Which of the following would you use to find the number of elements in an array named **arr**?

- arr[k]
- arr.length()
- arr.charAt(k)
- arr.length

Correct

Continue

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:02
Hello. You're going to learn the basics of a powerful programming construct, the array. Nearly every programming language in widespread use uses a similar structure to be able to represent many items with one variable. When you wrote code to help a genome scientist solve a problem, you could count the number of occurrences of each c, g, t and a in a strand of DNA. This is a real problem, that helps find protein coating regions rich in c,g content.

0:34
As a programmer learning about encryption and decryption, you'll need to count the number of recurrences of a, b, c and every letter of the alphabet through x, y and z. 26 counters. To be able to break a Caesar cipher or decrypt a message encrypted with a Caesar cipher, although you could do this with 26 variables, learn a new concept to help with decoding. We'll learn about arrays, which are homogeneous collections of values. Here you see post office mailboxes, they each look the same, but each is numbered differently. You can access box number 344 or box 345. And you can either store letters and packages in a box, or take them out of the box.

1:17
Arrays are similar concepts in programming. As you'll see one array could represent 26, or 1,026 counters, if your alphabet was huge, or if you were solving a different problem.

1:30
Let's look at code to help a genomic scientist. We'll look at code that counts occurrences of c, g, t, and a. To understand and motivate the problem of counting 26 letters.

1:42
As you can see, we've created four counters. Initialized each one to zero, and then incremented the appropriate counter, as we process every nucleotide in a digital strand of DNA.

1:56
This solution works, but it's very hard to scale having 26 counters, which we would need to count every letter A through Z as part of decrypting a message encrypted using a Caesar Cipher.

2:08
This isn't conceptually hard. We could use CA, CB, CC, and so on up through variables CY and CZ to have 26 variables. And we're gonna have 26, if statements to increment the appropriate variable. But writing the code and changing what we do with 26 values is time consuming, and difficult to change if we want to print something different, for example, when looking at the output.

2:34
We'll use an array, an indexed collection, to use one variable in place of 26. We'll break the Caesar cipher by counting occurrences of each character in an encrypted message. A message in English would typically have the letter e as the most frequently occurring character. So, once we found how many times each character occurs in an encrypted message it's likely that this character is E, and we can determine the shift using used them encrypting the message making the description process easy. In general, counting and collecting values in an important tool in running programs, so learning about arrays and the context of breaking a Caesar cipher will help in solving many other problems. You've seen the class storage resource, which was helpful in storing string values. That class was useful but it's use was limited. We'll expand on the idea of arrays, and the storage resource class later. For now, we need an indexed collection. Like the collection of mailboxes we saw, in which a number is used to access a specific location. This is the same concept you've seen with strings, in which an index is used to access a particular character in a string using the .charAt or .substring methods. Arrays can store any type of value, not just the type char that's used in strings.

3:55
We'll look at the concepts and code in using arrays. And see how they're similar to what you've seen before with strings.

4:03
You define an array similarly to defining a string.

4:08
When defining a string variable as you see here, you indicate the characters that are in the string and assign these characters to a variable. With an array, you must specify how many storage locations there are. And use the square brackets with the variable name to indicate the variable is an array. This code allocates 256 memory locations, each one holding an int with a value zero, which is the default value of integers in an array.

4:37
The concept of indexing is used to access elements of a string, and an array. With a string, the .charAt method is used with an index to access a specific character. The first character is zero, because we use zero based indexing. With arrays, the bracket operator acts as a specific element, again with zero based indexing. When writing code, you often need to know the number of characters in a string, or the number of elements in an array.

5:08
With a string, you use the Dot Length method to determine how many characters there are in the string. With an array, you use the values stored in Dot Length to access the number of storage locations allocated for the array. Note that that length is not a method for arrays. This is sometimes a source of confusion when writing code.

5:30
We'll look at the code to count the number of occurrences of every character a through z. You'll see this code and the concepts in it will help you solve many problems when programming. In this code, you'll see a variable named counters that will represent 26 different counters. The code will store the number of occurrences of a in counters subzero. We use sub as a shorthand for subscript, a term coming from mathematics.

5:58
We'll see that counter sub k Is the number of occurrences of the kth letter. By this we mean if the number of b's is at counter sub one and the number of z's is a counters sub 25. As you look at the code, you will see there at three parts. Just as there were with the code that counted the number of occurrences of c,g,a and t. In the string representing DNA.

6:23
In that code, four counters were defined and initialized to the zero. Here, there are 26 counters defined, and initialized to the zero. The array reference are variable counters takes the place of 26 different variables.

6:40
In the DNA counting code, we used a sequence of four, if statements to determine which counter to increment.

6:47
Here, we used the location of a character in the string alpha, as the index of the appropriate counter to increment. Note that A has index zero.

6:59
We even use both upper and lowercase As, using the Character.toLowerCase method, so that the index value returned by alpha.indexOf helps us increment the appropriate storage location in the counters array.

7:14
Finally, to print each result, we use the loop index K to both access the kth value stored in alpha, and the kth value stored in the counters array.

7:27
You'll gain experience in solving problems with array. Here's quick summary of what we've just introduced. Arrays are indexed collections of values. When defining an array, you will typically provide an integer value indicating how many elements can be stored in the array. It's possible to define a variable like x, as you see here, with no storage allocated for it, simply to define the type of the variable. This could be useful, for example, as a parameter in a method.

7:56
If you define an array by calling new you must provide an integer value for the number of array elements. In an int array, all locations will be initialized to zero. For a string array, all array locations are initialized to null. That is the value we have seen before that indicated there is no object being referenced.

8:16
Array locations are red and written using indexes. You can store a value in an array, as shown here. With S sub 3 getting the string hello. This is writing a value into an array location. You can also access or read an array location, as shown here. Well, on the right hand side of the assignment statement we see x sub 3 is used to assign or write to a value on the left hand side of your assignment statement x sub 2. Once the storage is allocated for an array, the array size does not change. This may be why.length is not a method but a value.

8:54
When an array is passed to a method, the contents of the locations referenced by the array can change. This is subtle, and you'll see examples of it when we use arrays to solve problems. Have fun coding.

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?