# String Basics

*Author: Vincent Lau*



## Learning Objectives

> String Creation
>
> String Concatenation
>
> String Methods I equals(), length(), chatAt()

## String Creation

```
1 // Basic way to create a string
2 String name = "Venturenix LAB";
3
4 // There are some other ways to create String, it will be covered in later
  chapters.
```

# String Concatenation

## Using the `+` Operator

The `+` operator can be used to concatenate two or more strings together. When one of the operands is a string, Java automatically performs string concatenation.

Here's an example:

```
String str1 = "Hello";
String str2 = " World!";
String result = str1 + str2; // result is "Hello World!"
```

In this example, the `+` operator concatenates the content of `str1` and `str2` to create the string "Hello World!".

## Combining Strings & Other Data Types

String concatenation can also be performed by combining strings with other data types using the `+` operator. In this case, Java **automatically converts the non-string data types to strings before concatenating them**.

Here's an example:

```
int number = 42;
String message = "The answer is: " + number; // message is "The answer is: 42"
```

In this example, the `+` operator combines the string "The answer is: " with the integer value of `number` (42), resulting in the string "The answer is: 42".

```
char a = 'A';
String message2 = "The answer is: " + a; // message2 is "The answer is: A"

int age = 25;
String message = "I am " + age + " years old."; // message is "I am 25 years old."

double price = 9.99;
String item = "Book";
String description = "The " + item + " costs $" + price; // description is "The Book costs $9.99"

```

```java
boolean isJavaFun = true;
String message = "Is Java fun? " + isJavaFun; // message is "Is Java fun? true"

char grade = 'A';
String result = "Your grade is " + grade; // result is "Your grade is A"

String name = null;
String message = "Hello, " + name; // message is "Hello, null"

// Same applied to other primitives
```

# String Method I

- At this stage, you don't need to worry about what the method is. Just to feel and try to memorize the function of each of them. Understand the result and reason to have it. Try to code in VSCode!

## Method 1: length()

The `length()` method returns the length of the string (number of characters).

```java
String str = "Hello";
int length = str.length(); // length is 5
```

## Method 2: equals()

The `equals()` method compares the content of the current string with the content of another object (typically another string) and returns a boolean value indicating whether they are equal or not. It overrides the `equals()` method inherited from the `Object` class.

Here's an example demonstrating the usage of the `equals()` method:

```java
String str1 = "Hello";
String str2 = "Hello";
String str3 = "World";

boolean isEqual1 = str1.equals(str2); // isEqual1 is true
boolean isEqual2 = str1.equals(str3); // isEqual2 is false
```

In the example above, `isEqual1` is `true` because `str1` and `str2` have the same content ("Hello"). However, `isEqual2` is `false` because `str1` and `str3` have different content

("Hello" vs. "World").

## Method 3: charAt(int index)

The `charAt()` method returns the character at the specified index within the string.

**The index starts from 0.**

```
String str = "Hello";
char ch = str.charAt(0); // ch = 'H'
char ch = str.charAt(1); // ch = 'e'
char ch = str.charAt(2); // ch = 'l'
char ch = str.charAt(3); // ch = 'l'
char ch = str.charAt(4); // ch = 'o'
char ch = str.charAt(5); // what is the value of ch?
```