break and continue

In this lesson we will learn how to have more control over loops using the break and continue statements.

We'll cover the following

The `break` Statement

`break` is used for prematurely stopping a loop. When Dart finds a `break` statement, it breaks from the loop regardless of whether the iterations have been completed or not.

It's mostly used with a conditional statement. Based on the condition, the loop will either need to exit or not.

Let's look at an example similar to the one we looked at with for-loops, where we have a list of integers and we only want to print the even integers from the list. In this example, however, we only want to know what the first occurrence of an even integer is.

```
main() {
  var intList = [7,3,9,6,2,5,4];

  for(var i in intList){
    if(i % 2 == 0){
      print(i);
      break;
    }
  }
}
```

This is very similar to the example we looked at in the lesson about `for` loops. In that example, we wanted to print every even integer in the list. However, in this example, we have inserted a `break` statement in our conditional statement. When the first even integer is found, the condition of the `if` statement would become `true`, resulting in the execution of **line 6** and **line** 7. On **line** 7 we are telling the loop to break, so only the number **6** is printed and not **2** and **4**.

The `continue` Statement

`continue` is used to skip the running iteration and move on to the next one, regardless of if there are still lines of code to be executed.

Imagine you're responsible for hiring a new employee for your company that has 5 years of work experience. You go over the list of candidates. Candidates with less than 5 years of experience are skipped, while candidates with more than 5 years of experience are called for an interview.

We will store the experience of our candidates in a list. The first candidate's experience will be at index **0**, the second candidate's experience will be at index **1**, and so on.

Let's write some Dart code that will be able to tell us which candidate should be called for an interview.

```
main() {
  var experience = [5,1,9,7,2,4];

  for(var i = 0; i < experience.length; i++){
    var candidateExperience = experience[i];
    if(candidateExperience < 5){
      continue;
    }
    print("Call candidate $i for an interiew.");
  }
}
```

If, on an iteration, the value of `candidateExperience` is less than **5**, **line** 7 of the code snippet above is executed which skips the current iteration and moves back to the top of the loop. Hence, the print statement will never be executed for those iterations.

Let's move on to the `switch` and `case` statements in the next lesson.