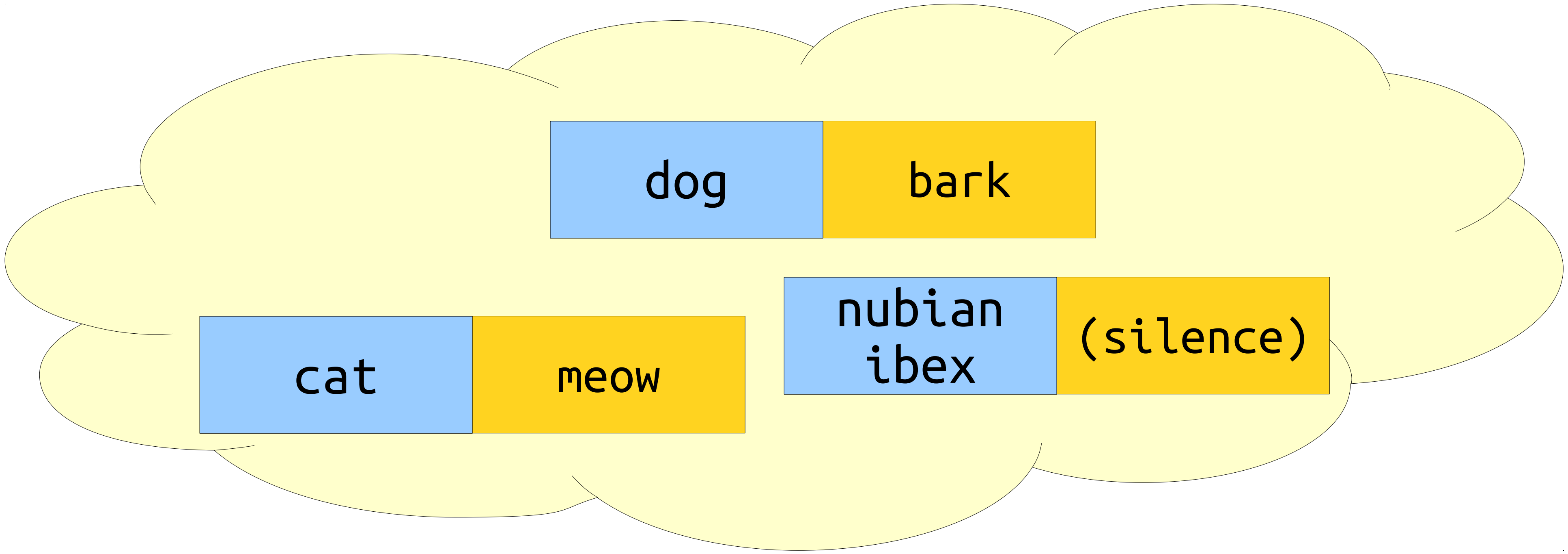
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
```



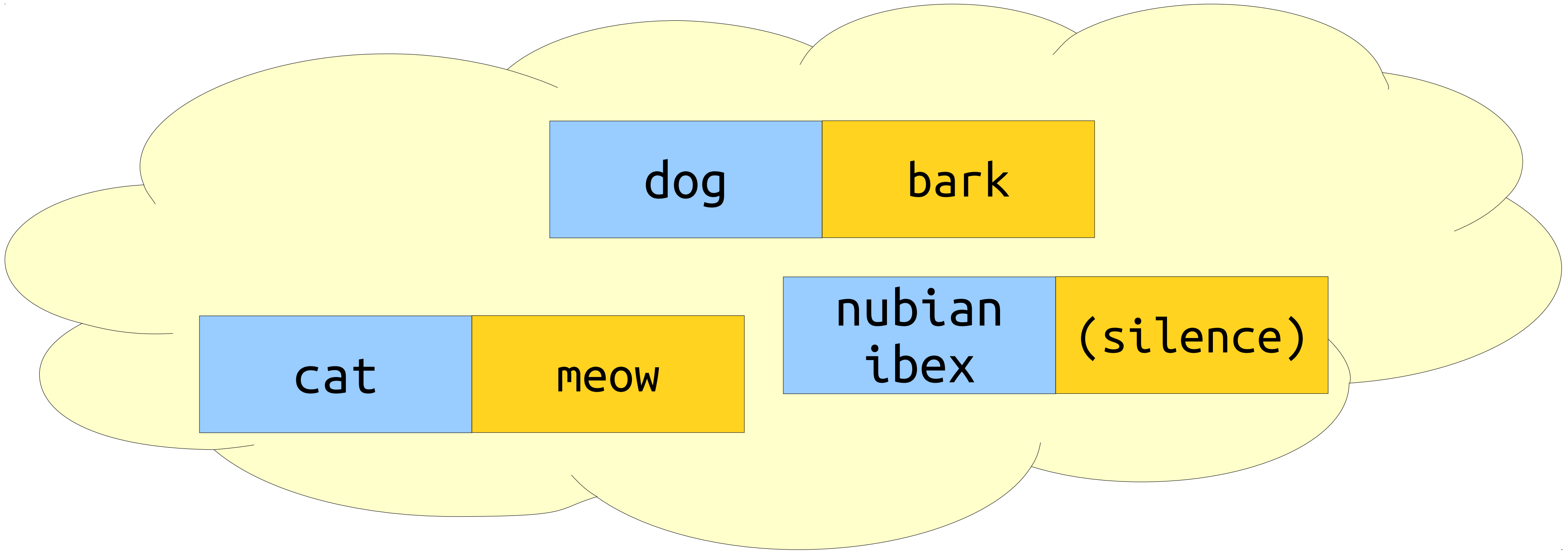
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
```



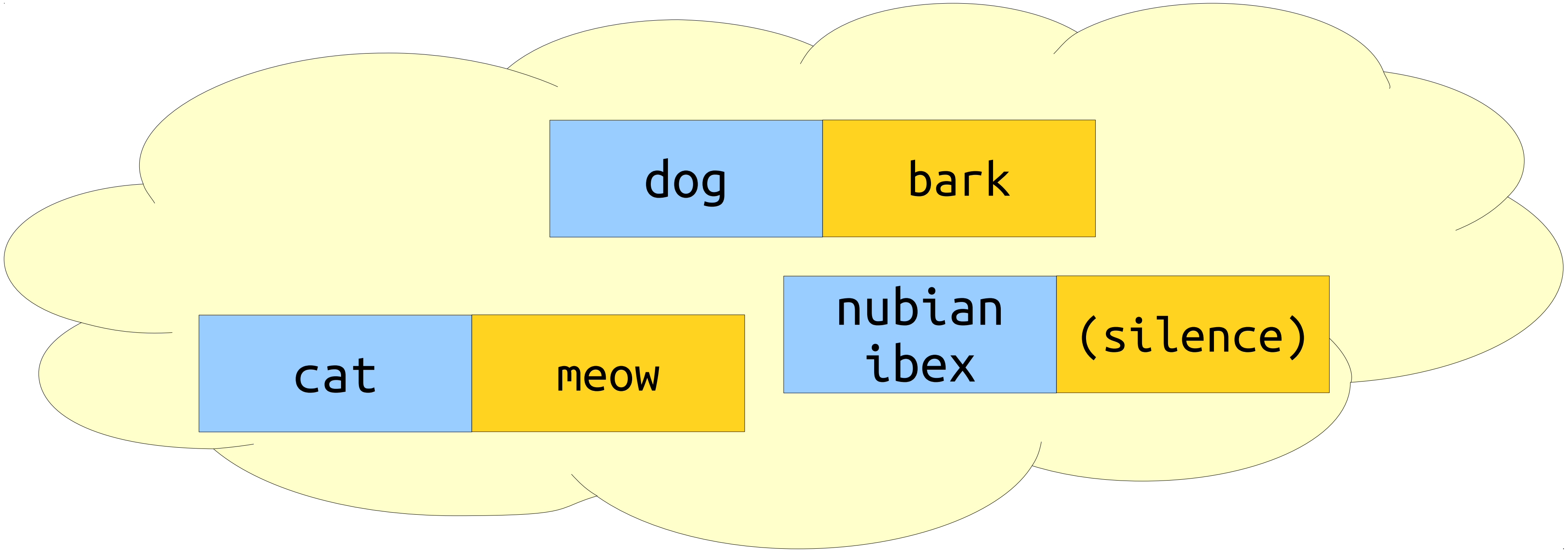
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
```



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
```



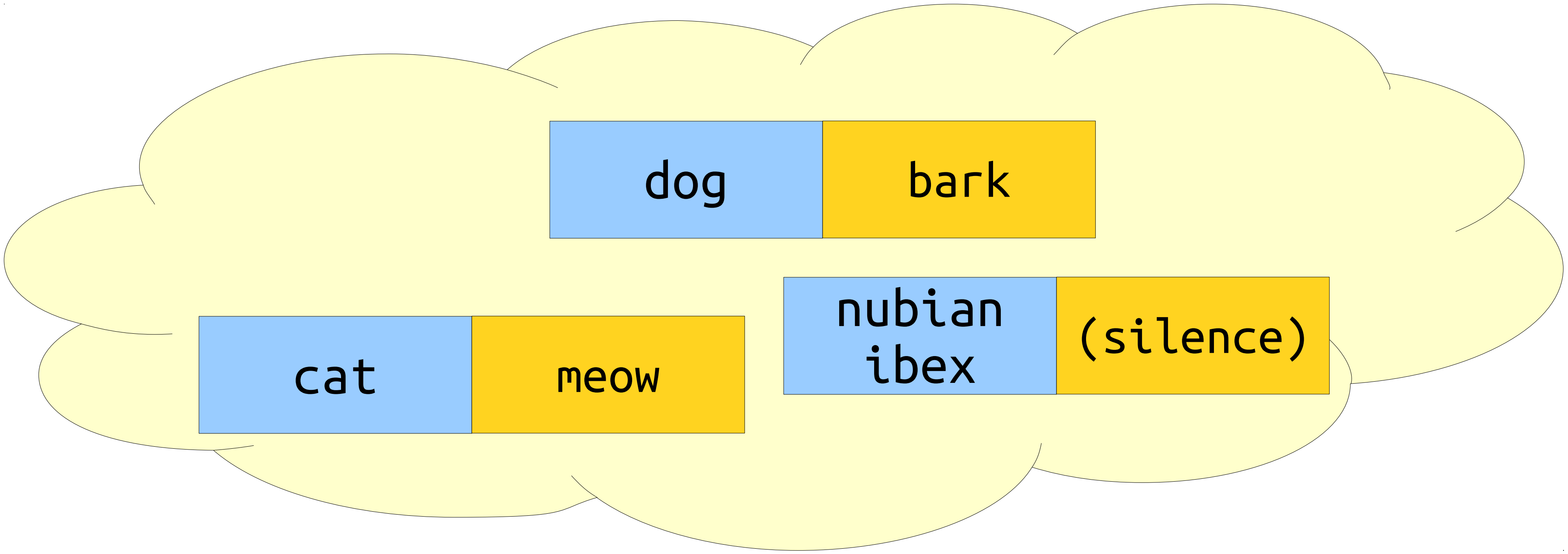
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");
```



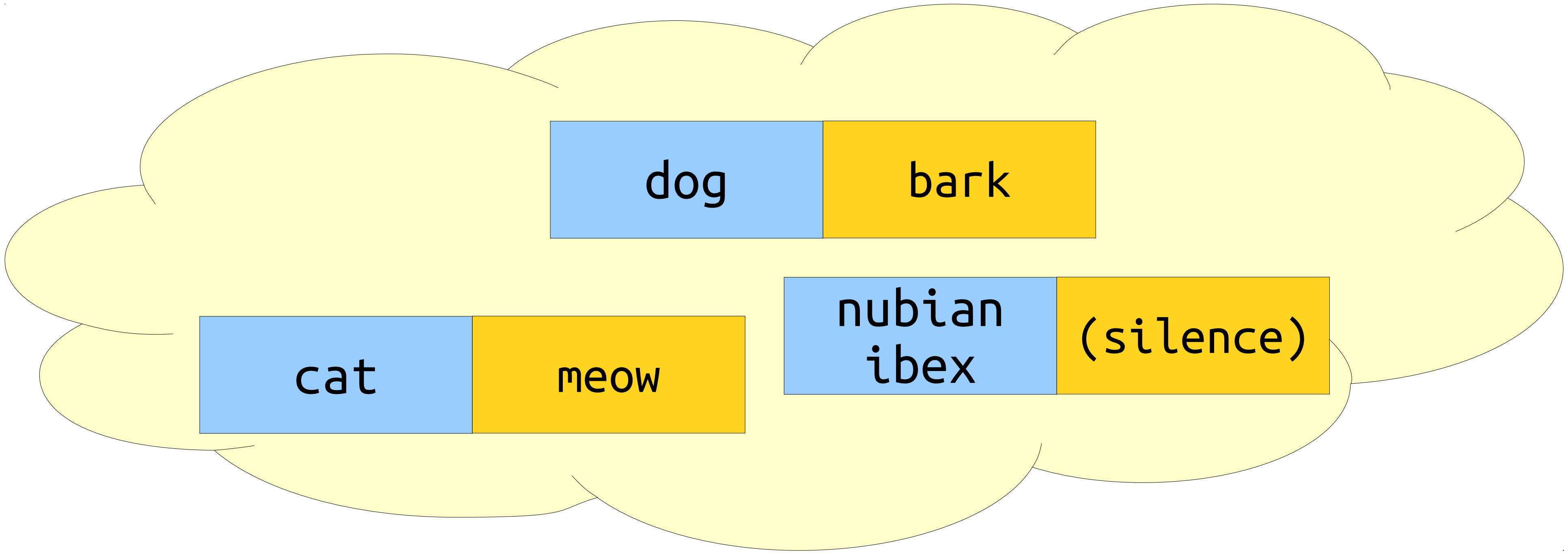
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(s
animals.get("dog");
```

To look up the value associated with a key:

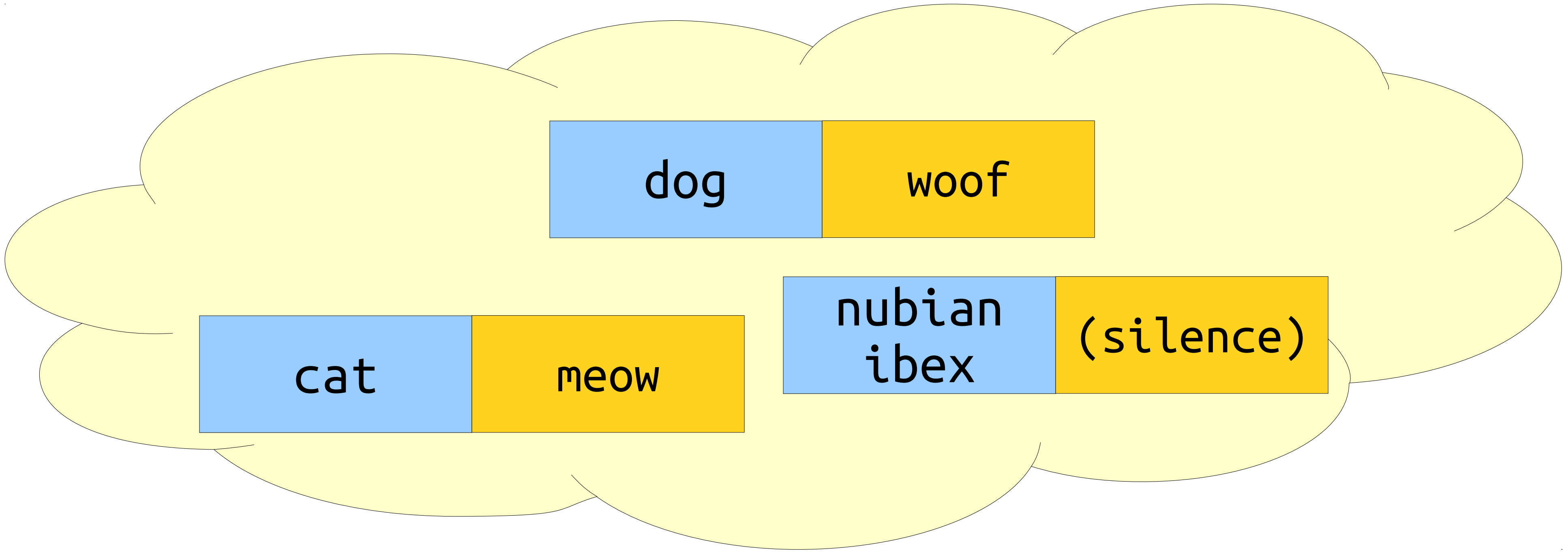
***map*.get(*key*)**



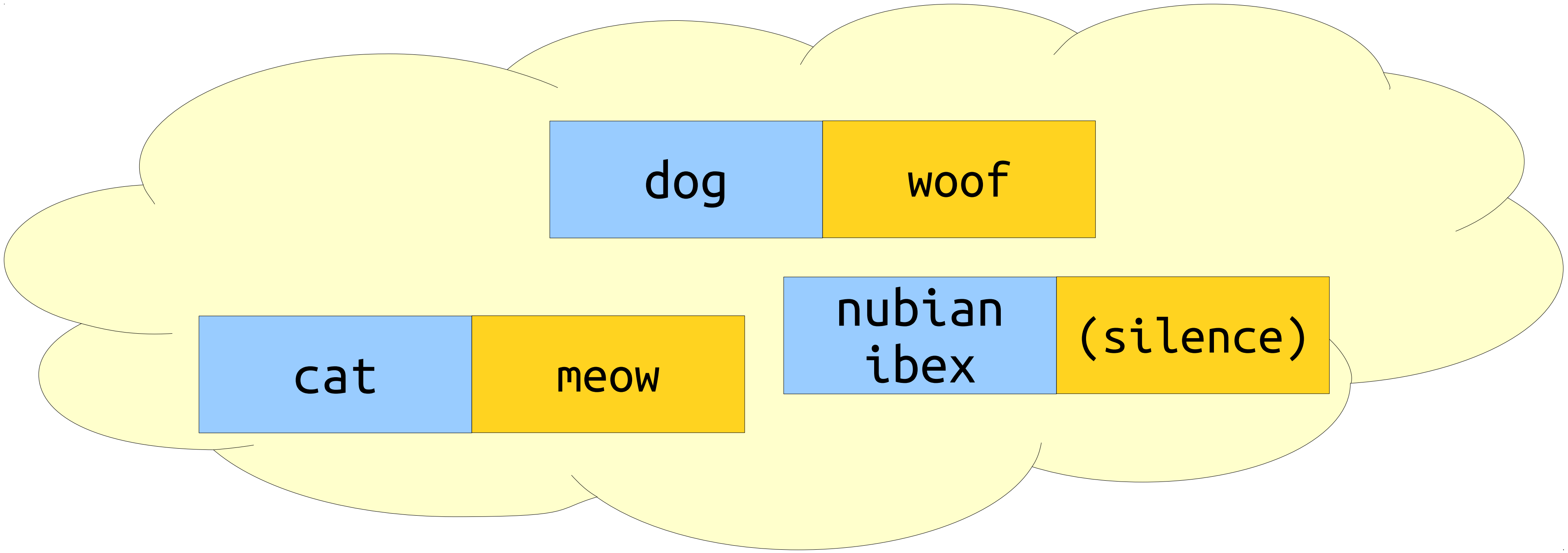
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");         // Returns "bark"
```



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");          // Returns "bark"

animals.put("dog", "woof");
```

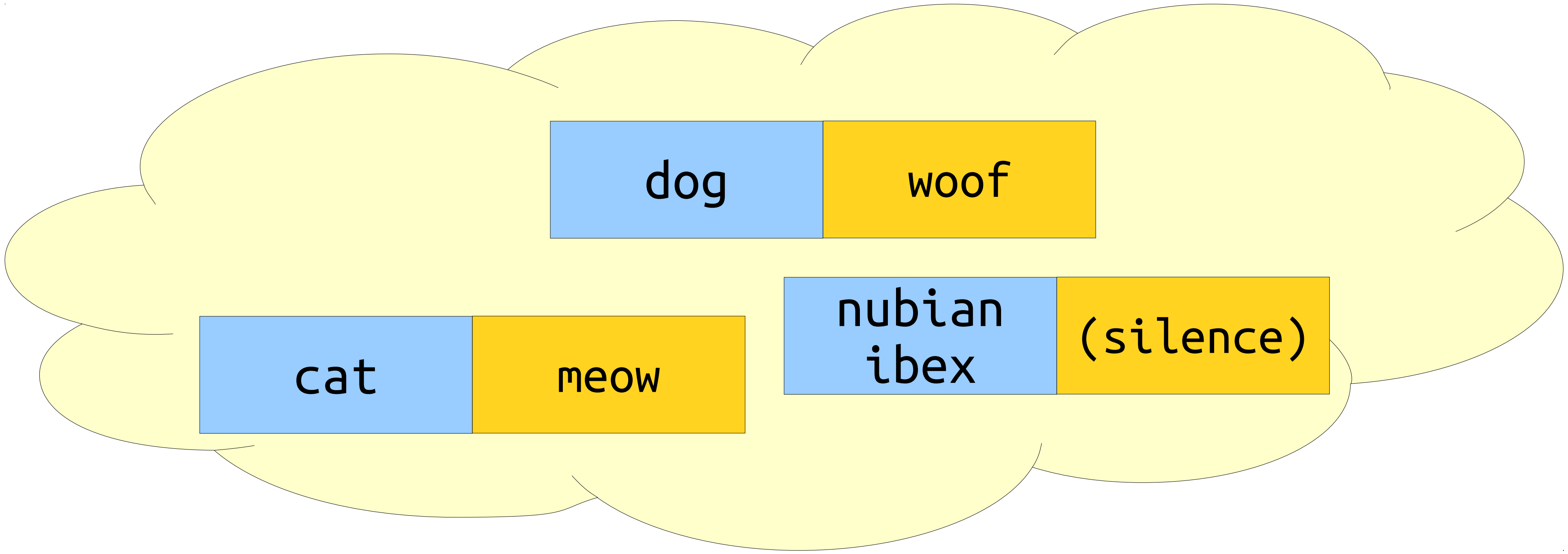


```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");         // Returns "bark"

animals.put("dog", "woof");
```

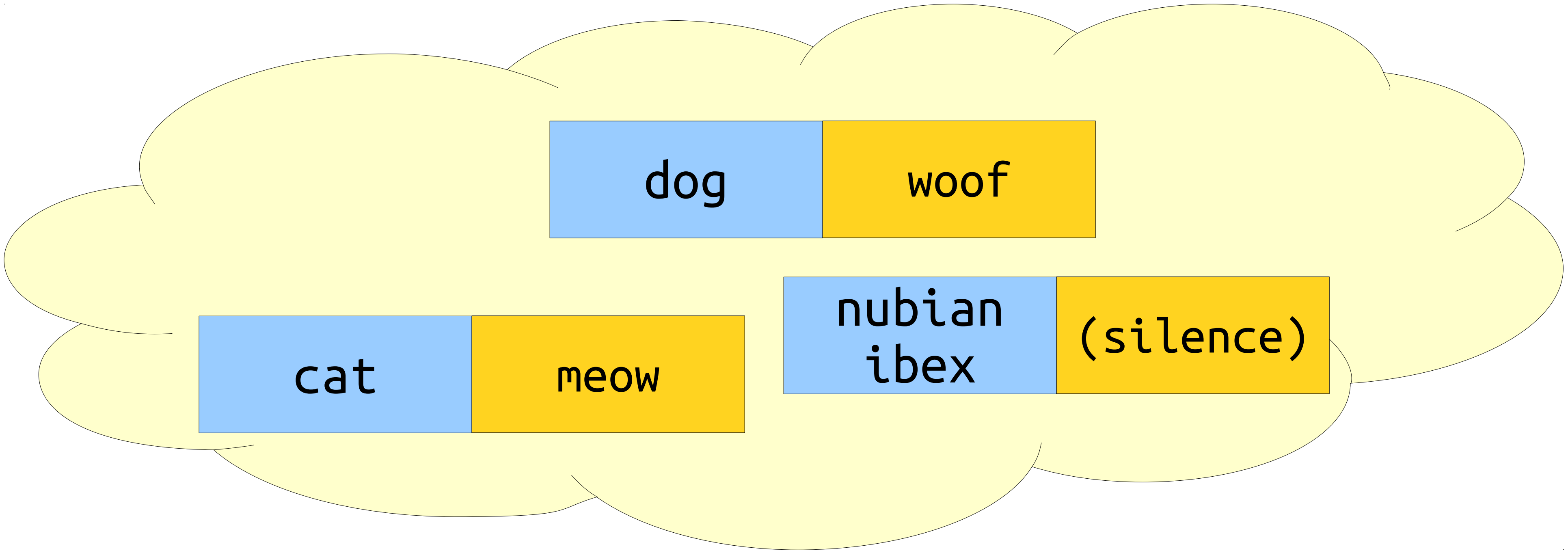


```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(sil
animals.get("dog");          // F

animals.put("dog", "woof");
```

If you **put** a key/value pair where the key exists, the old value is replaced.

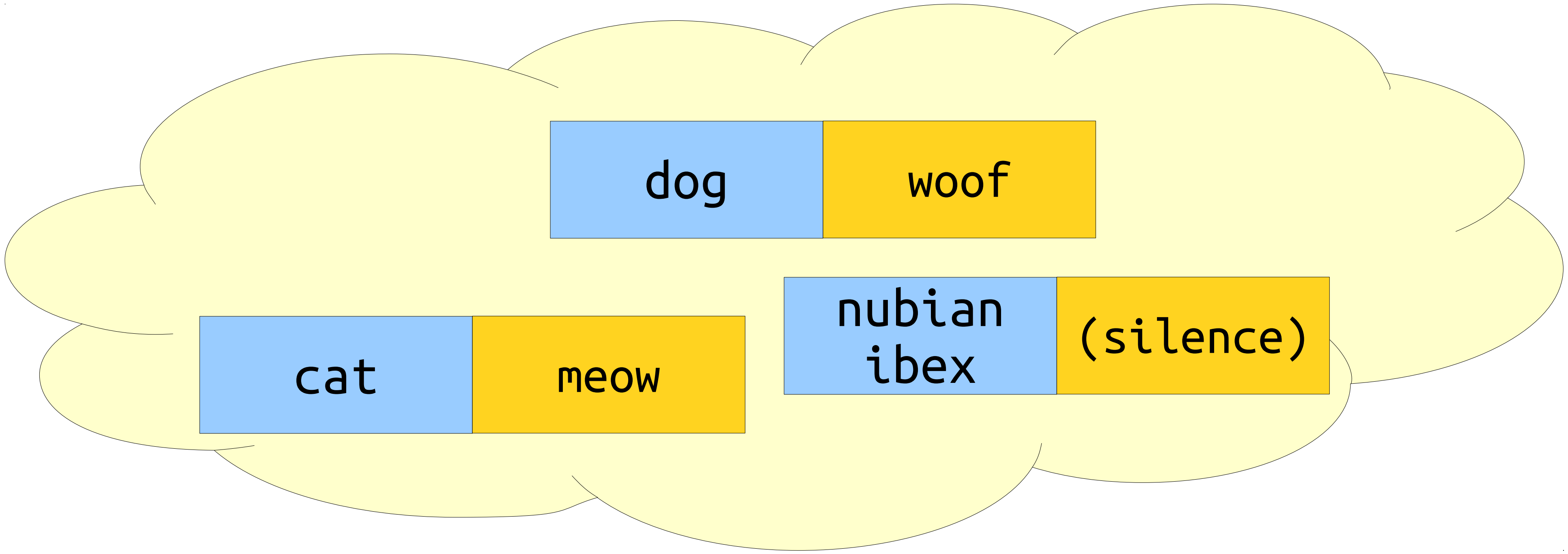


```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");         // Returns "bark"

animals.put("dog", "woof");
animals.get("fox");
```

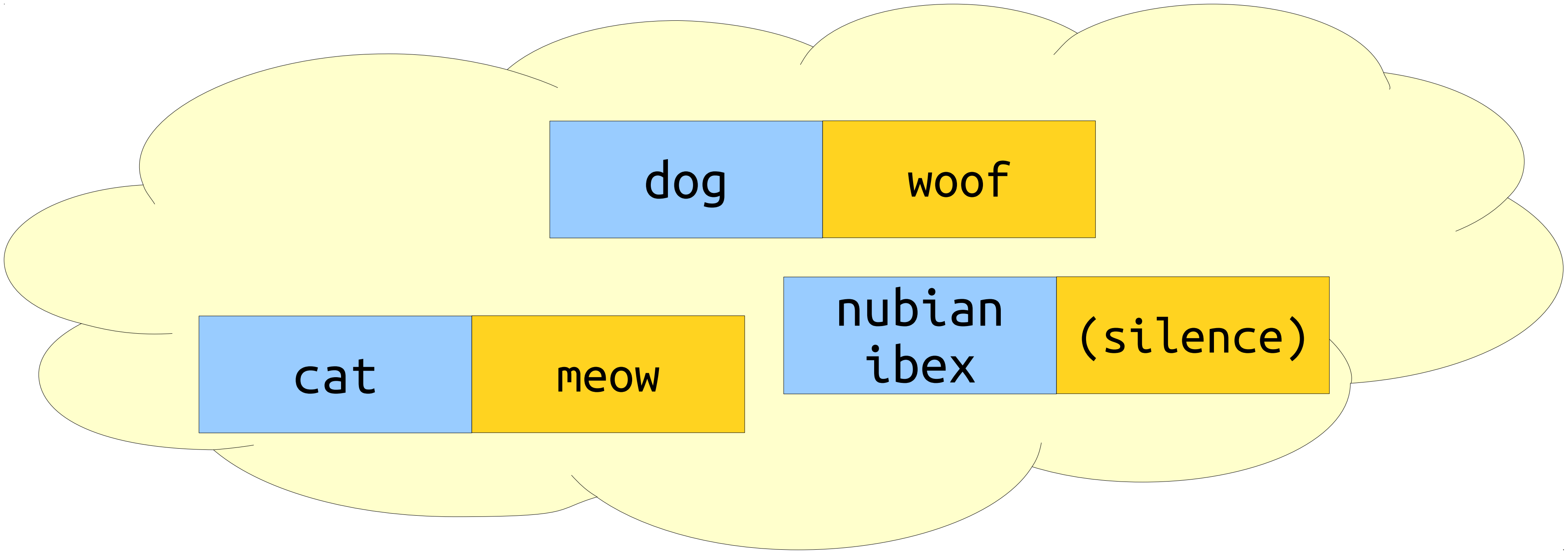


```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(si
animals.get("dog");          //

animals.put("dog", "woof");
animals.get("fox");
```

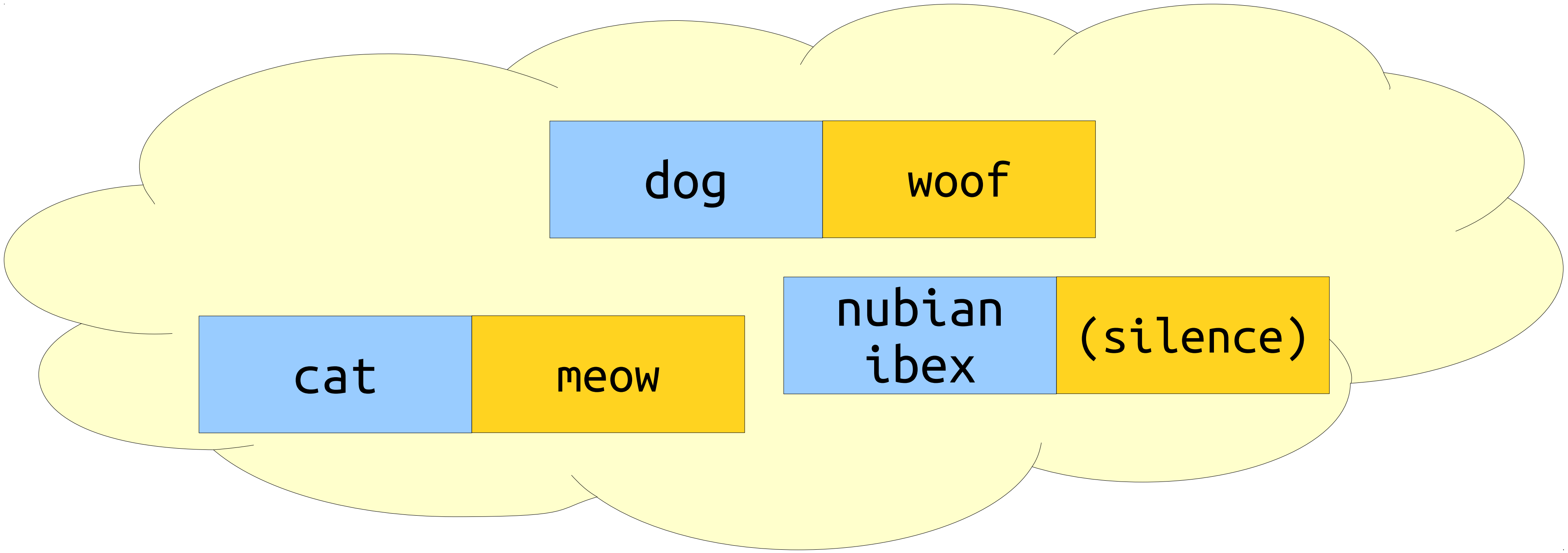
If you **get** a key that isn't in a map, the method returns **null**.



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");           // Returns "bark"

animals.put("dog", "woof");
animals.get("fox");           // Returns null
```



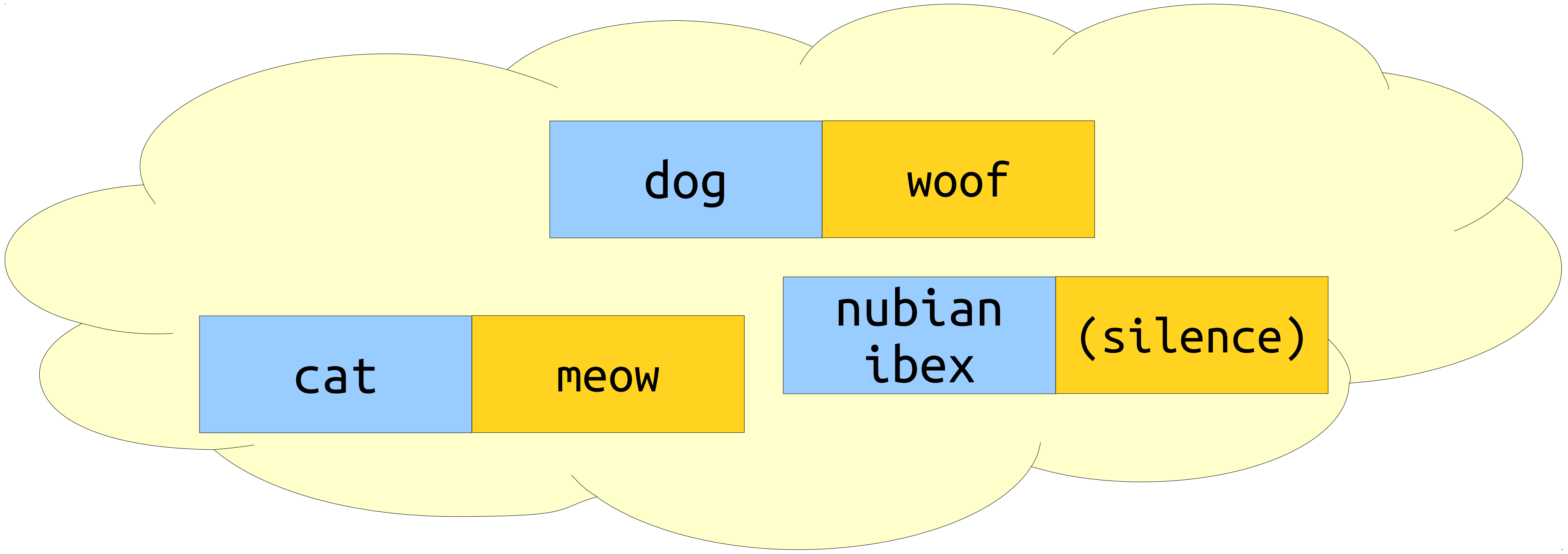
```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");            // Returns "bark"

animals.put("dog", "woof");
animals.get("fox");            // Returns null
animals.containsKey("cat");
```



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(si
animals.get("dog");          //

animals.put("dog", "woof");
animals.get("fox");          //
animals.containsKey("cat");
```

You can check whether a key exists in the map:

***map*.containsKey(*key*)**



```
HashMap<String, String> animals =
     new HashMap<String, String>();

animals.put("dog", "bark");
animals.put("cat", "meow");
animals.put("nubian ibex", "(silence)");
animals.get("dog");           // Returns "bark"

animals.put("dog", "woof");
animals.get("fox");           // Returns null
animals.containsKey("cat"); // Returns true
```

Basic `HashMap` Operations

- HashMap has two type arguments:

  `HashMap<`***KeyType***`, `***ValueType***`>`

- To insert a key/value pair:

  map`.put(`***key***`, `***value***`)`

- To look up the value associated with a key:

  map`.get(`***key***`)`

- To check whether a key exists:

  map`.containsKey(`***key***`)`

Making `HashMap` Shine

Exploring the US

Time-Out for Announcements!

Midterms Graded

- Midterms graded, available for pickup in a filing cabinet near Keith's office.
  - Check the email for details!
- If you'd like to submit your exam for a regrade, attach a regrade request form (available online) to the front of your exam and hand it to Keith or Alisha.
  - Deadline: Wednesday at 4:15PM.

Assignment 6

- Assignment 6 (Array Algorithms) is due Friday.
- **Recommendation:** Complete all three parts of the assignment by Wednesday – the LaIR will be way less crowded!

Second Midterm Exam

- The second midterm exam is next **Tuesday, March 3** from **7PM – 10PM**.
- Same format as the first exam:
  - Closed-book, closed-computer, open-one-double-sided-8.5”×11”-sheet-of-notes.
  - We'll be providing a reference sheet with common methods, which will be available for preview on the course website.
- Practice exam is this **Thursday, February 26** from **7PM – 10PM** in **Cubberly Auditorium**.
- Need to take the exam at an alternate time? Contact Alisha ASAP. Please also let us know why you need to take the exam at an alternate time.

Back to CS106A!

Making Music

The Keyboard File Format

note-file-name
x
y
width
height
is white key?

The xkcd Color Survey

The xkcd Color Survey

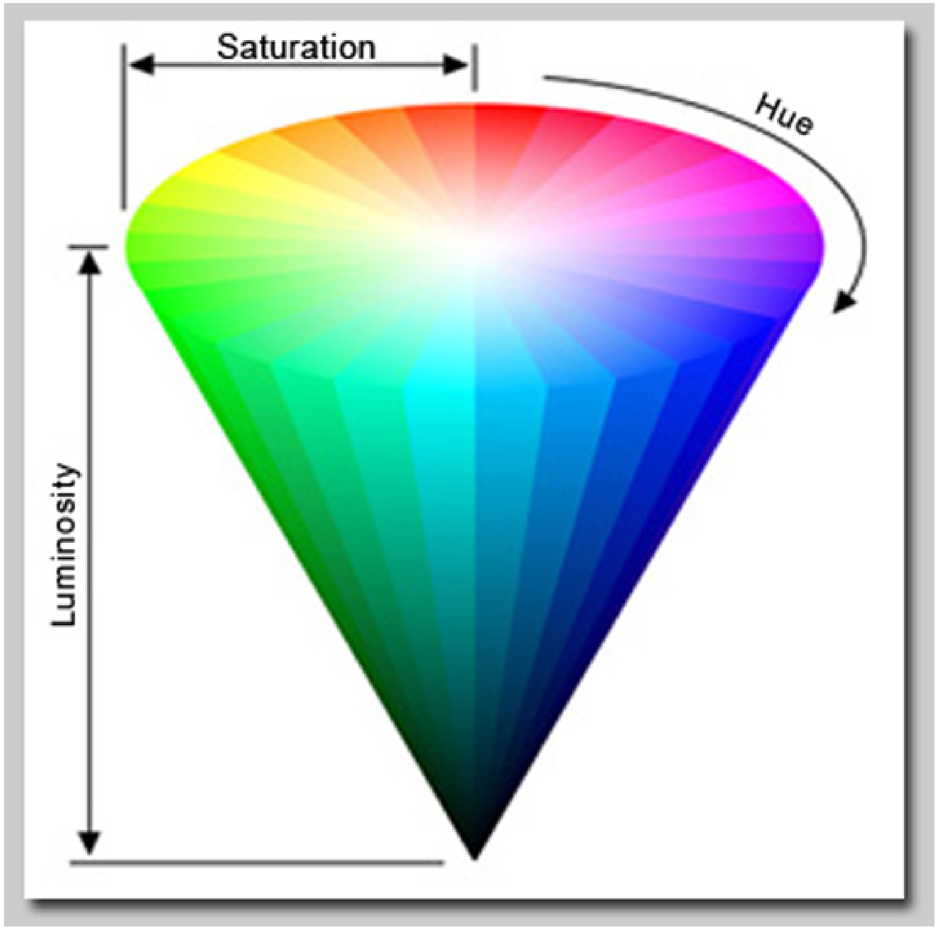
- Volunteers (online) were shown a randomly-chosen color and asked to name the color.
- The result is (after filtering) about 2.8 million RGB triplets and their names.
- What do people think the colors are?

The Color File Format

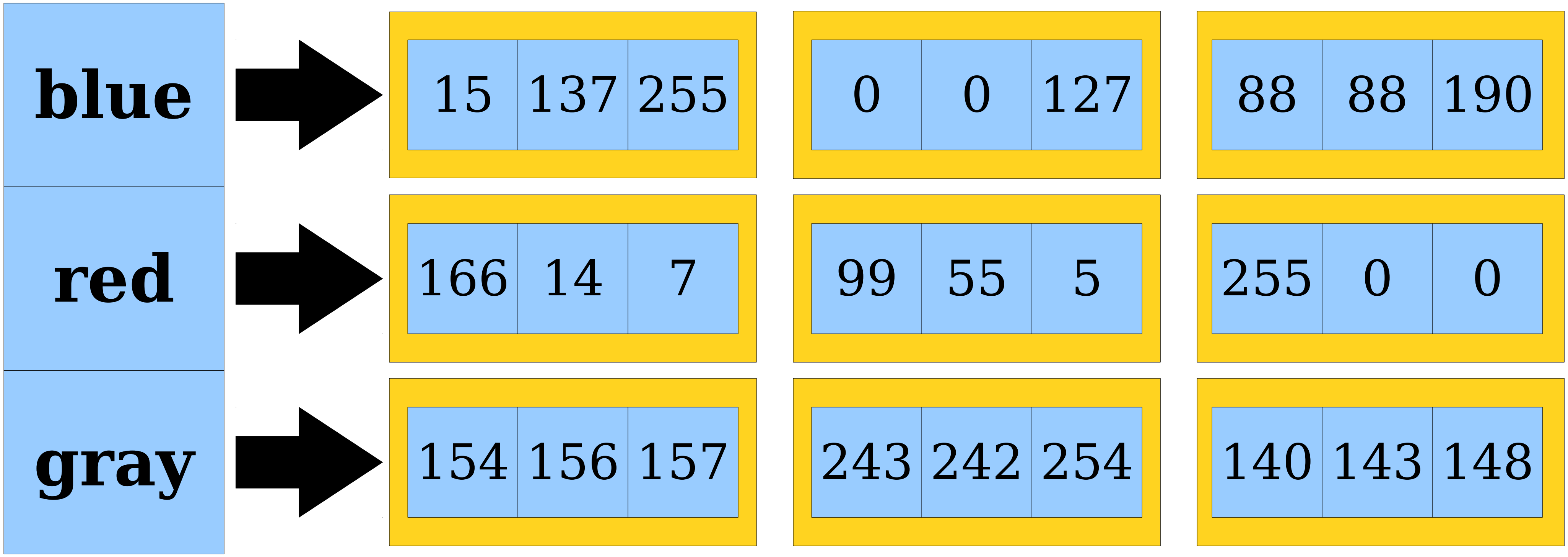
color-name
red
green
blue

Displaying Colors

- HSB color format:
  - Choose the ***hue*** (which color), ***saturation*** (how intense), and ***brightness*** (absolute brightness).
  - Each choice in the range from 0.0 to 1.0.

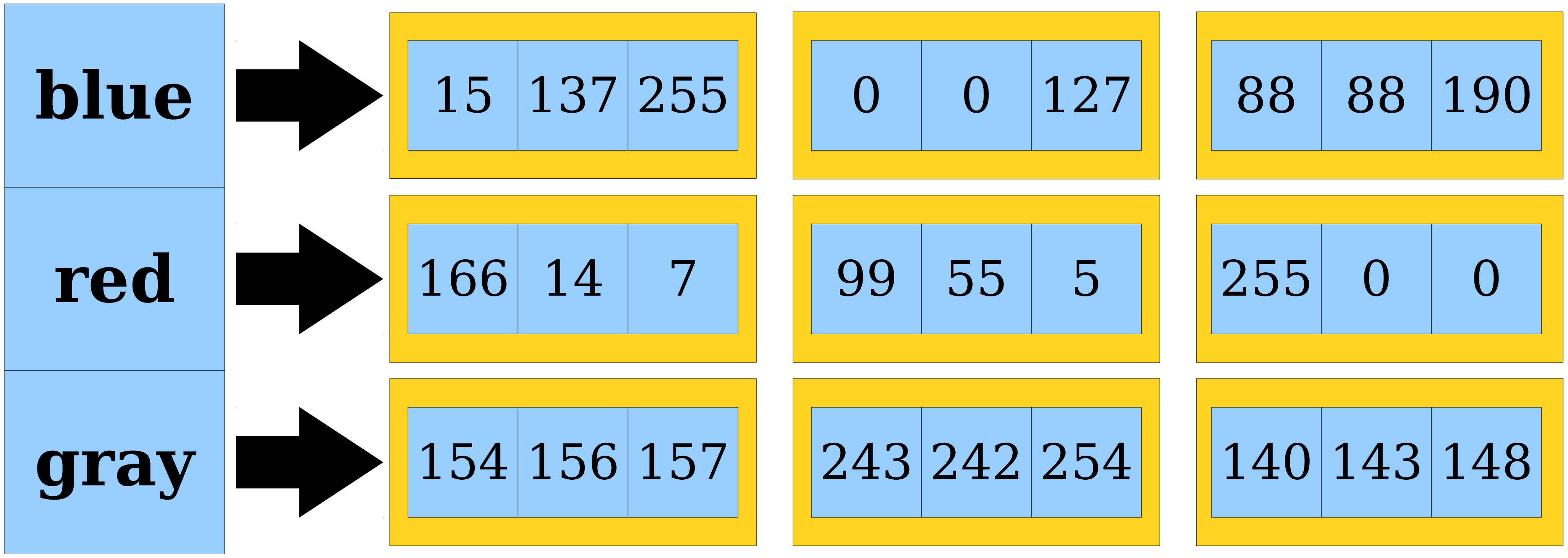


How to Structure the Data?



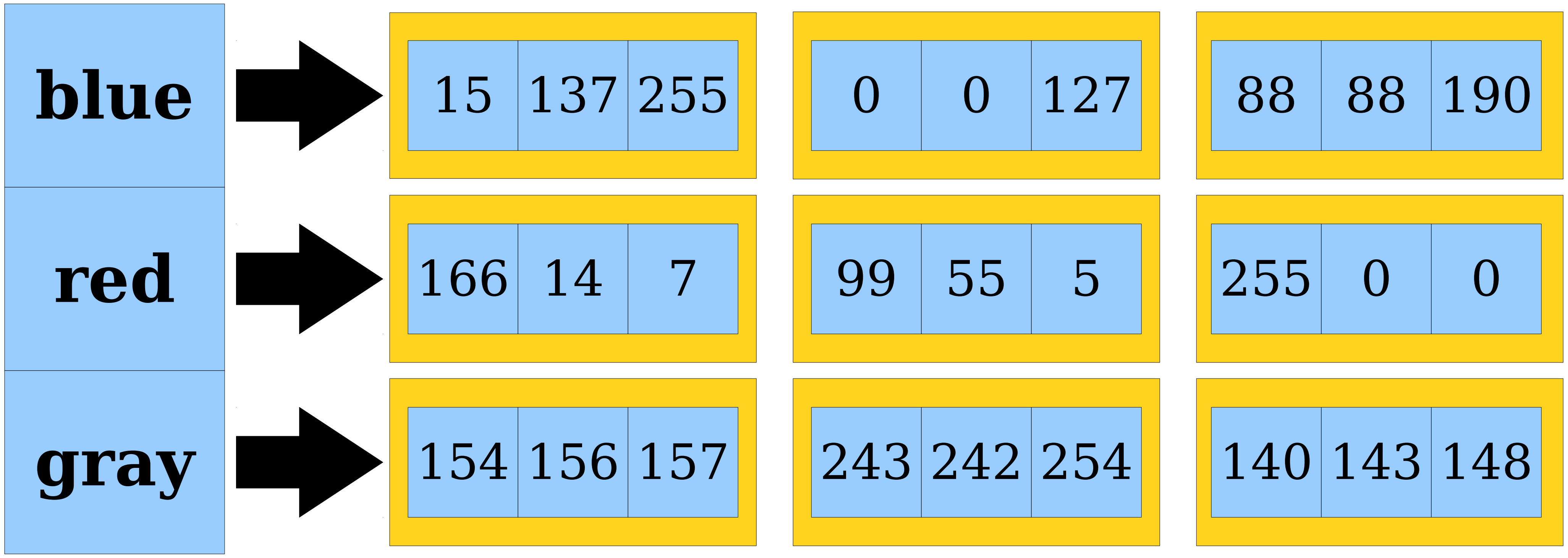
associate each color name with a list of colors

How to Structure the Data?



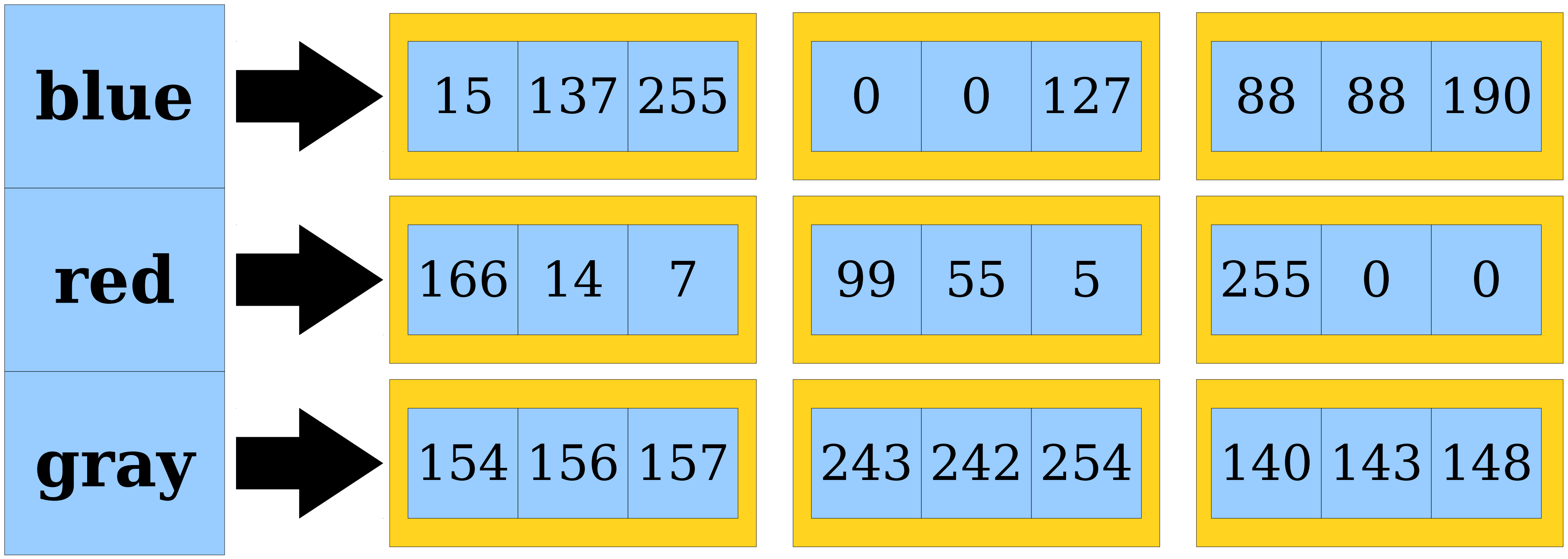
HashMap<*color name* , *list of colors*>

How to Structure the Data?



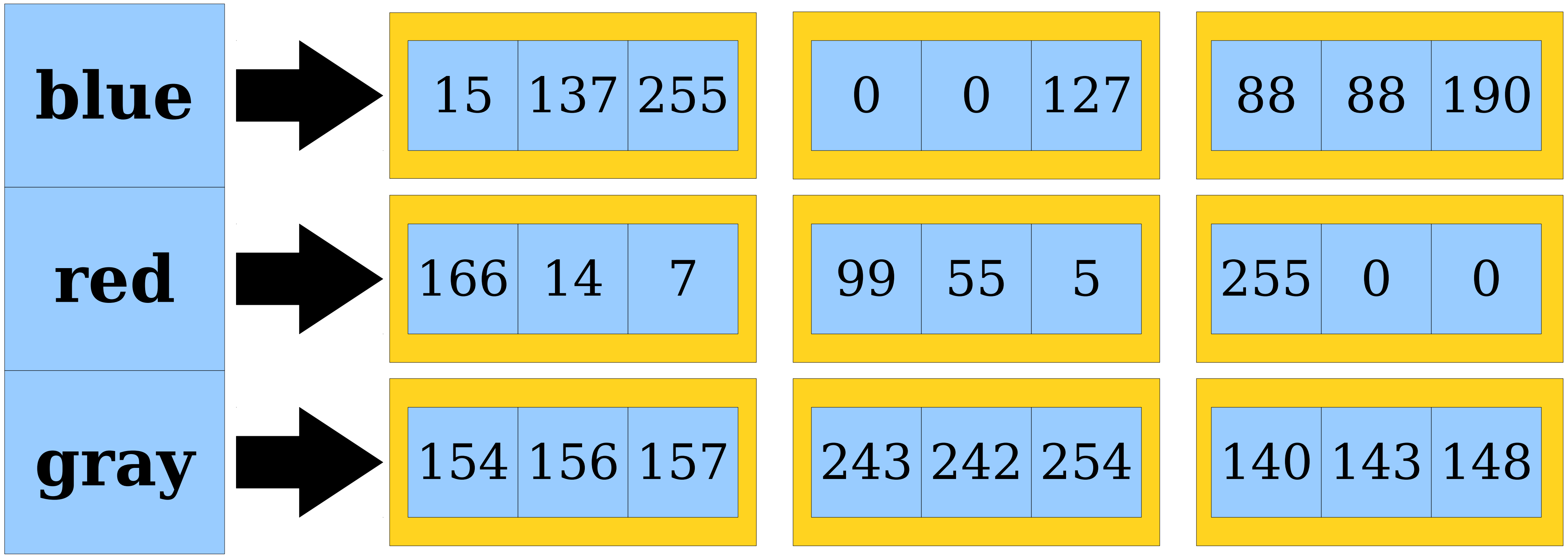
`HashMap<String,` ***list of colors*****`>`**

How to Structure the Data?



HashMap<String, ArrayList<*color*>>

How to Structure the Data?



`HashMap<String, ArrayList<Color>>`

For More Information

http://blog.xkcd.com/2010/05/03/color-survey-results/