



Strings

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

- understand the role and use of strings
- use string methods to manipulate String text
- understand the benefits and drawbacks of StringBuilder
- use interpolation to format strings
- understand the use of and reasons behind String Resource Files



Assignment

- Task
 - Write an app that can check if a string is a palindrome.
- Repo
 - https://github.com/LambdaSchool/Android_Strings
- Submission
 - Compress the project directory into a zip archive and then send it to your PM in a DM
- Challenge
 - Try making the app so that it ignores capitalization, punctuation and spaces when checking the string.





A Student Can

understand the role and use of strings

String

- Array of characters
- Wrapped using the String object
- All “Words” are treated as String objects

```
char[] array = new char[] { 'a', 'b', 'c' };  
String string = new String(array);  
  
String string = "abc";
```



A Student Can

use string methods to manipulate String
text

String Manipulation

- Multiple strings concatenated with '+'
- Compare strings with equals(string)
- Access Character with charAt(index)
- Replace Characters with another character with replace(target, replacement)
- Get substring with substring(start Index, end index)

```
String helloWorld = "Hello " + "World";

String hello = "hello";
hello.equals("hello");

char l = hello.charAt(2);
String help = hello.substring(0, 3).replaceLast('l', 'p');
```



A Student Can

understand the benefits and drawbacks of
StringBuilder

StringBuilder

```
StringBuilder builder = new StringBuilder();  
builder.append("Hello");  
builder.append("World");  
builder.insert(5, " ");  
System.out.println(builder.toString());
```

- Higher performance concatenation
- `target.append(String)`
 - concatenates string to end of target
- `target.insert(index, String)`
 - overwrites target starting at index and overwriting for length of String



A Student Can

use interpolation to format strings

String Interpolation

- %[argument_index\$][flags][width][.precision]dataType
- flags
 - ' ' - leading spaces
 - '0' - zero-padded
 - ',' - grouping separators
 - '-' - left justified
- data type (conversion)
 - 'd' - decimal integer
 - 'f' - decimal number
 - 's' - string

```
String.format("%s %s", "Hello", "World");  
// Hello World  
  
String.format("$ %0,4.2f", 2500f);  
// $ 2,500.00  
  
System.out.format("Agent %03d", 7);  
// Agent 007
```



A Student Can

understand the use of and reasons behind
String Resource Files

String Resource File

```
1 <resources>
2     <string name="app_name">Rainbow</string>
3 </resources>
```

- Array of characters
- Wrapped using the String object
- All “Words” are treated as String objects