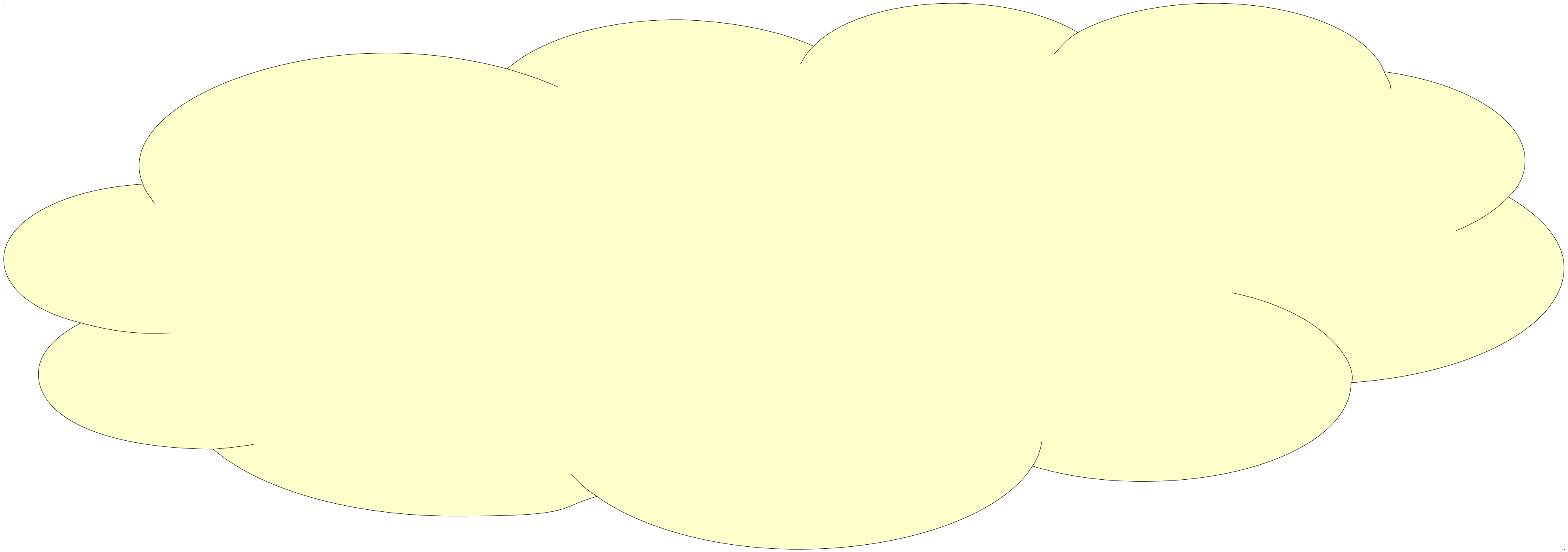
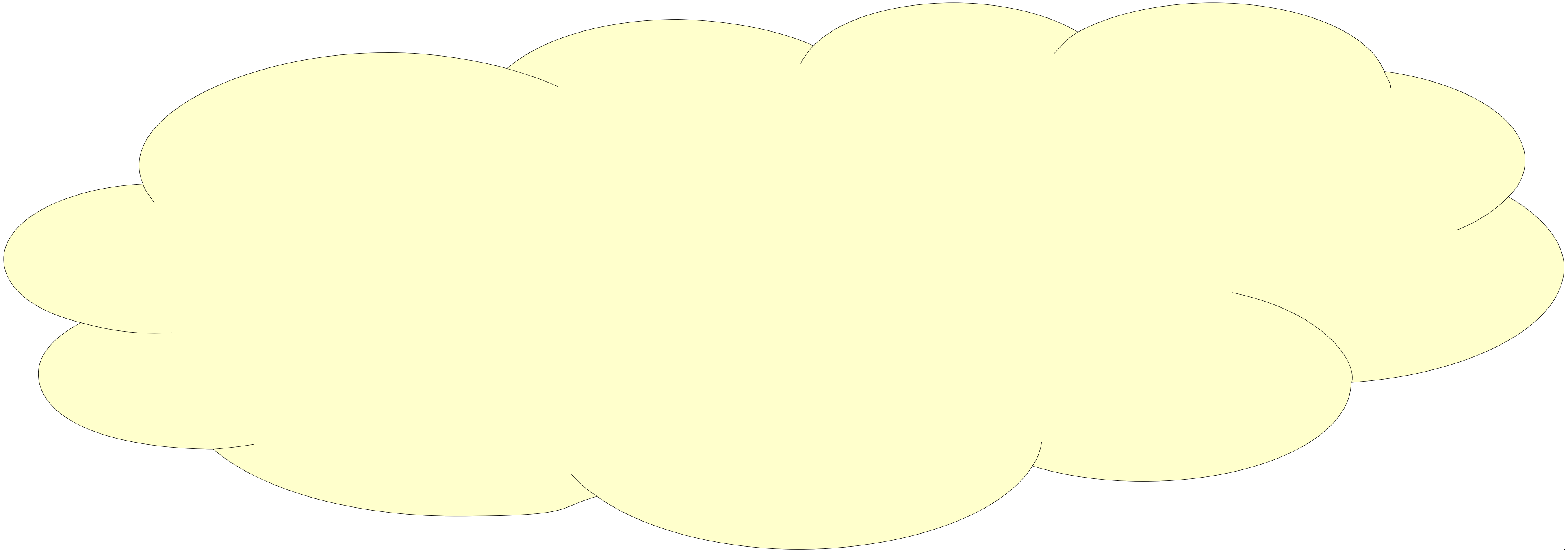
HashMap

Not All Data is Linear

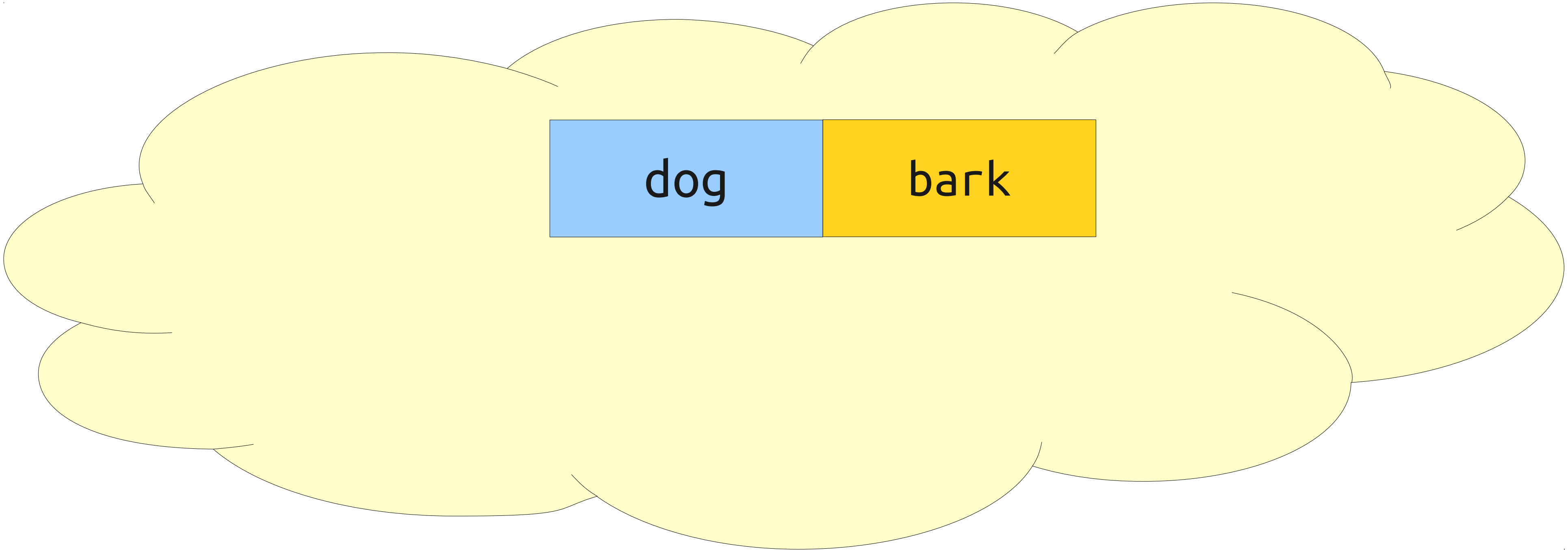
```
HashMap<String, String> animals =
     new HashMap<String, String>();
```



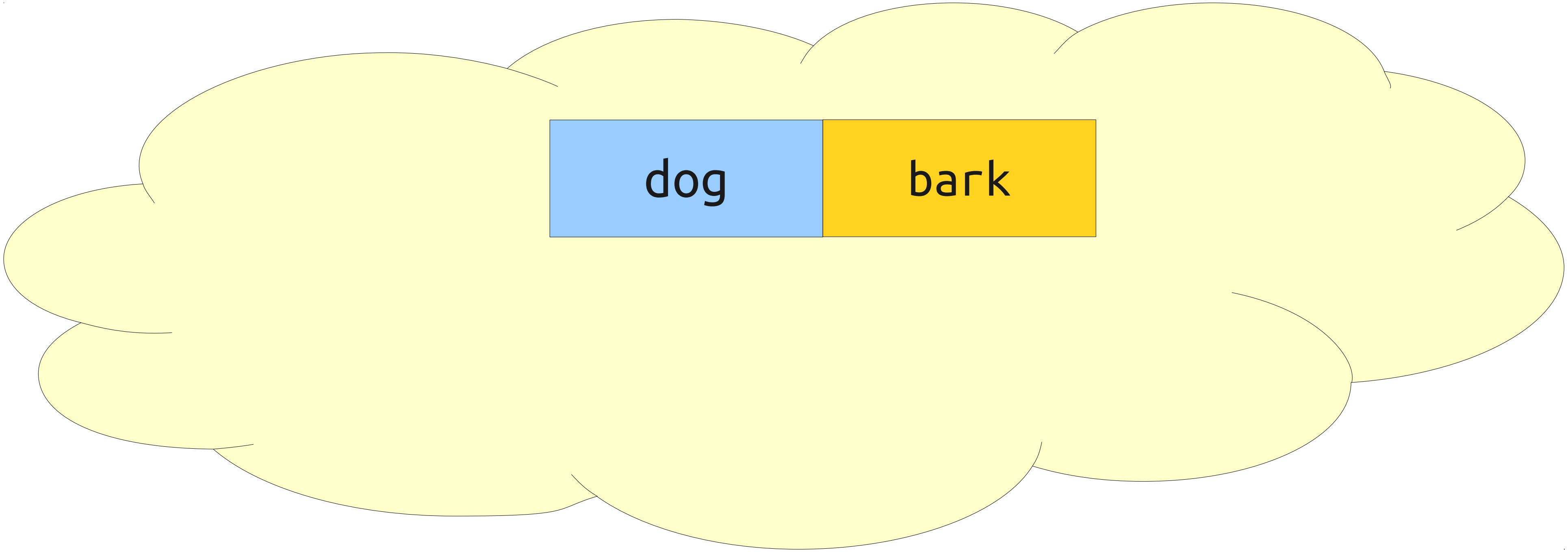
```
HashMap<String, String> animals =
     new HashMap<String, String>();
```



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

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



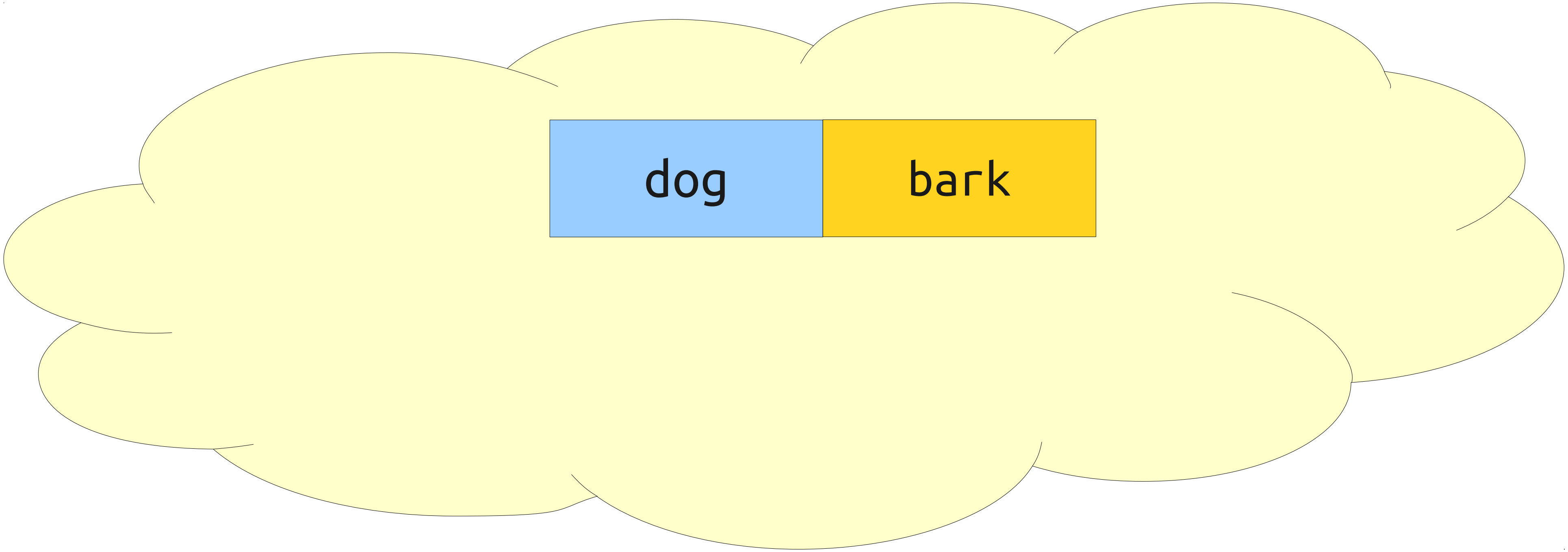
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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



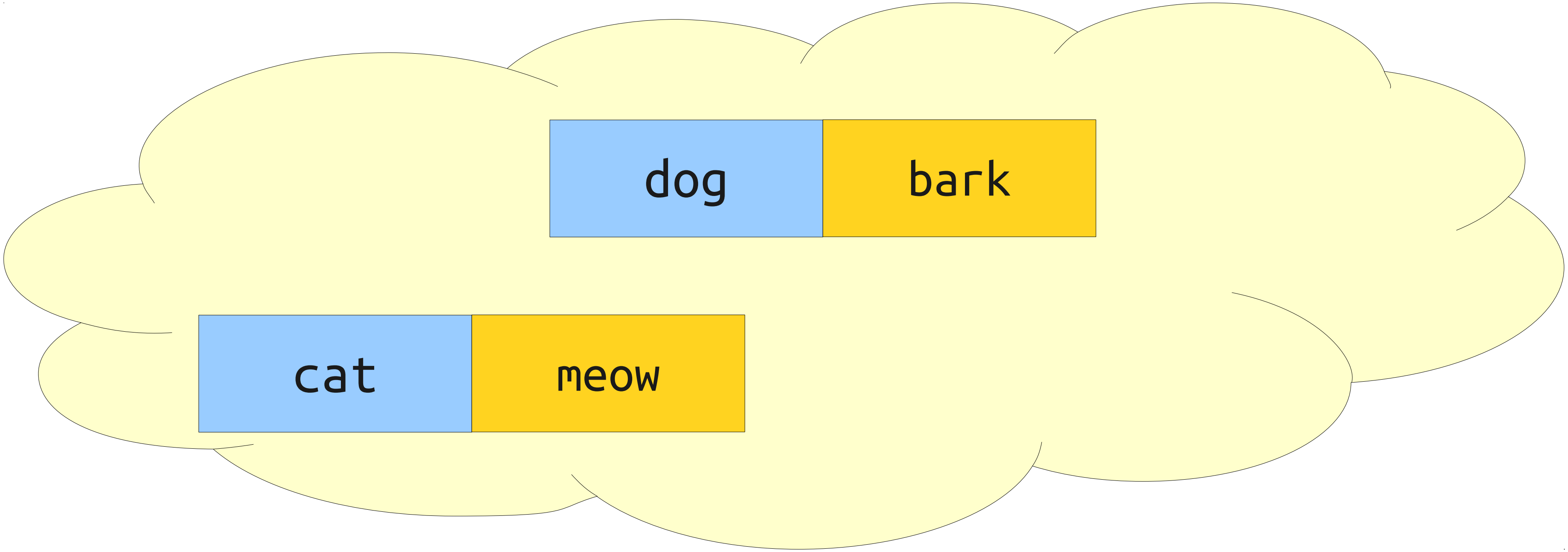
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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

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

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



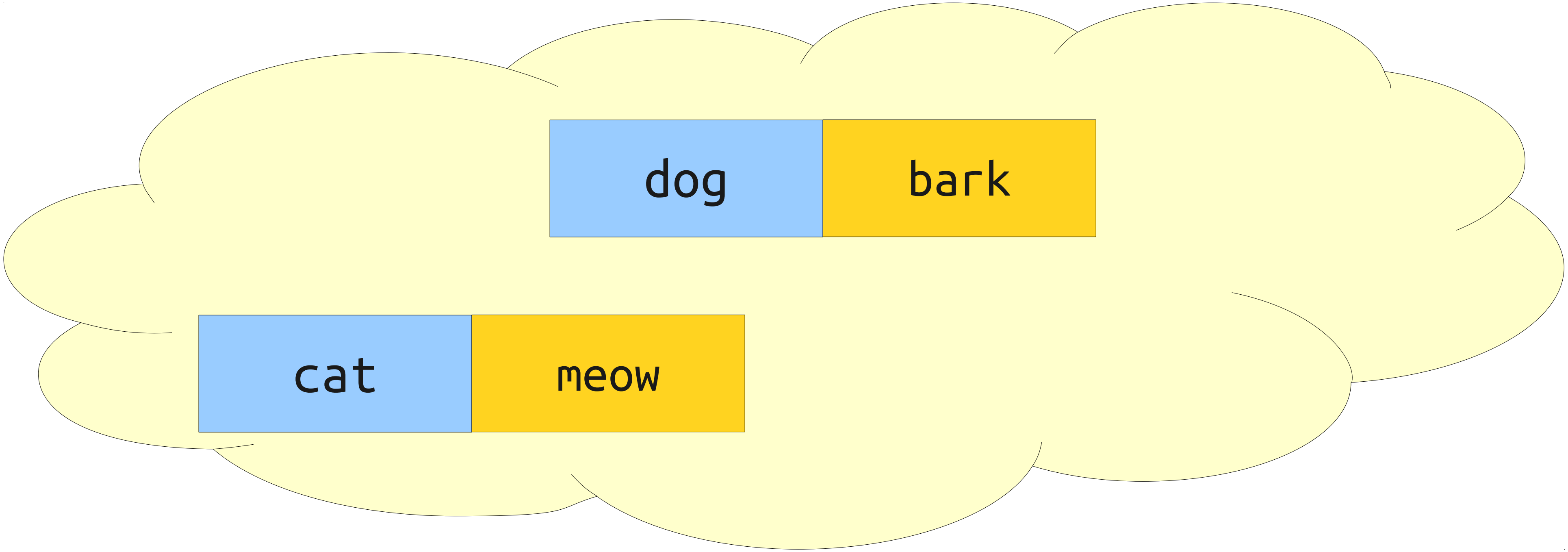
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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



dog bark

cat meow

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

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



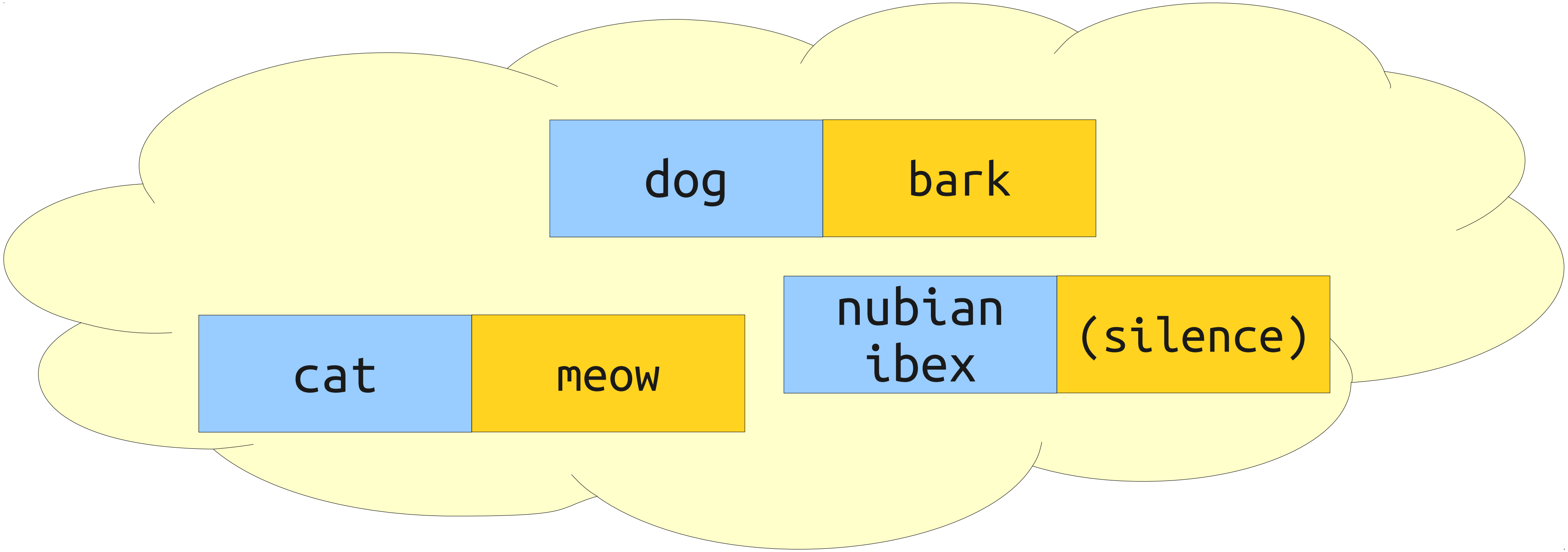
dog

bark

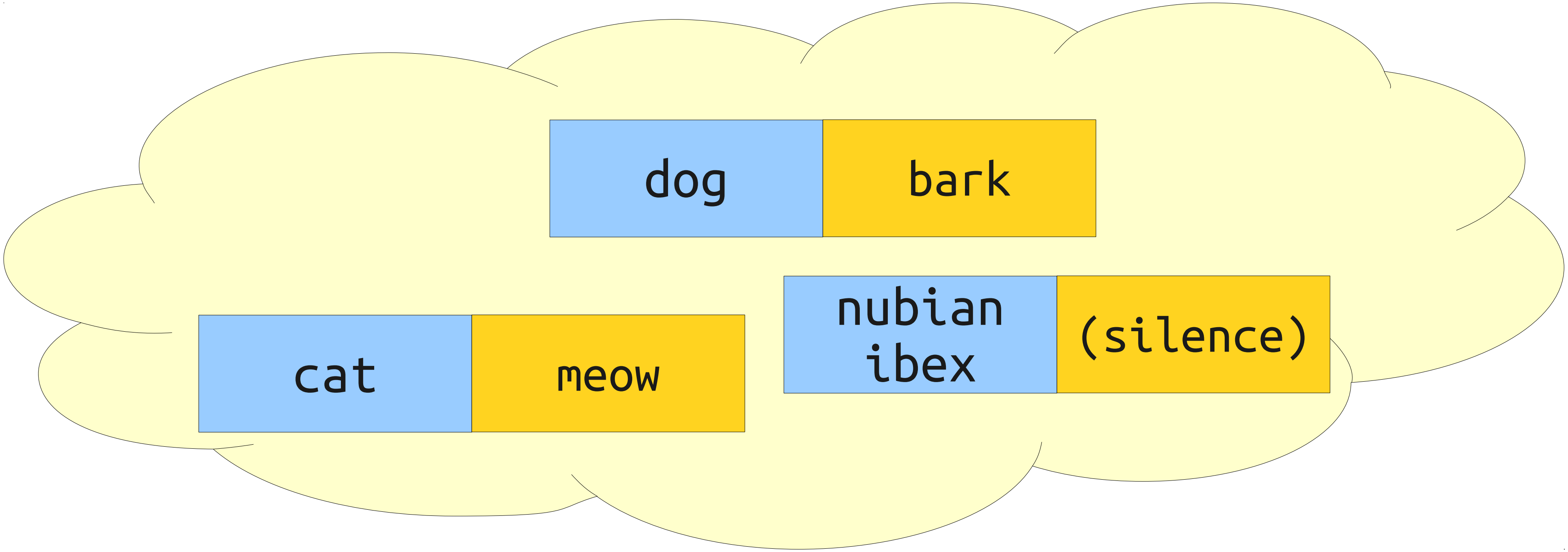
cat

meow

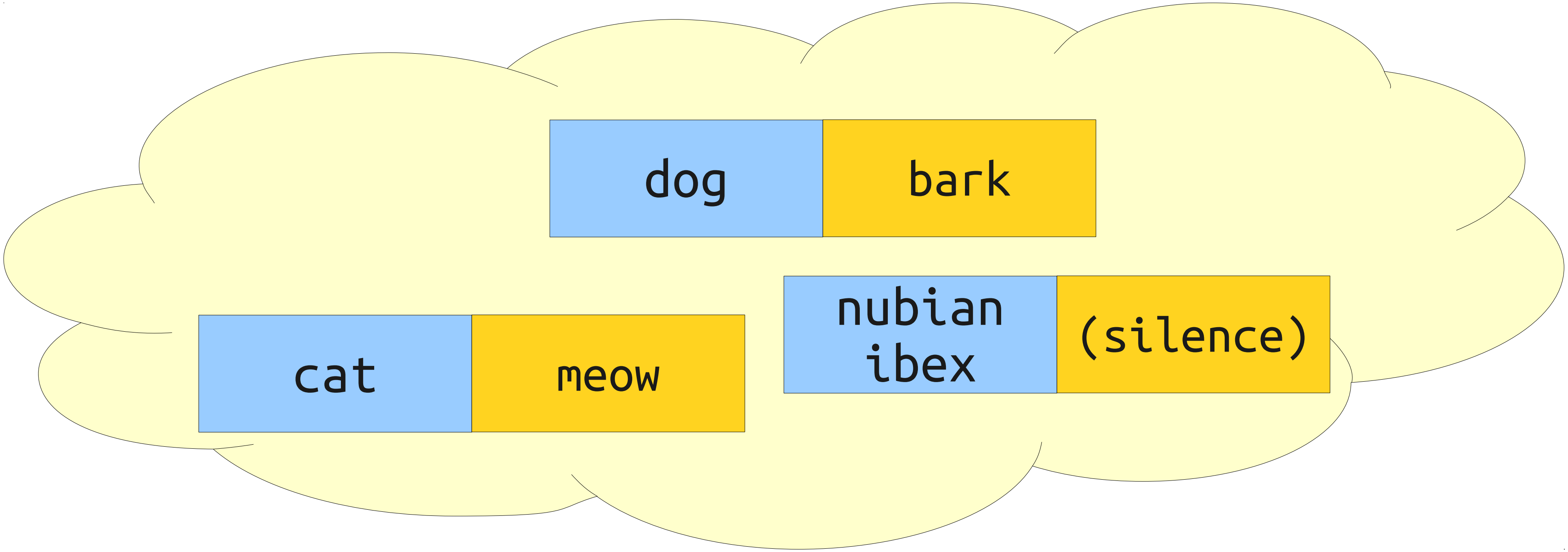
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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");
```



dog bark

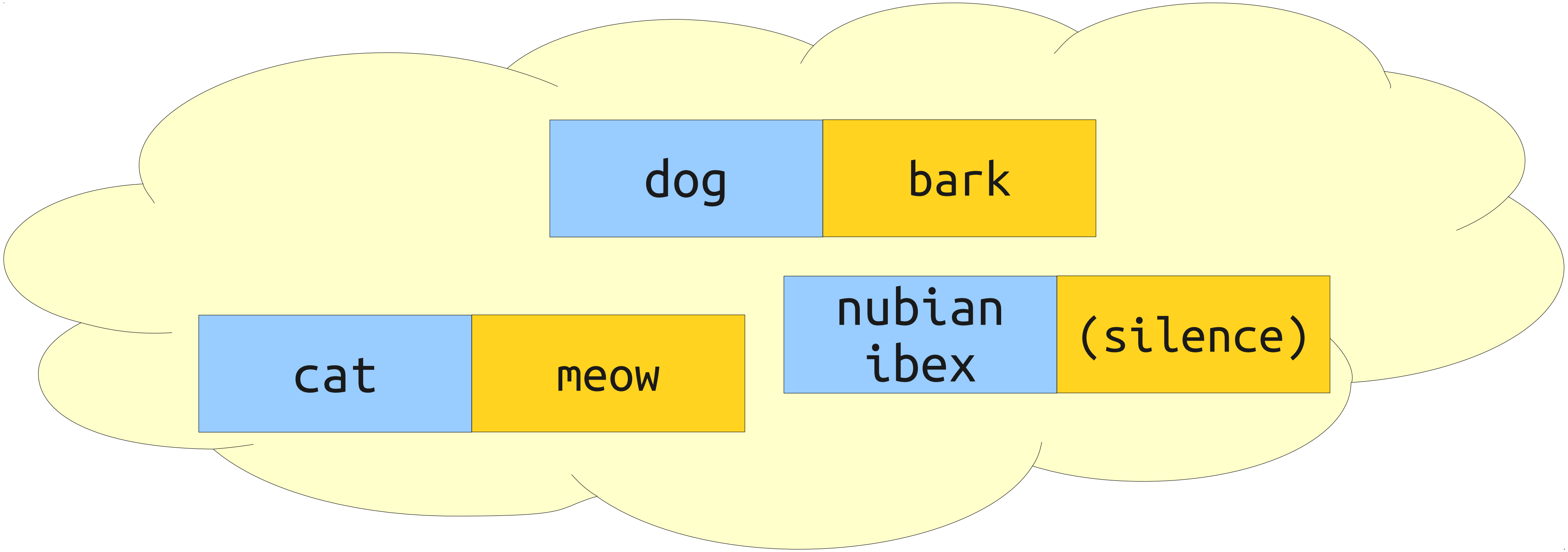
nubian ibex (silence)

cat meow

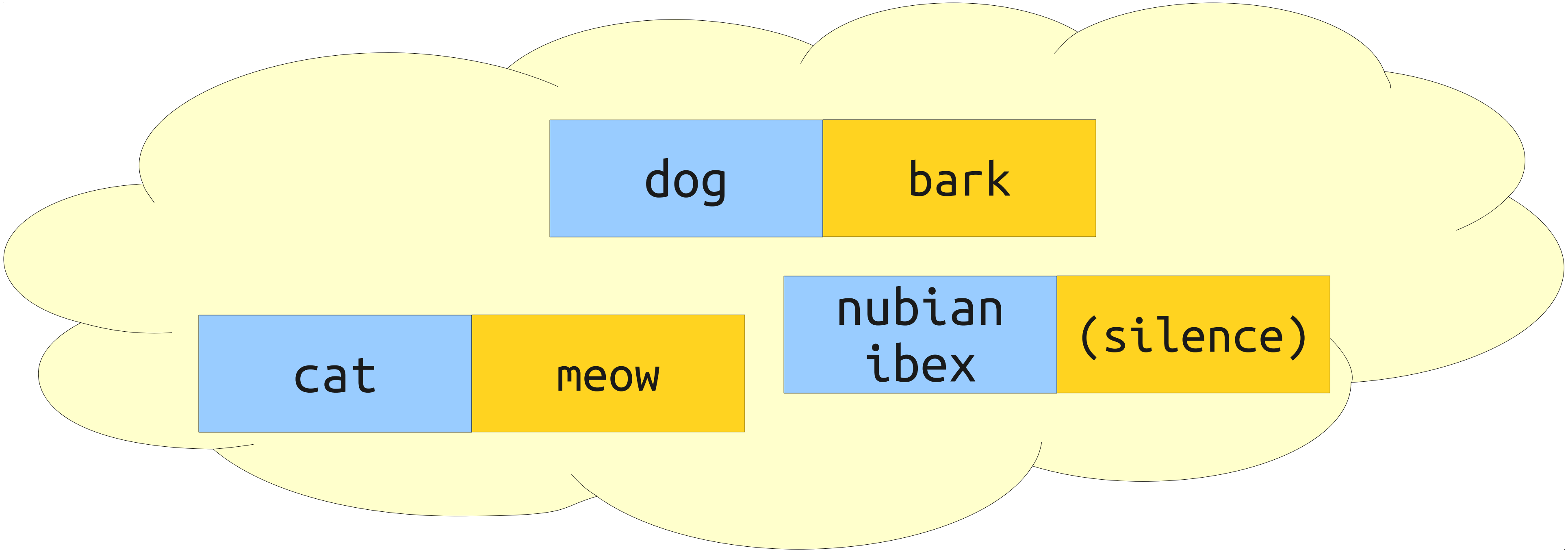
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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"
```

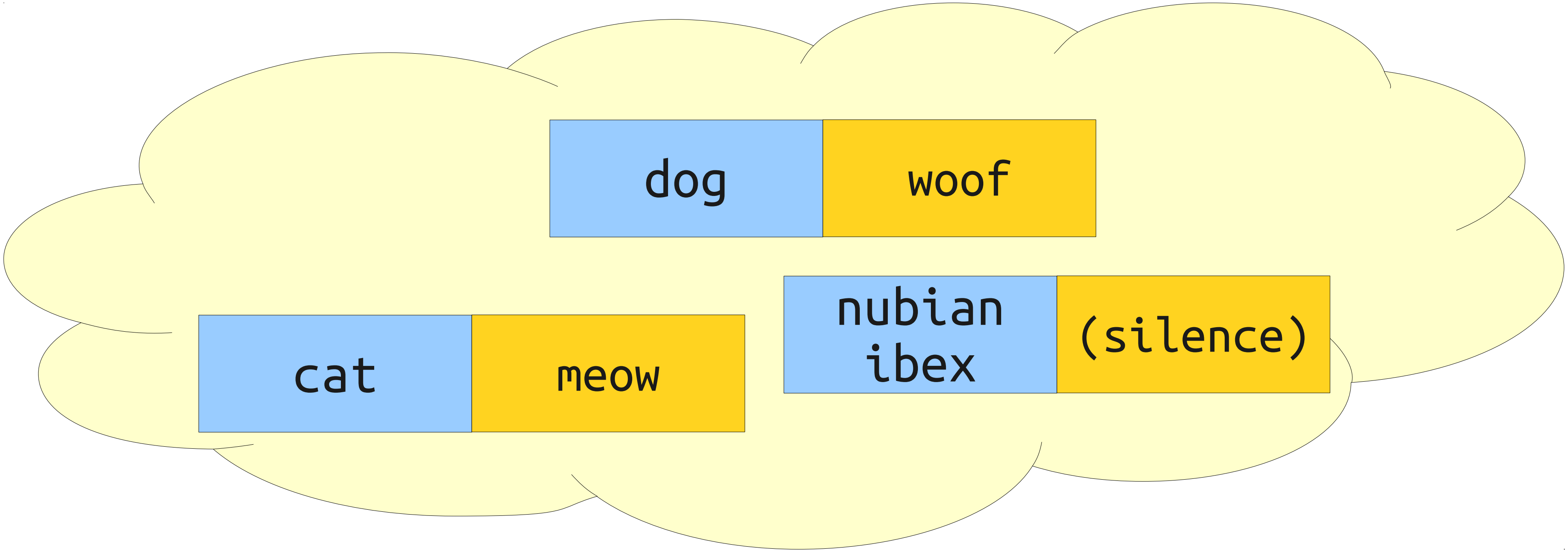


dog bark

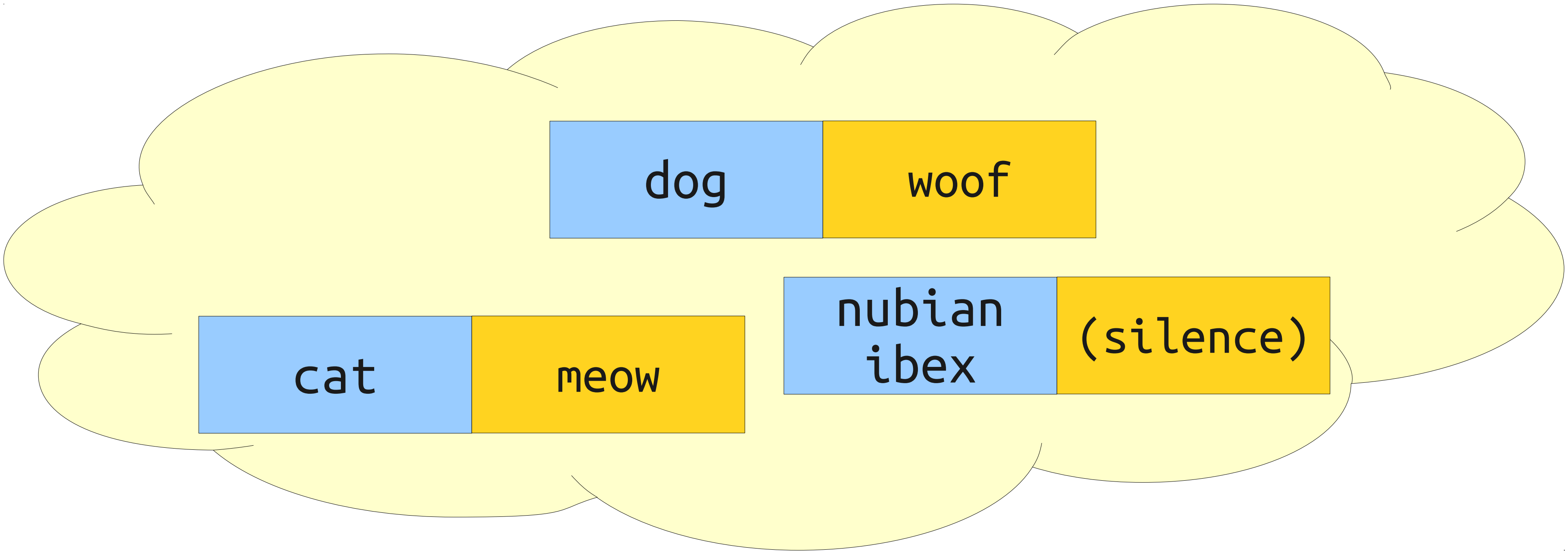
cat meow

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");          // 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");
```



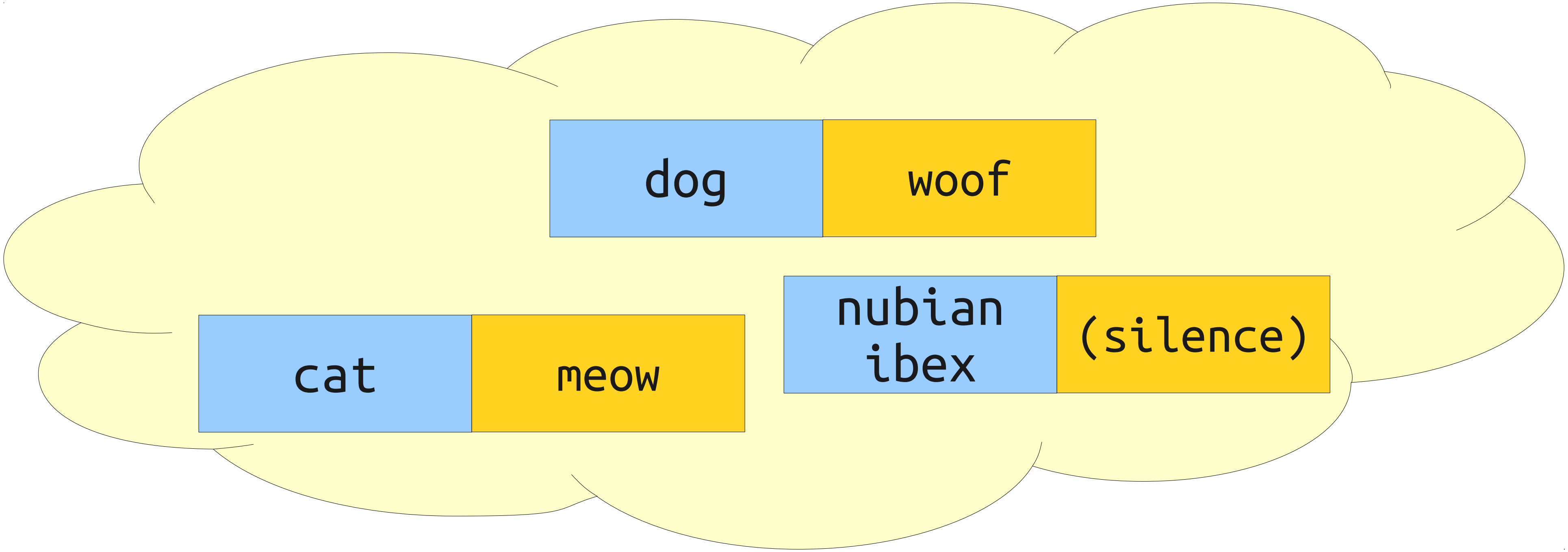
dog woof

cat meow

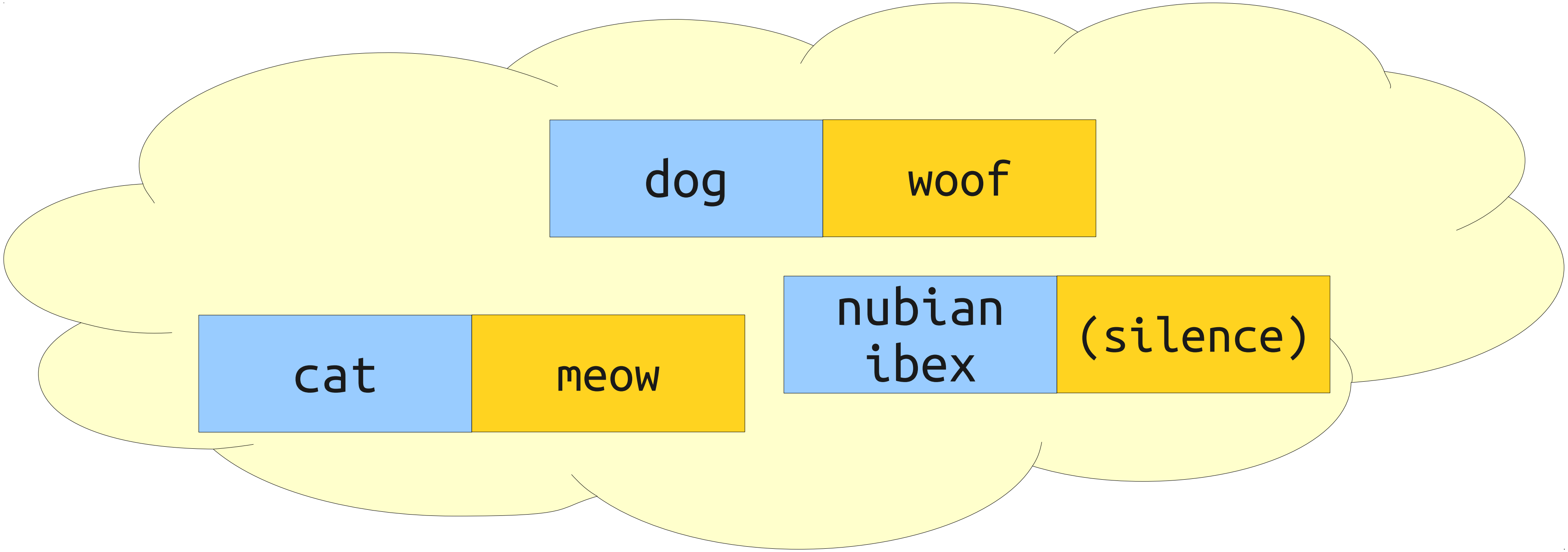
nubian ibex (silence)

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

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

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

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

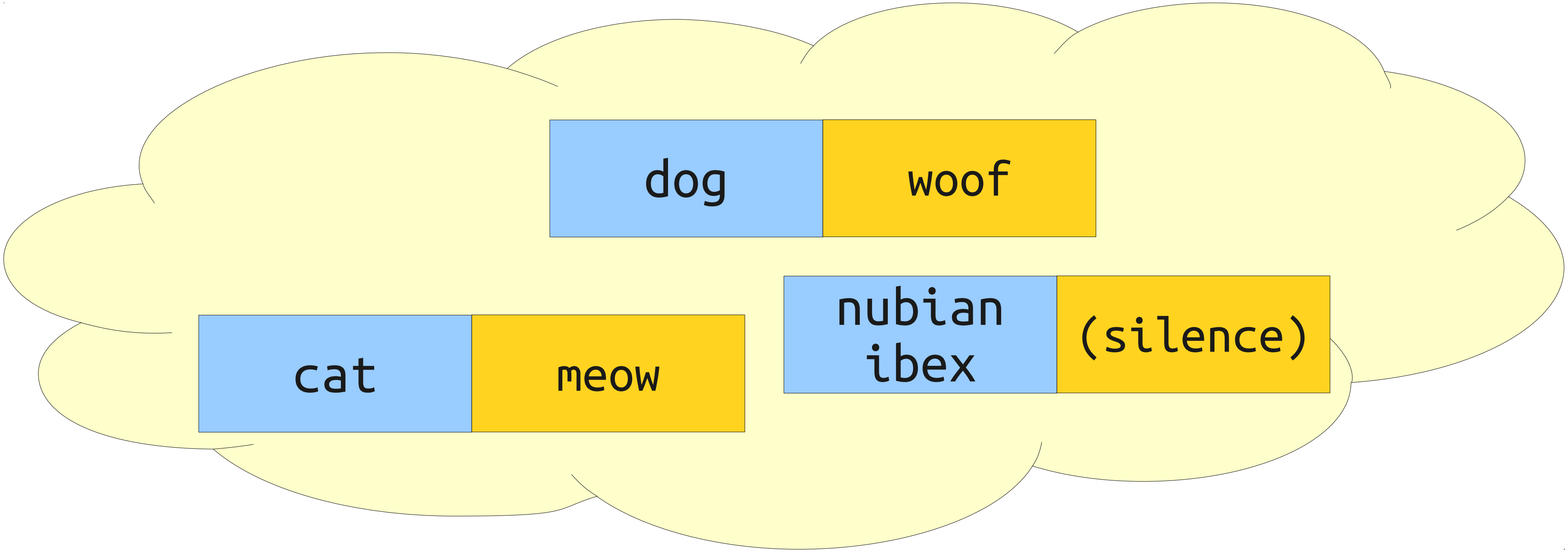


```
HashMap<String, String> animals =
     new HashMap<String, String>();

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");
```

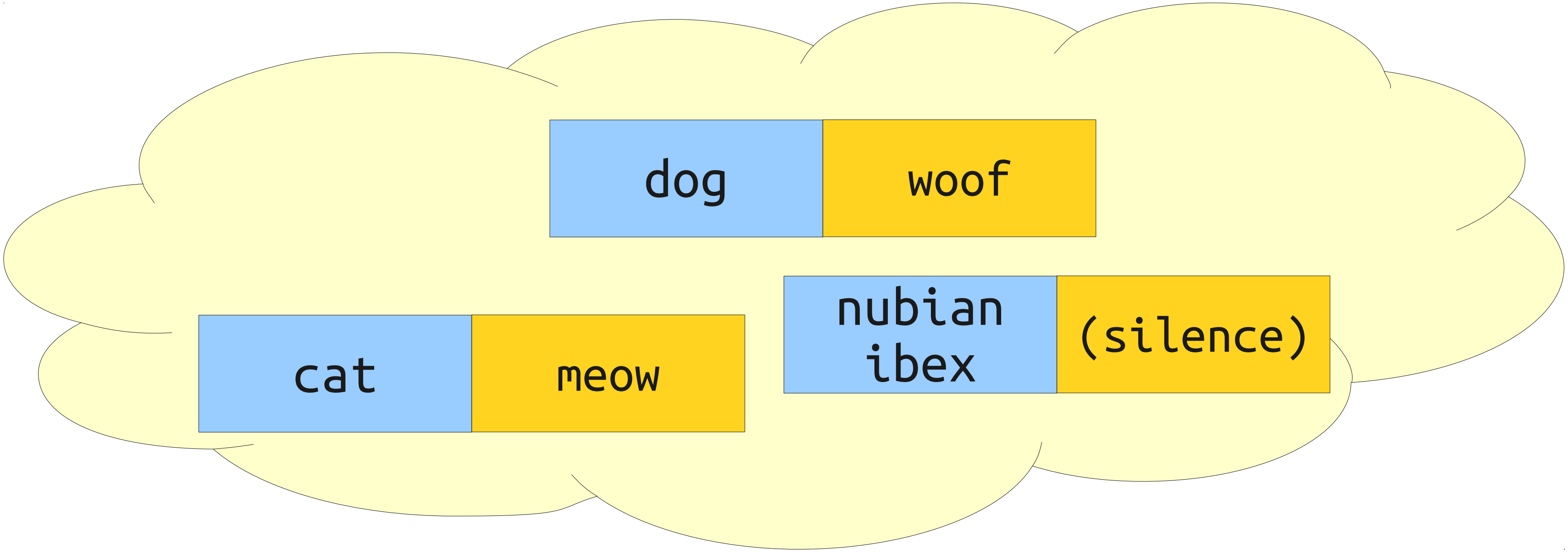


```
HashMap<String, String> animals =
     new HashMap<String, String>();

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

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

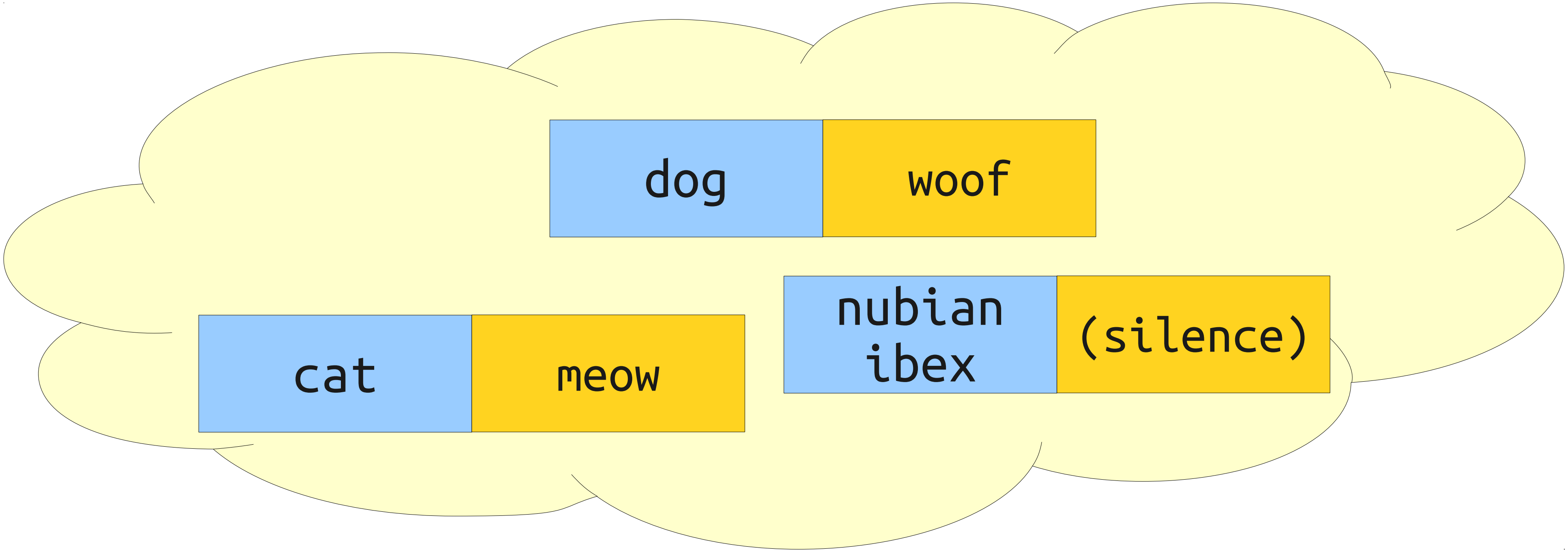
If you **get** a key that isn't in a map, the method returns **null**.



```
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
```



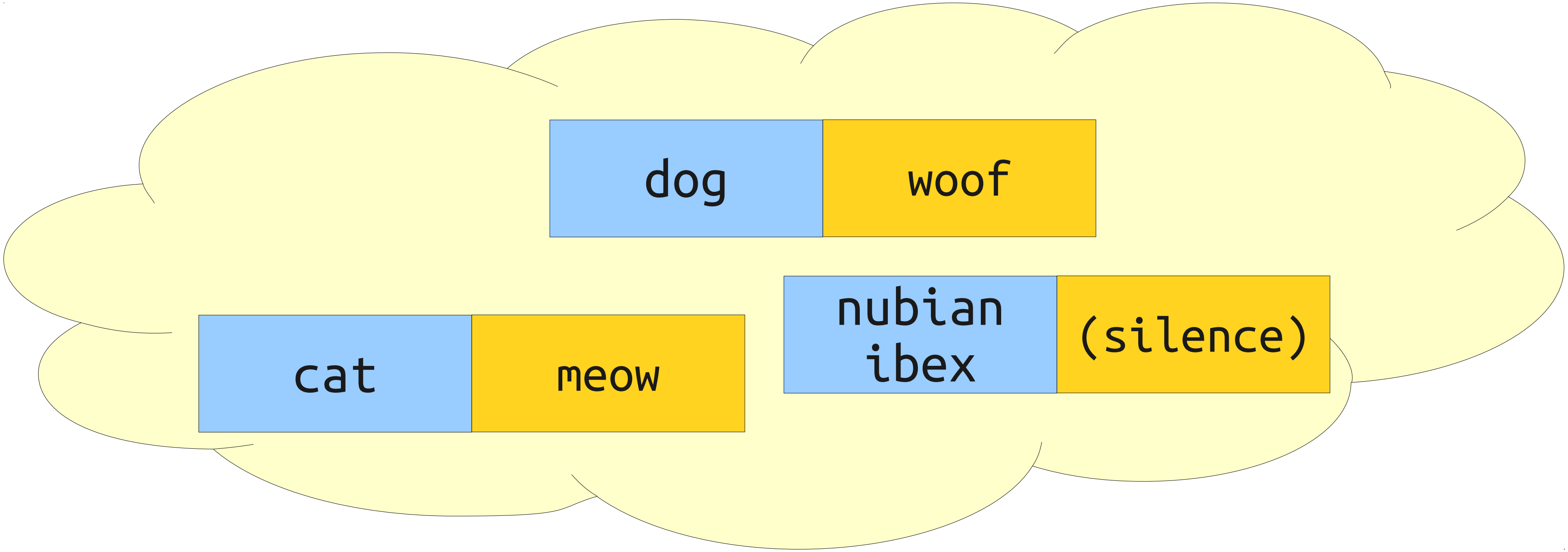
```
HashMap<String, String> animals =
     new HashMap<String, String>();

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!

Friday Four Square!

Today at 4:15PM at Gates CS

Midterms Graded

- Midterms graded, available for pickup in a filing cabinet in the Gates B Wing.
    - Right inside the entrance marked “Stanford Engineering Venture Fund Laboratories.”
- If you'd like to submit your exam for a regrade, attach a coversheet letting us know what to look at and hand your exam to Vikas or Keith.
    - Deadline: next Wednesday at 3:15PM.

Assignment 5

- Assignment 5 (Array Algorithms) is due one week from today.
- **Recommendation:** Complete Steganography by Monday and start working on Tone Matrix.

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

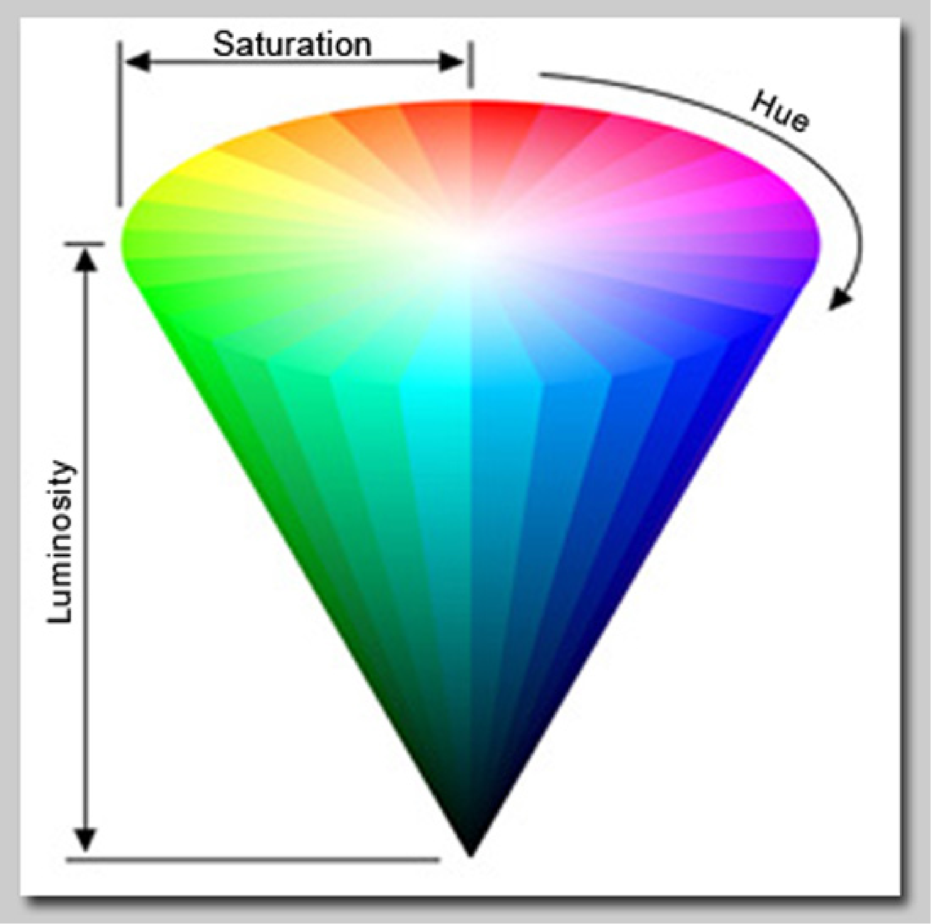
red

green

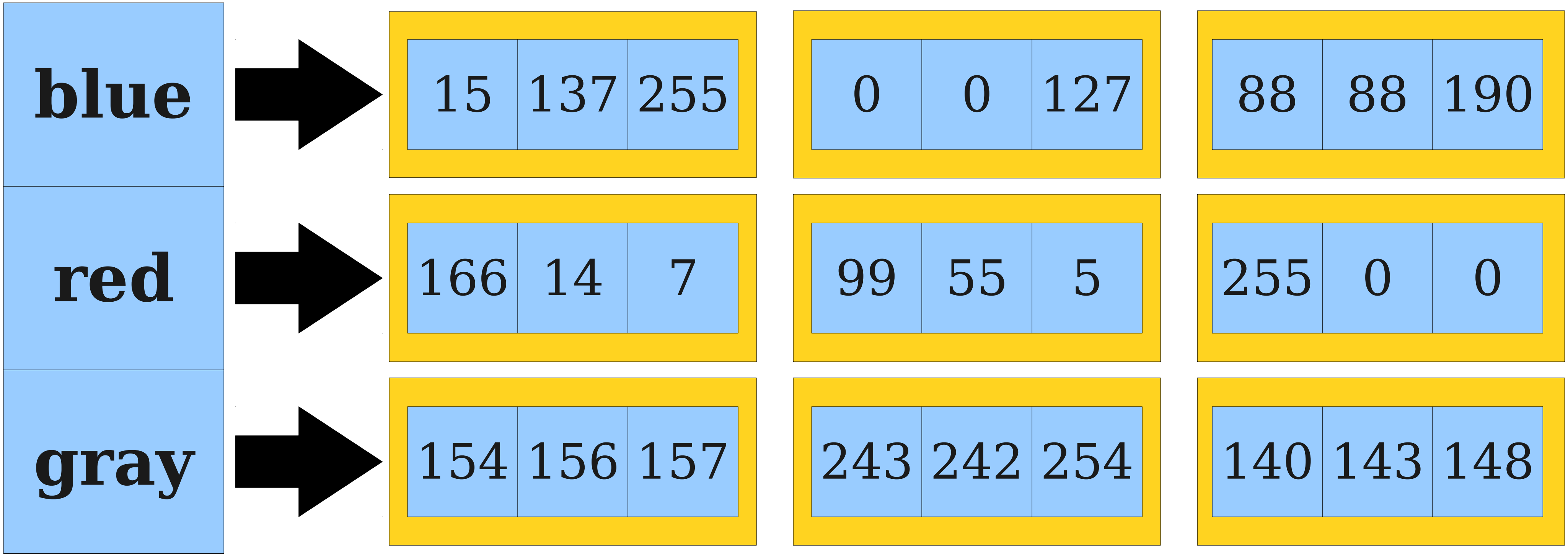
blue

Displaying Colors

- The HSB Color Format
  - Choose the **hue** (what color), **saturation** (how intense), and **brightness** (absolute brightness).
  - Each choice in the range from 0.0 to 1.0.



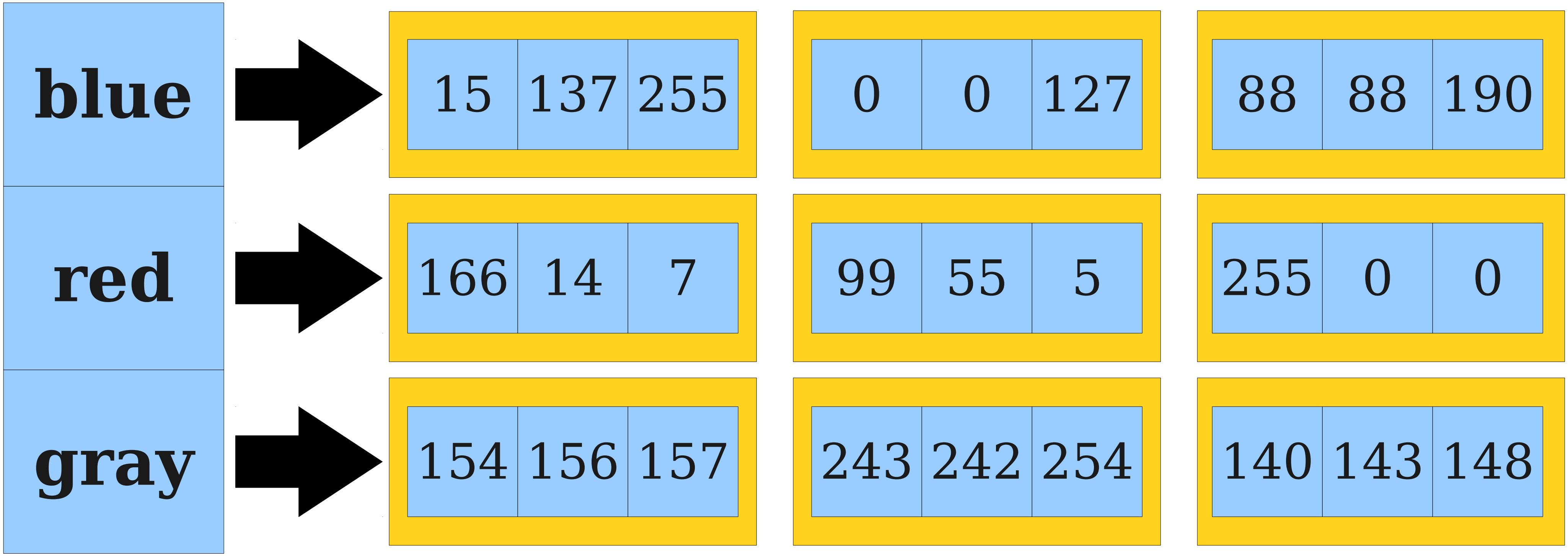
How to Structure the Data?



blue	15 137 255	0 0 127	88 88 190
red	166 14 7	99 55 5	255 0 0
gray	154 156 157	243 242 254	140 143 148

associate each color name
with a list of RGB triplets

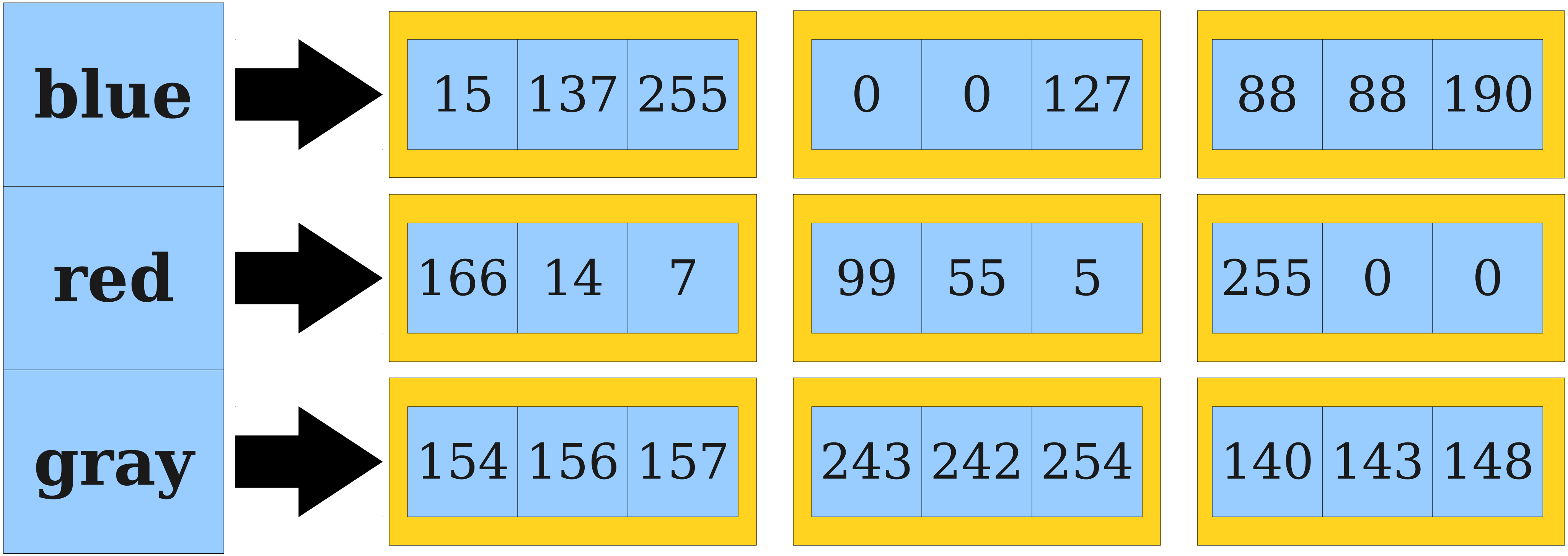
How to Structure the Data?



blue	15 137 255	0 0 127	88 88 190
red	166 14 7	99 55 5	255 0 0
gray	154 156 157	243 242 254	140 143 148

`HashMap<`***color name*** , ***list of RGB triplets***`>`

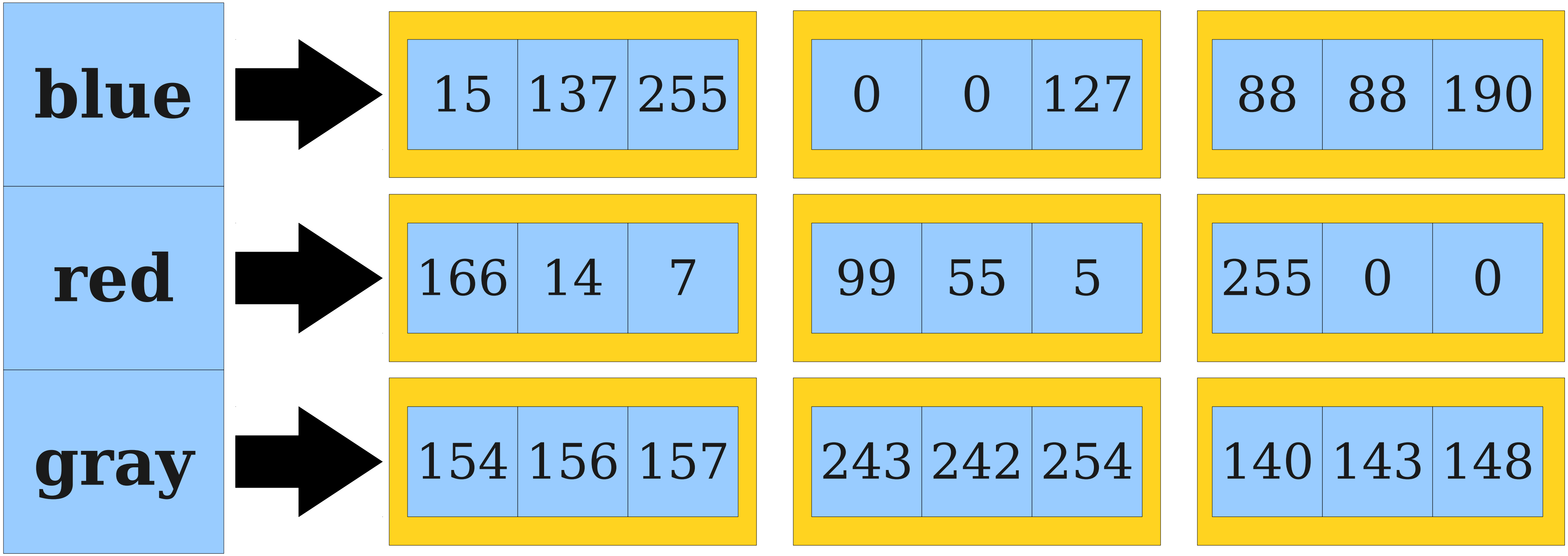
How to Structure the Data?



blue	15 137 255	0 0 127	88 88 190
red	166 14 7	99 55 5	255 0 0
gray	154 156 157	243 242 254	140 143 148

`HashMap<String,` ***list of RGB triplets*** `>`

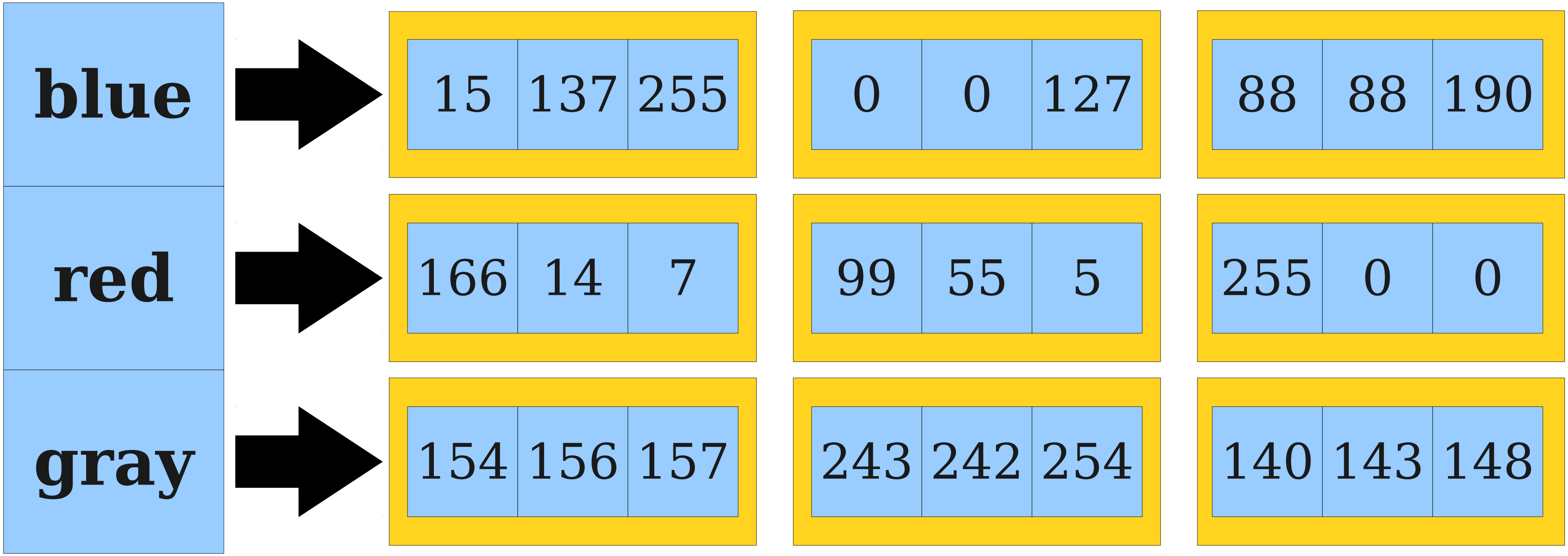
How to Structure the Data?



blue	15 137 255	0 0 127	88 88 190
red	166 14 7	99 55 5	255 0 0
gray	154 156 157	243 242 254	140 143 148

`HashMap<String, ArrayList<`*RGB triplet*`>>`

How to Structure the Data?



blue	15	137	255	0	0	127	88	88	190
red	166	14	7	99	55	5	255	0	0
gray	154	156	157	243	242	254	140	143	148

```
HashMap<String, ArrayList<int[]>>
```

For More Information

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