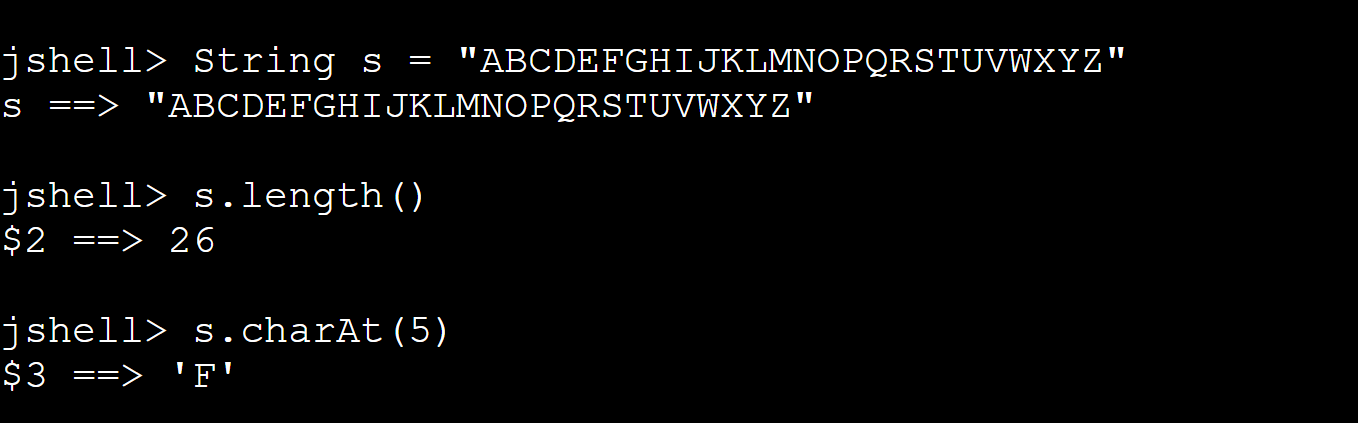
**Genes workshop**

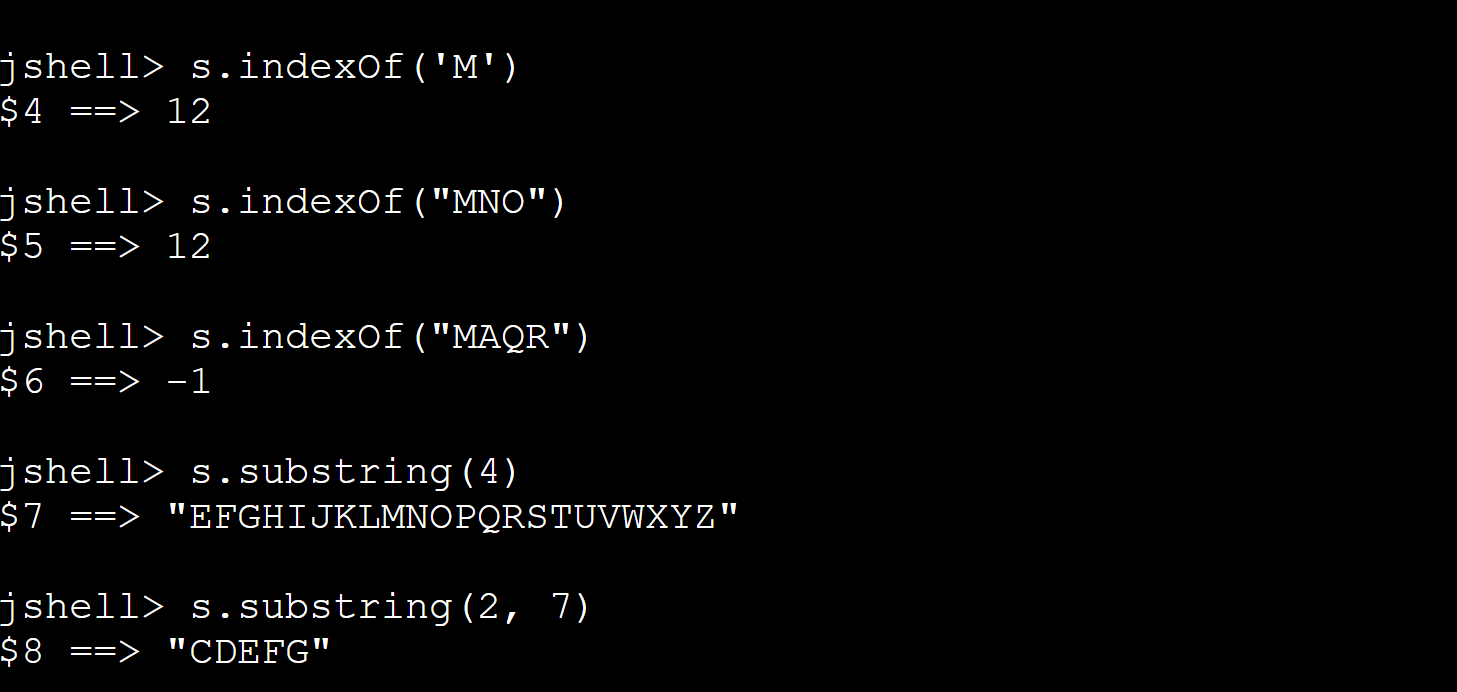
The aim of this workshop is to help students feel more comfortable with using and writing functions. We will use JShell, which makes writing functions more straightforward. It’s important to remind students of the differences though (public static).

**A reminder about Strings**

All the functions we’ll use today will handle Strings and use their associated functions. For a String ‘s’, the functions we will care about are:

* s.length()
* s.charAt(int index)
* s.indexOf(char c)
* s.indexOf(String sub)
* s.substring(int start)
* s.substring(int start, int end)

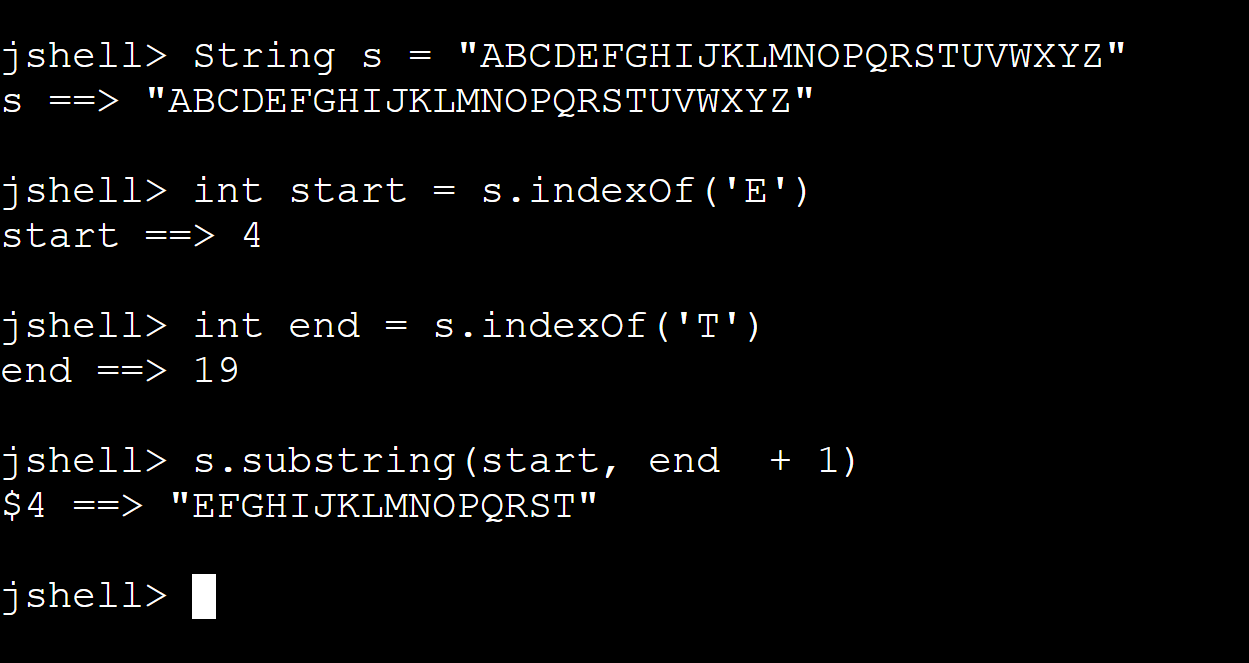
Demonstrate the use of these functions with a test String. Below is one possible example.



Strings warm-up question: Ask students to solve the following question. Start with the String

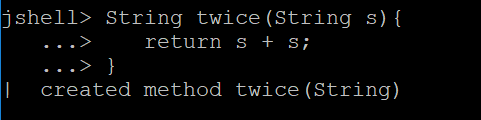
s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ". From this String, construct a new String which contains all letters between ‘E’ and ‘T’ (including both). Students may only use String functions.

Solution:



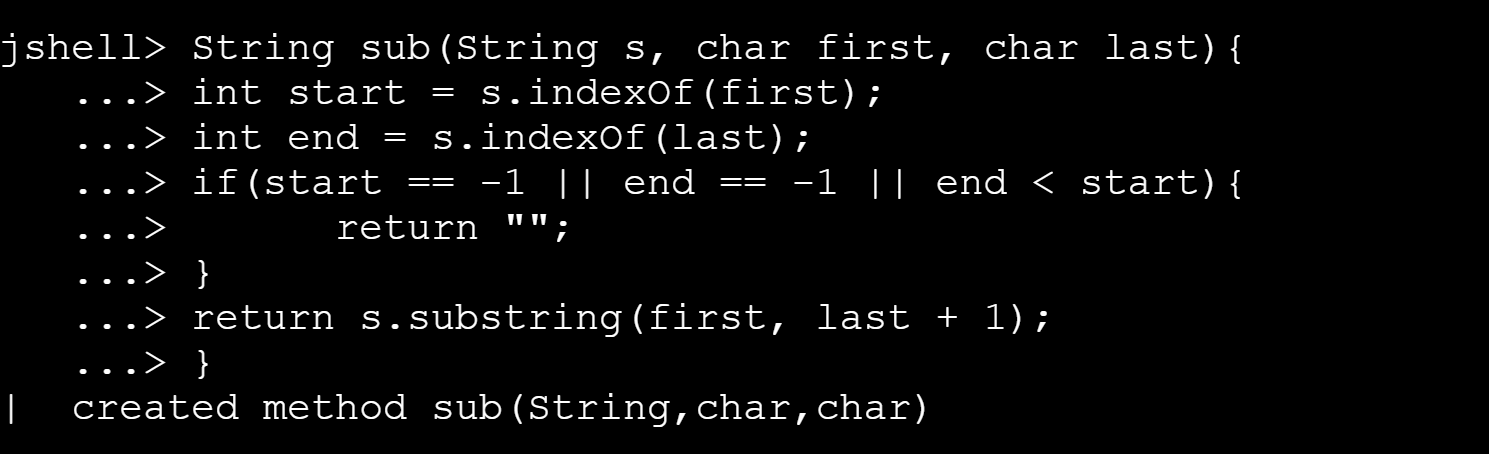
**Functions - Reminder**

Let’s recall the function from a previous workshop:



Remind students about writing functions, and the difference between functions in JShell and in Java.

Functions warm-up question: Ask students to write a function which receives one String and two characters. The functions should return the substring of the input starting from the first character and terminating at the second character (including both). If one of the characters does not exist or if the last character appears before the first character return the empty String.



**Finding Genes**

For our purposes a DNA strand is considered to be a String in which all characters are one of the four possibilities: ‘A’, ‘C’, ‘G’, ‘T’ (in reality every such letter represents one of four possible biological compounds).

Every three letters compose a single word, or codon (in reality, it codes for amino acids).

Thus, a valid DNA strand should have length divisible by three. Discuss with students how to check that.

A DNA strand can contain many genes. A gene is a sequence of codons inside the strand which is enclosed between two special codos, a start codon and an end codon.

Today, we will only consider one possible start and end codons. The start codon is given by “ATG” and the stop codon is given by “TAG”.

For example if the DNA is "GGTCCGATGCCTGGGCTCTAGGAA", it contains a single gene "ATGCCTGGGCTCTAG".

Another example if given with the DNA "ATGTTCAAATAGCCCTTA" and the gene "ATGTTCAAATAG".

The main task for today is to write a function, which takes a DNA strand as input and returns a single gene which is contained inside it.

Here is the description:

/\*\* Finds the first gene in a given DNA strand, if there is one. Returns the

\* first string of codons (triplets) that starts with "ATG" and ends with "TAG",

\* inclusive. The length of this string must be a multiple of 3.

\* If no such gene is found returns "".

\*/

Before letting the students solve the question, discuss briefly the possible challenges.

* What happens if “TAG” appears before “ATG”.
* What happens if the String looks like this “ATGGTAG”, which is not a valid gene?

Here is the solution:

public String findGene(String dna) {

int start = dna.indexOf("ATG");

if (start == -1) {

return "";

}

int stop = dna.indexOf("TAG", start + 3);

if ((stop - start) % 3 == 0) {

return dna.substring(start, stop + 3);

}

else {

return "";

}

}

**Possible extensions**

Ask students the following questions:

* What can we change if we want to allow for the input String to be with lower case letters as well?
* Suppose we want to find two genes in the DNA strand, instead of one. How can we do it?
* Suppose we want to find ALL genes in the DNA strand, can we even do it, by just using String functions?