**Java programming**

There are 4 types of variables in Java programming language:

* Instance Variables (Non-Static Fields)
* Class Variables (Static Fields)
* Local Variables
* Parameters

class Java {

public static void main(String[] args) {

int februaryDays = 29;

String result;

// ternary operator

result = (februaryDays == 28) ? "Not a leap year" : "Leap year";

System.out.println(result);

}

}

String[] array = new String[100];

int[] age = {12, 4, 5, 2, 5};

**Example 1: Java Constructor**

class Main {

private String name;

// constructor

Main() {

System.out.println("Constructor Called:");

name = "Programiz";

}

public static void main(String[] args) {

// constructor is invoked while

// creating an object of the Main class

Main obj = new Main();

System.out.println("The name is " + obj.name);

}

}

In Java, two or more [methods](https://www.programiz.com/java-programming/methods) may have the same name if they differ in parameters (different number of parameters, different types of parameters, or both). These methods are called overloaded methods and this feature is called method overloading. For example:

void func() { ... }

void func(int a) { ... }

float func(double a) { ... }

float func(int a, float b) { ... }

String language;

// constructor with no parameter

Main() {

this.language = "Java";

}

// constructor with a single parameter

Main(String language) {

this.language = language;

}

public void getName() {

System.out.println("Programming Langauage: " + this.language);

}

public static void main(String[] args) {

// call constructor with no parameter

Main obj1 = new Main();

// call constructor with a single parameter

Main obj2 = new Main("Python");

obj1.getName();

obj2.getName();

}

}

**Output**:

Programming Language: Java

Programming Language: Python

|  |  |
| --- | --- |
| Methods | Description |
| [substring()](https://www.programiz.com/java-programming/library/string/substring) | returns the substring of the string |
| [replace()](https://www.programiz.com/java-programming/library/string/replace) | replaces the specified old character with the specified new character |
| [charAt()](https://www.programiz.com/java-programming/library/string/charat) | returns the character present in the specified location |
| [getBytes()](https://www.programiz.com/java-programming/library/string/getbytes) | converts the string to an array of bytes |
| [indexOf()](https://www.programiz.com/java-programming/library/string/indexof) | returns the position of the specified character in the string |
| [compareTo()](https://www.programiz.com/java-programming/library/string/compareto) | compares two strings in the dictionary order |
| [trim()](https://www.programiz.com/java-programming/library/string/trim) | removes any leading and trailing whitespaces |
| [format()](https://www.programiz.com/java-programming/library/string/format) | returns a formatted string |
| [split()](https://www.programiz.com/java-programming/library/string/split) | breaks the string into an array of strings |
| [toLowerCase()](https://www.programiz.com/java-programming/library/string/tolowercase) | converts the string to lowercase |
| [toUpperCase()](https://www.programiz.com/java-programming/library/string/touppercase) | converts the string to uppercase |
| [valueOf()](https://www.programiz.com/java-programming/library/string/valueof) | returns the string representation of the specified argument |
| [toCharArray()](https://www.programiz.com/java-programming/library/string/tochararray) | converts the string to a char array |

 we have used the setter method (setName()) to assign value to the variable and the getter method (getName()) to access the variable.