

Validating a phone number using regular expression is tricky because the phone number can be written in many formats and can have extensions also.

For example, here are some of the common way of writing phone numbers:

* 1234567890
* 123-456-7890
* 123-456-7890 x1234
* 123-456-7890 ext1234
* (123)-456-7890
* 123.456.7890
* 123 456 7890

**Phone Number Validation in Java**

Here I am using [**java regular expressions**](https://www.journaldev.com/634/regular-expression-in-java-regex-example) to validate any of the above format phone numbers.

package com.journaldev.util;

public class PhoneNumberValidator {

public static void main(String[] args) {

System.out.println("Phone number 1234567890 validation result: "+validatePhoneNumber("1234567890"));

System.out.println("Phone number 123-456-7890 validation result: "+validatePhoneNumber("123-456-7890"));

System.out.println("Phone number 123-456-7890 x1234 validation result: "+validatePhoneNumber("123-456-7890 x1234"));

System.out.println("Phone number 123-456-7890 ext1234 validation result: "+validatePhoneNumber("123-456-7890 ext1234"));

System.out.println("Phone number (123)-456-7890 validation result: "+validatePhoneNumber("(123)-456-7890"));

System.out.println("Phone number 123.456.7890 validation result: "+validatePhoneNumber("123.456.7890"));

System.out.println("Phone number 123 456 7890 validation result: "+validatePhoneNumber("123 456 7890"));

}

private static boolean validatePhoneNumber(String phoneNo) {

//validate phone numbers of format "1234567890"

if (phoneNo.matches("\\d{10}")) return true;

//validating phone number with -, . or spaces

else if(phoneNo.matches("\\d{3}[-\\.\\s]\\d{3}[-\\.\\s]\\d{4}")) return true;

//validating phone number with extension length from 3 to 5

else if(phoneNo.matches("\\d{3}-\\d{3}-\\d{4}\\s(x|(ext))\\d{3,5}")) return true;

//validating phone number where area code is in braces ()

else if(phoneNo.matches("\\(\\d{3}\\)-\\d{3}-\\d{4}")) return true;

//return false if nothing matches the input

else return false;

}

}

Here is the output of the above phone number regex validation program.

Phone number 1234567890 validation result: true

Phone number 123-456-7890 validation result: true

Phone number 123-456-7890 x1234 validation result: true

Phone number 123-456-7890 ext1234 validation result: true

Phone number (123)-456-7890 validation result: true

Phone number 123.456.7890 validation result: true

Phone number 123 456 7890 validation result: true

The best part of this program is that it’s easy to understand and you can extend it easily to support other phone number formats also.

# **How to validate phone number in Java (regular expression)**

By [mkyong](https://www.mkyong.com/author/mkyong/" \o "mkyong class=) | January 27, 2009 | Updated : August 30, 2012 | Viewed : 185,968 | +292 pv/w

Regular expression pattern in Java always is the best method to validate an user’s phone number. Here i provide a regex pattern to determines if the phone number is in correct format, the pattern **force starting with 3 digits follow by a “-” and 7 digits at the end**.

\\d{3}-\\d{7}

**Explanation**  
\\d = only digit allow  
{3} = length

All phone numbers must in “xxx-xxxxxxx” format. For example  
1) 012-6677889 – Passed  
2) 01216677889 – Failed , “-” missing  
3) A12-6677889 – Failed , only digit allow  
4) 012-66778899 – Failed, only 7 digits at the end

Full source code of phone number validation in Java

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class ValidatePhoneNumber {

public static void main(String[] argv) {

String sPhoneNumber = "605-8889999";

//String sPhoneNumber = "605-88899991";

//String sPhoneNumber = "605-888999A";

Pattern pattern = Pattern.compile("\\d{3}-\\d{7}");

Matcher matcher = pattern.matcher(sPhoneNumber);

if (matcher.matches()) {

System.out.println("Phone Number Valid");

}

else

{

System.out.println("Phone Number must be in the form XXX-XXXXXXX");

}

}

}

Regex(@"^[0-9]{10}$");

use this it will work properly. it will also check if any alphabets are there or not. in this we can only enter 10 digits number