Learning Objectives: String Iteration

- **Define string iteration**
- **Identify two ways to iterate over a string**
- **Explain the inner workings of string iteration**

Iteration: For Loop

Iterating Over Strings

You have seen how you can make a copy of individual characters in a string with their index. Iterating over a string allows you to deal with each character of a string individually. You start with the character at index `0` and move through the end of the string.

```
String myString = "Hello world";

for (int i = 0; i < myString.length(); i++) {
  System.out.print(myString.charAt(i));
}
```

challenge

What happens if you:

- Change the value of `myString` to `"10, 11, 12, 13, 14"`?
- Change the value of `myString` to `"\u25A3\u25A8\u25D3\u25CC\u25A2"`?
- Change the `print` statement to `System.out.println(myString.charAt(i))`?
- Change the `print` statement to `System.out.print(myString)`?

Behind the Scenes

Click on the code visualizer link below to open up the visualizer. Then, step through the code by continuously clicking the `Forward` button. Notice how the iterating variable `i` goes through each index from beginning to end.

Code Visualizer

Iteration: While Loop

While Loop

String iteration is most often done with a `for` loop. However, a `while` can be used as well.

```
String myString = "Calvin and Hobbes";
int i = 0;

while (i < myString.length()) {
  System.out.print(myString.charAt(i));
  i++;
}
```

Code Visualizer

challenge

What happens if you:

- Change the loop to `while (i <= myString.length())`?
- Copy the original code but change the `print` statement to `System.out.print(i)`?
- Copy the original code but remove `i++`?

Comparing While & For Loops

```
String myString = "Java";

for (int i = 0; i < myString.length(); i++) {
  System.out.print(myString.charAt(i));
}
```

```java
String myString = "Java";
int i = 0;

while (i < myString.length()) {
  System.out.print(myString.charAt(i));
  i++;
}
```

Above are two ways of iterating through a string. The first way uses the `for` loop and the second uses a `while` loop. Both produces the same result. However, the `for` loop is usually preferred because it requires less code to accomplish the same task.