Lesson 3

Strings and Index

Today's Lesson Plan

1. What is a String?
2. Concatenation of Strings
3. Concept of an Index
4. Types of Strings and Methods
    a. .length()
    b. .substring(start)
    c. .substring(start,end)
    d. .indexOf()
    e. .equals()
5. System.out.print

What is a String?

- A sequence of characters in Java
- Variable type (like int, boolean, double, etc.)
- NOT a primitive data type
- The characters are always surrounded by quotations (“”)

Examples:

```
“Hello World!”
“I like to eat ice cream”
“Today is my birthday :)”
“!@#$%^&*()123”
```

How to Declare a String

There are two ways to declare a String:

```
String name1 = new String("John Doe");

String name2 = "John Doe";
```

These two do the same job. They both create a new String variable that stores "John Doe".

Escape Characters

Escape Characters	Description
\t	It is used to insert a **tab** in the text at this point.
\'	It is used to insert a **single quote** character in the text at this point.
\"	It is used to insert a **double quote** character in the text at this point.
\r	It is used to insert a **carriage return** in the text at this point.
\\	It is used to insert a **backslash character** in the text at this point.
\n	It is used to insert a **new line** in the text at this point.
\f	It is used to insert a **form feed** in the text at this point.
\b	It is used to insert a **backspace** in the text at this point.

Length

- Calculates the length of the String
- Different from the highest index value
- Syntax: `.length()`

Examples:

String	Length
`"Amazing".length()`	7
`"So Amazing"`	10
`"Wow! So Cool!"`	13

Length - Try it out!

Find an online compiler and switch your language to java.

Type out the following code into your program:

```
String str = "[write anything you want here]";

System.out.println(str.length());
```

Click the Run button at the top and wait for the compiler to show your output. Your code should print out the length of your string.

Remember: Add a semicolon at the end of every statement!

Equals

- Checks if the String is equal to the String passed in the parameter
- Will return a boolean (true/false)
- Syntax: `stringName.equals(String str)`

Examples:

Statement	Return
`"Amazing".equals("Amazing")`	`true`
`"Hello".equals("hello")`	`false`
`"HI".equals("hi")`	`false`

Equals - Try it out!

Go back to your online compiler.

Type out the following code into your program:

```
String str1 = “baby shark”;
String str2 = “Baby Shark”;
System.out.println(str1.equals(str2));
```

Click the Run button at the top and wait for the compiler to show your output. Your code should print `false`.

indexOf

- Returns the index where the substring is found
- If the substring is not in the string, returns -1
- Syntax: .indexOf(“”)
- Index is different from length (Index starts counting from 0)

Examples:

Statement	Output
“Outstanding move!”.indexOf(“stand”)	3
“Hello friend”.indexOf(“Bye”)	-1

Index Concepts

Each character in a String is given an **index** starting at 0 and continuing until the end of the String.

For example, given a String `"A Good Day"`

The indexes of the String can be shown below

A		G	o	o	d		D	a	y
0	1	2	3	4	5	6	7	8	9

The index of the character `'G'` is 2.

Substring Pt.1

- A **substring** is a sequence of characters within a String
- The first type of substring method takes only the starting index and returns the rest of the string
- Uses the format of `stringName.substring(int start);`

Example: Given the String “Awesome”

A	w	e	s	o	m	e
0	1	2	3	4	5	6

`“Awesome”.substring(2);`

- Will return the string `“esome”`

`“Awesome”.substring(4);`

- Will return the string `“ome”`

Substring Pt.2

- The second type of substring method takes the start index and the end index and returns the characters in between
- Includes the starting index, doesn't include the ending index
- Uses the format of:

```
stringName.substring(int start, int end);
```

Example: Given the String "Amazing"

A	m	a	z	i	n	g
0	1	2	3	4	5	6

`"Amazing".substring(0,3);`

- Will return the string `"Ama"`

`"Amazing".substring(4,5);`

- Will return the string `"i"`

Substring - Try it out!

Go back to your online compiler.

Type out the following code into your program:

```
String str = "write anything you want here";

System.out.println(str.substring(0,2));
```

You can add more lines of code to test it out.

Click the Run button at the top and wait for the compiler to show your output. Your code should print the characters in your String.

Questions?

Contact us here:

Email:
codingpower101@gmail.com
Website:
codingpower.org