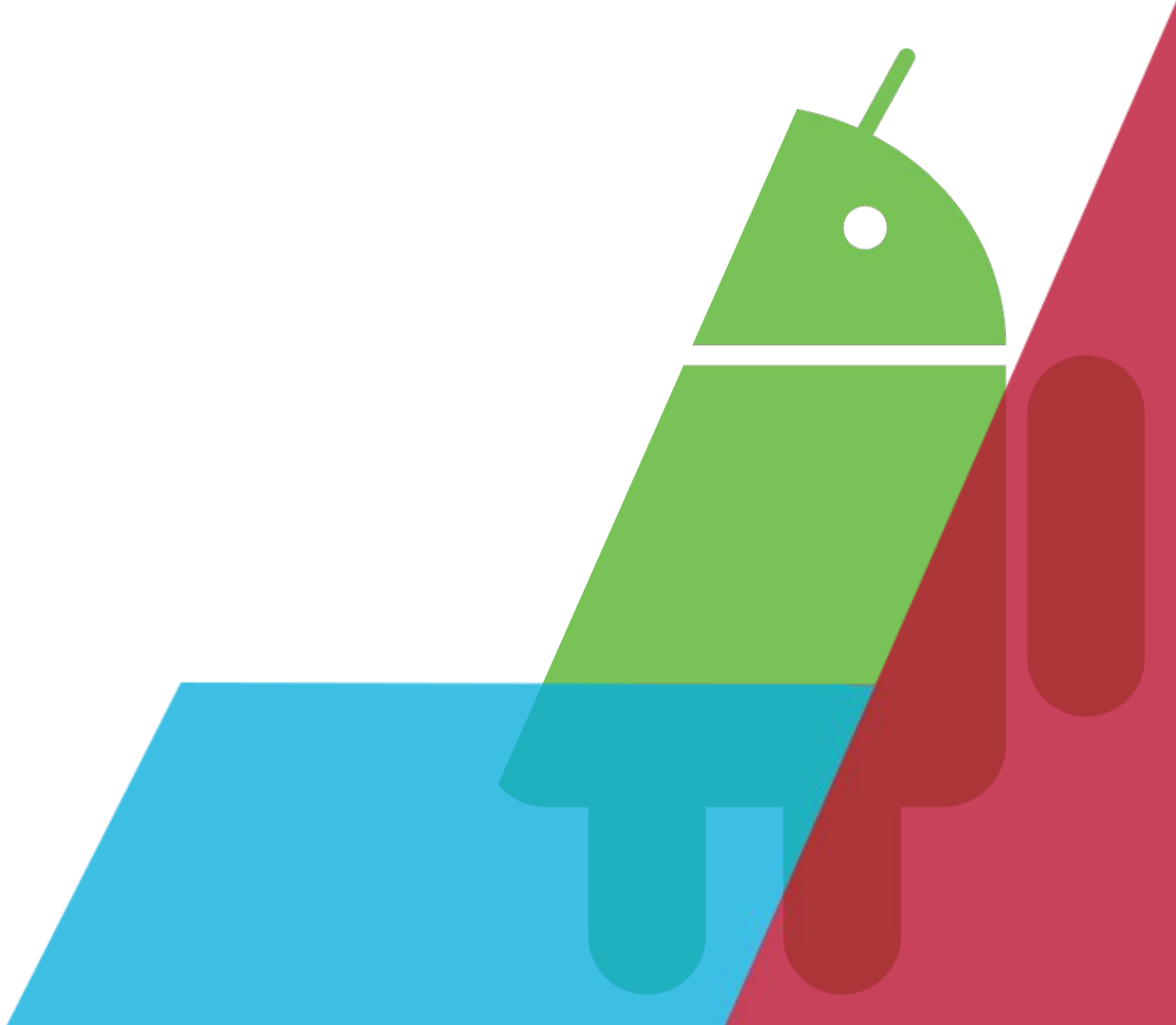
Collections

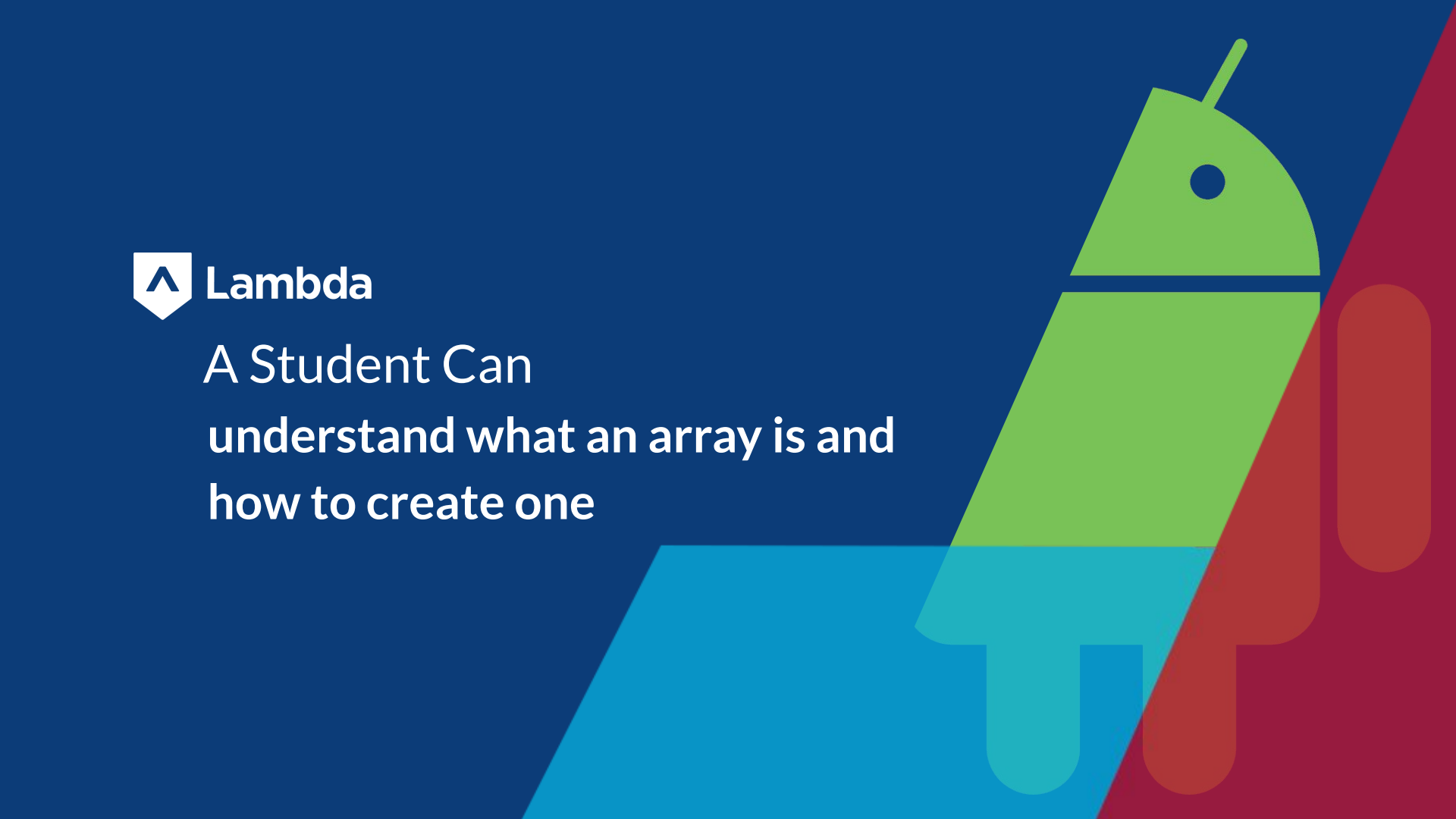


Upon completion of this module, a student will be able to

- understand what an array is and how to create one
- access and manipulate data in an array
- iterate through and array with a loop
- store data in a more advanced data structure
- use maps to store key-value paired data

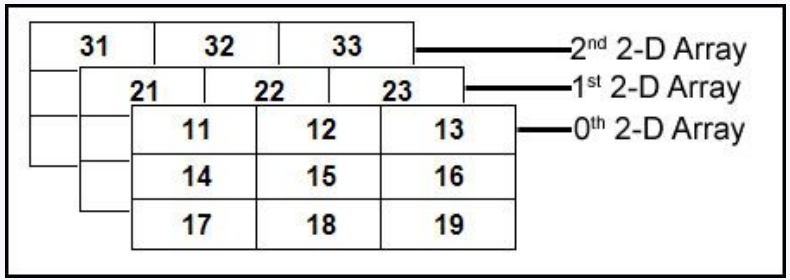
Project

- Task
  - For this project you'll build an app with a text field for the user to enter a word and loot for synonyms in a (small) database.
- Repo
  - https://github.com/LambdaSchool/Android_Collections
- Challenge
  - Try making it so that if a any word in the element array is equivalent to the value entered, that all its sibling synonyms are found and returned.


Lambda
A Student Can
understand what an array is and
how to create one

Arrays

- Data Structure
- Fixed Size
  - define size on creation
- Collection of Variables of the same type



Declare Arrays

- Declare array and size
- Fill with starter values

```
fun arrayDemo() {
    val size = 2
    val default = -1

    val array = Array(size) {
        default
    }
}
```

Challenge

- Declare an array of size 5 populated with 0

Solution

```
fun arrayChallenge() {
    val myArray = Array(5) { 0 }
}
```


Lambda
A Student Can
access and manipulate data in an array

Manipulating Array Data

- Arrays are indexed values
  - first value at 0
  - second at 1
  - etc.
- Accessed with the Index Operator

```
fun arrayDemo() {
    ...

    val array = Array(size) {
        default
    }

    array[2] = 45

    println(array[5])
}
```

Challenge

1. Take your array and add values to each of the elements
2. Print the contents of each of the elements

Solution

```
fun arrayAccessChallenge() {
    val myArray = Array(5) { 0 }

    myArray[0] = 12
    myArray[1] = 14
    myArray[2] = 3654
    myArray[3] = 567
    myArray[4] = 492

    println(myArray[0])
    println(myArray[1])
    println(myArray[2])
    println(myArray[3])
    println(myArray[4])
}
```

A Student Can

iterate through and array with a loop

Manipulating Array Data

- Arrays are indexed values
  - first value at 0
  - second at 1
  - etc.
- Accessed with the Index Operator

```
fun arrayDemo() {
    ...

    val array = Array(size) {
        default
    }

    array[2] = 45

    println(array[5])
}
```

Iterate Through an Array

- For loop
  - Get object through index
  - ..
    - Start and end (both inclusive)
  - until
    - Start and end (start inclusive)
- For each loop
  - Get object through `it` value

```
fun forLoopArrayDemo() {
    val myArray = Array(5) { it + 1}

    for(i in 0..myArray.size - 1) {
        println(myArray[i])
    }
    for(i in 0 until myArray.size) {
        println(myArray[i])
    }
    myArray.forEach {
        println(it)
    }
}
```

Challenge

- Use a loop to print the contents of your array

Solution

```
fun arrayAccessChallenge() {
    val myArray = Array(5) { 0 }

    myArray[0] = 12
    myArray[1] = 14
    myArray[2] = 3654
    myArray[3] = 567
    myArray[4] = 492

    myArray.forEach { println(it) }
}
```


Lambda
A Student Can
store data in an expandable data structure

Lists

- Expandable Array
  - grows when full
- Standard
  - Can’t change (Constant)
- Mutable
  - Can change
- Accessed like array

```
fun listDemo() {
    val readOnly = listOf(0, 1, 2, 3)
    println(readOnly[0])
//      readOnly[2] = 4 // can't set

    val mutable = mutableListOf(
        0, 1, 2, 3)
    println(mutable[0])
    mutable[3] = 1
    mutable.add(4)
}
```

Challenge

- Create a list, put all the numbers from 1 to 100 in it and print the list

Solution

```
fun listChallenge() {
    val myList = mutableListOf<String>()
    for(i in 0..100) {
        myList.add(i.toString(16))
    }

    myList.forEach { println(it) }
}
```


Lambda
A Student Can
use maps to store key-value
paired data

Map

- Key-Value
  - Index is key variable
- Declare type of key and value

```
fun mapDemo() {
    val stringMap = mutableMapOf<String,
Int>()

    stringMap["one"] = 1
    stringMap["two"] = 1

    val intMap = mutableMapOf<Int, String>()

    intMap[1] = "First"
    intMap[10] = "Tenth"
}
```

Challenge

1. Create a String, String map
2. Populate it with the keys and values shown here
    - Keys on left, Values on right


```
{
   "id": "299534",
   "title": "Avengers: Endgame",
   "poster_path": "/or06FN3.jpg",
   "original_language": "en",
   "release_date": "2019-04-24"
}
```

Solution

```
fun mapChallenge() {
    val movies = mutableMapOf<String, String>()

    movies["vote_average"] = "8.4"
    movies["title"] = "Avengers: Endgame"
    movies["poster_path"] =
"/or06FN3Dka5tukK1e9sl16pB3iy.jpg"
    movies["original_language"] = "en"
    movies["release_date"] = "2019-04-24"
}
```