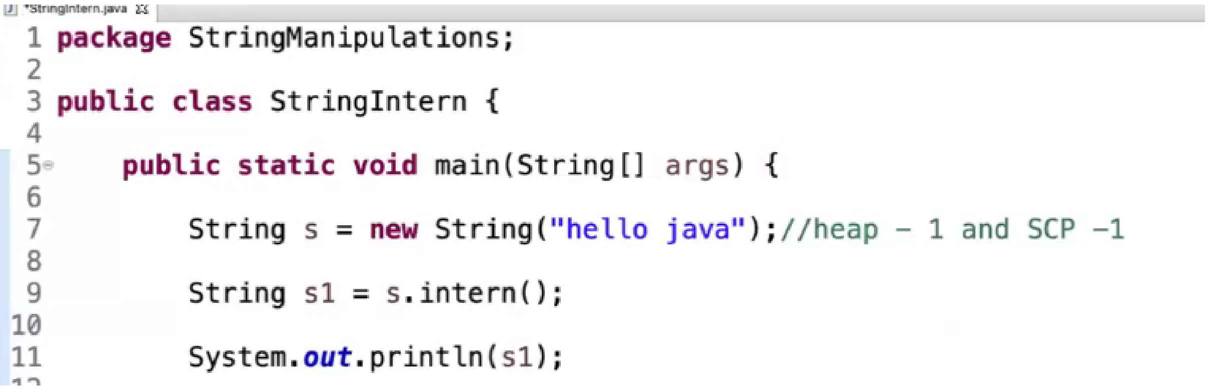
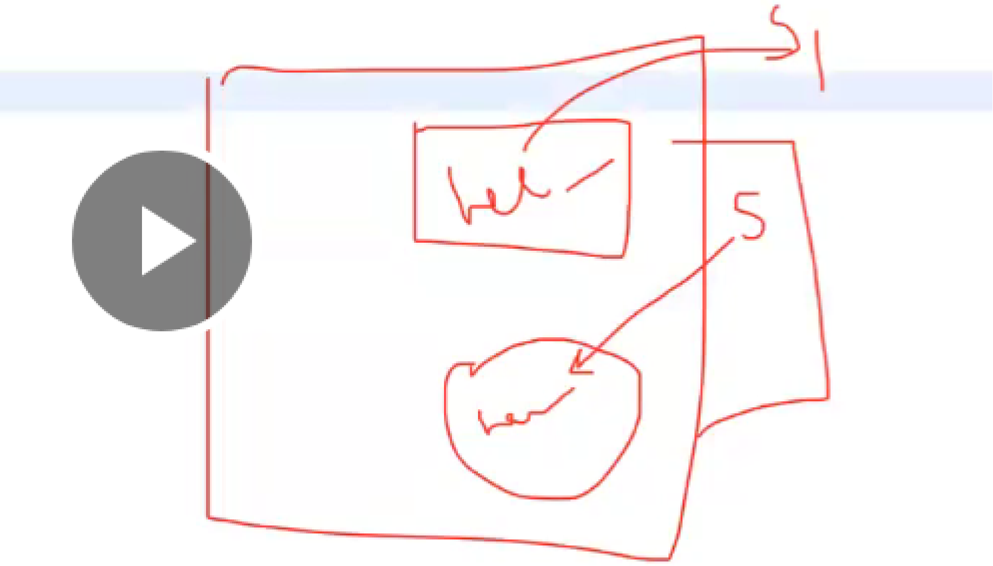
Intern-

When scp does not have reference give reference to it. This is for strings created using new. because the new keyword creates strings in two areas, but the scp one is actually unreferenced, which intern can help. if scp not referred then garbage collector will come and collect it.

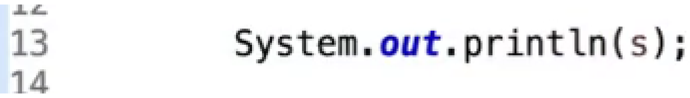


*StringIntern.java

```java
package StringManipulations;

public class StringIntern {

    public static void main(String[] args) {

        String s = new String("hello java");//heap - 1 and SCP -1

        String s1 = s.intern();

        System.out.println(s1);
```

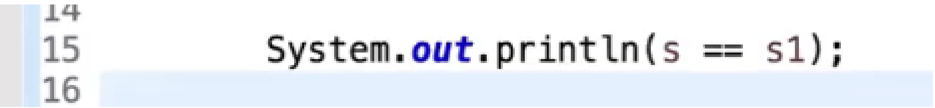


Hello java



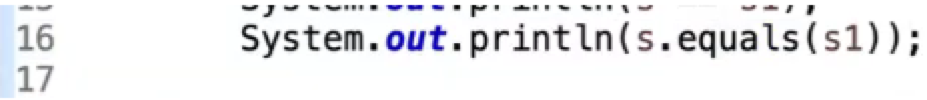
```java
        System.out.println(s);
```

Hello java



```
        System.out.println(s == s1);
```

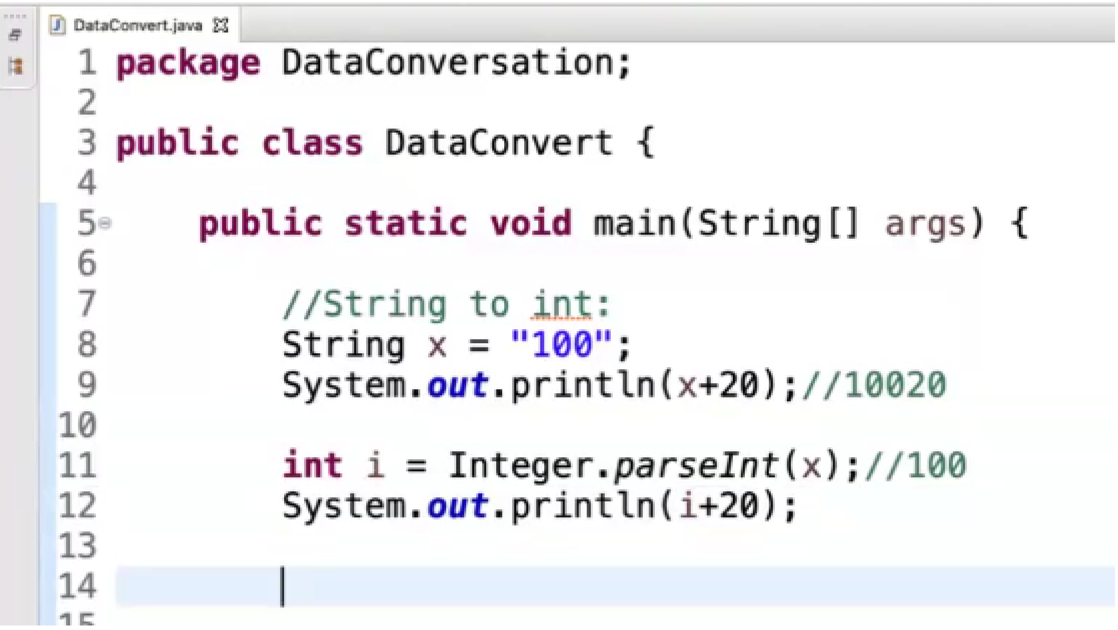
False



```
        System.out.println(s == s1);
        System.out.println(s.equals(s1));
```

True

data conversion-

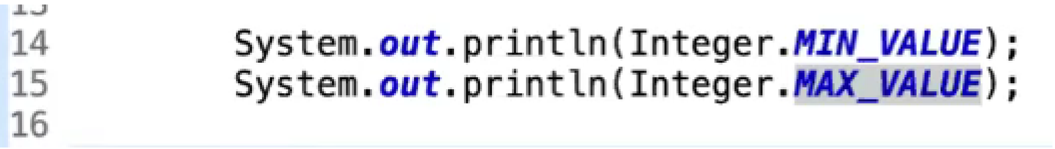


```
package DataConversation;

public class DataConvert {

    public static void main(String[] args) {

        //String to int:
        String x = "100";
        System.out.println(x+20);//10020

        int i = Integer.parseInt(x);//100
        System.out.println(i+20);

```

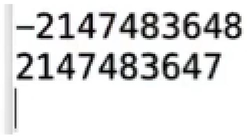
10020

120

Range-

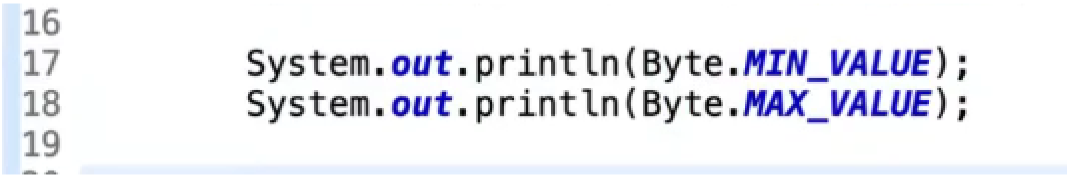


```
        System.out.println(Integer.MIN_VALUE);
        System.out.println(Integer.MAX_VALUE);
```

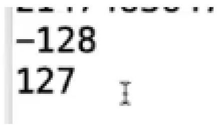


```
-2147483648
2147483647
```

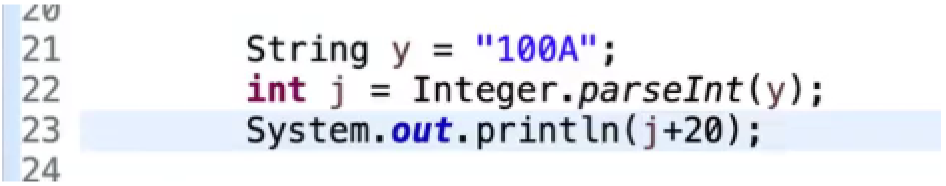
Byte-



```
        System.out.println(Byte.MIN_VALUE);
        System.out.println(Byte.MAX_VALUE);
```



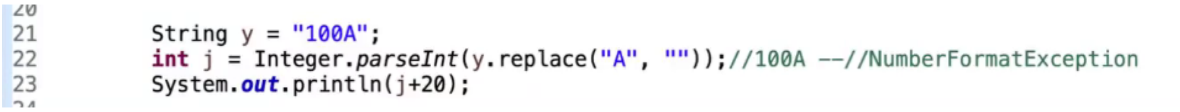
```
-128
127
```



```
        String y = "100A";
        int j = Integer.parseInt(y);
        System.out.println(j+20);
```

Number format exception.

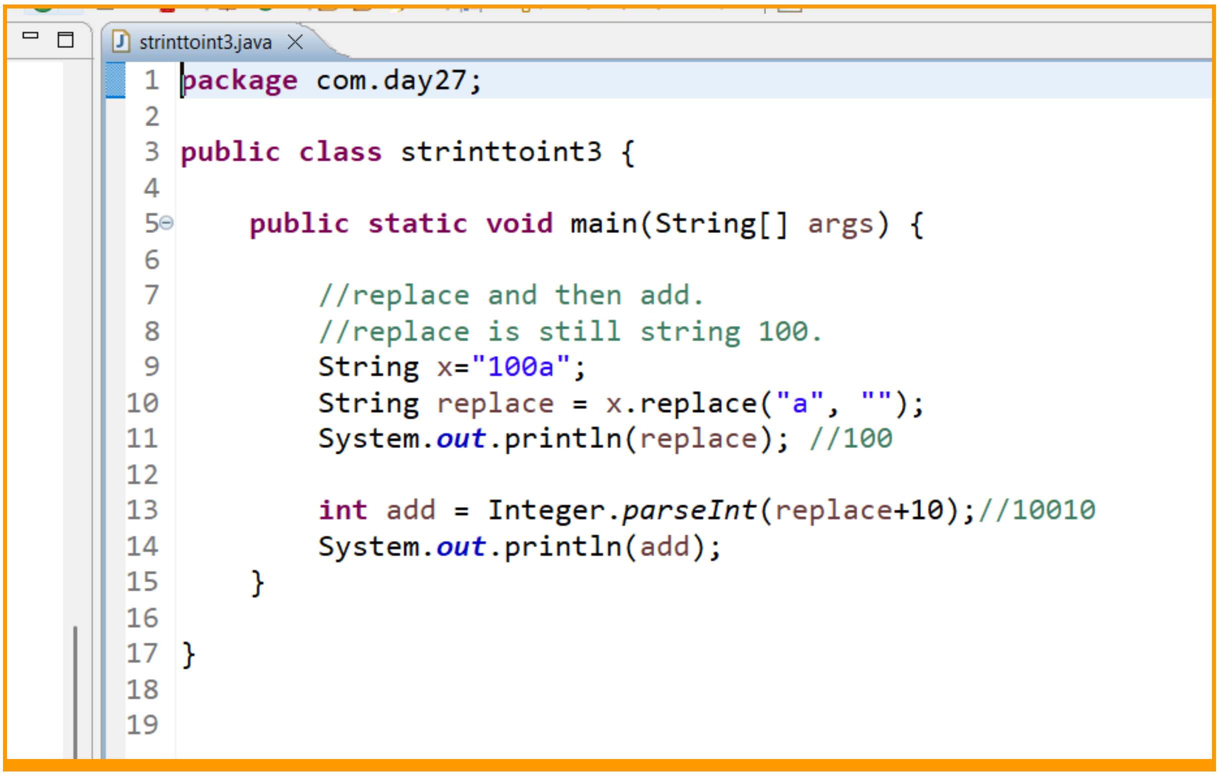
Solving it using replace-



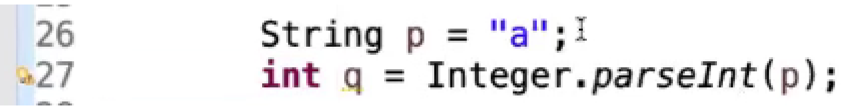
```
        String y = "100A";
        int j = Integer.parseInt(y.replace("A", ""));//100A --//NumberFormatException
        System.out.println(j+20);
```

120

paste strinttoint3-

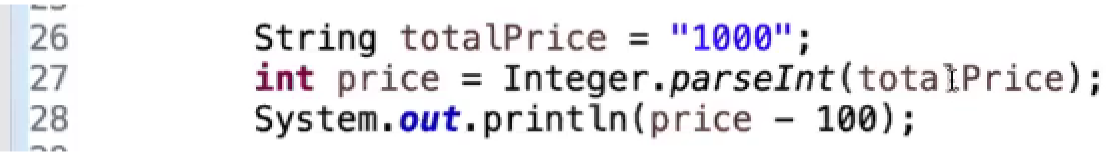


```
package com.day27;

public class stringtoint3 {

    public static void main(String[] args) {

        //replace and then add.
        //replace is still string 100.
        String x="100a";
        String replace = x.replace("a", "");
        System.out.println(replace); //100

        int add = Integer.parseInt(replace+10);//10010
        System.out.println(add);
    }

}
```



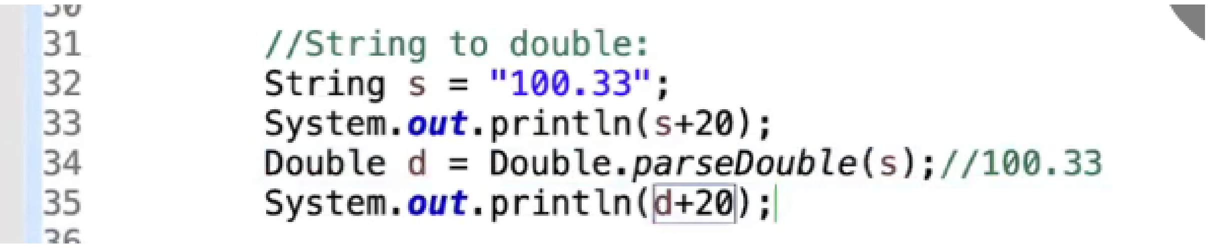
```
        String p = "a";
        int q = Integer.parseInt(p);
```

Number format exception.



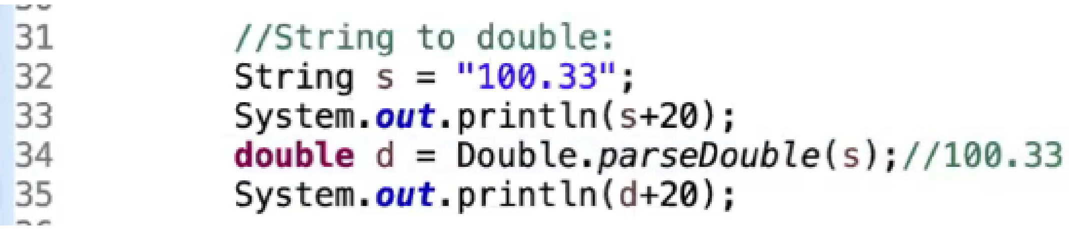
```
        String totalPrice = "1000";
        int price = Integer.parseInt(totalPrice);
        System.out.println(price - 100);
```

900



```
        //String to double:
        String s = "100.33";
        System.out.println(s+20);
        Double d = Double.parseDouble(s);//100.33
        System.out.println(d+20);
```

This also ok



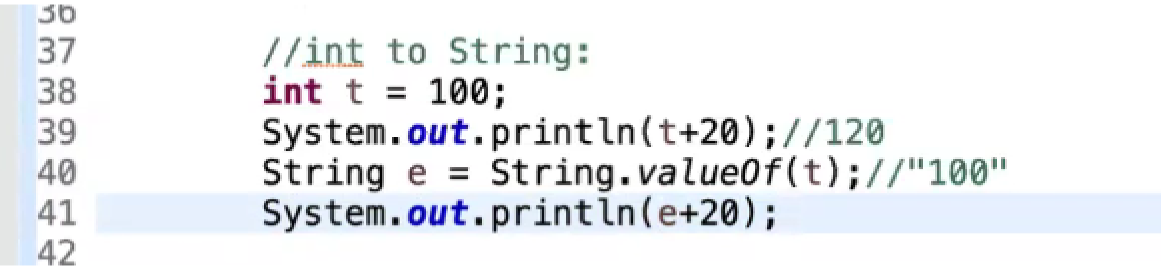
```
        //String to double:
        String s = "100.33";
        System.out.println(s+20);
        double d = Double.parseDouble(s);//100.33
        System.out.println(d+20);
```

120.33

Wrapper can hold primitive and non primitive values.

Value of-

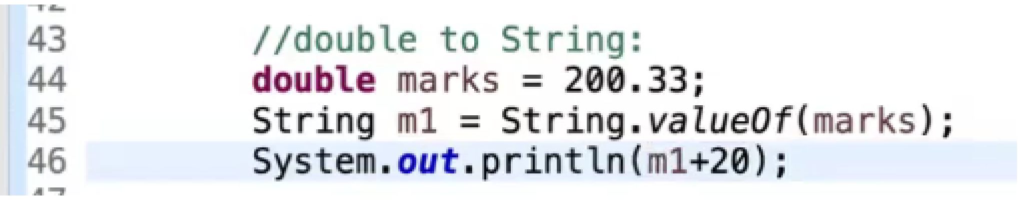
Convert anything to string format.



```
        //int to String:
        int t = 100;
        System.out.println(t+20);//120
        String e = String.valueOf(t);//"100"
        System.out.println(e+20);
```

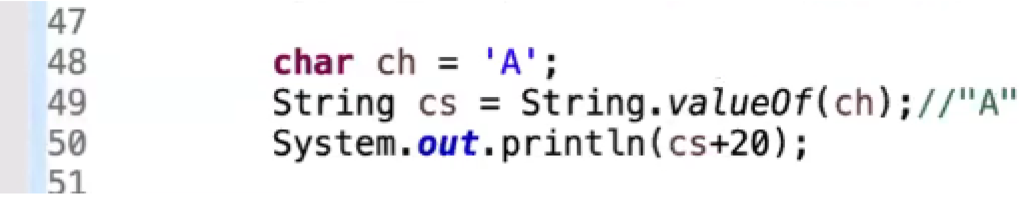
120

10020



```
        //double to String:
        double marks = 200.33;
        String m1 = String.valueOf(marks);
        System.out.println(m1+20);
```

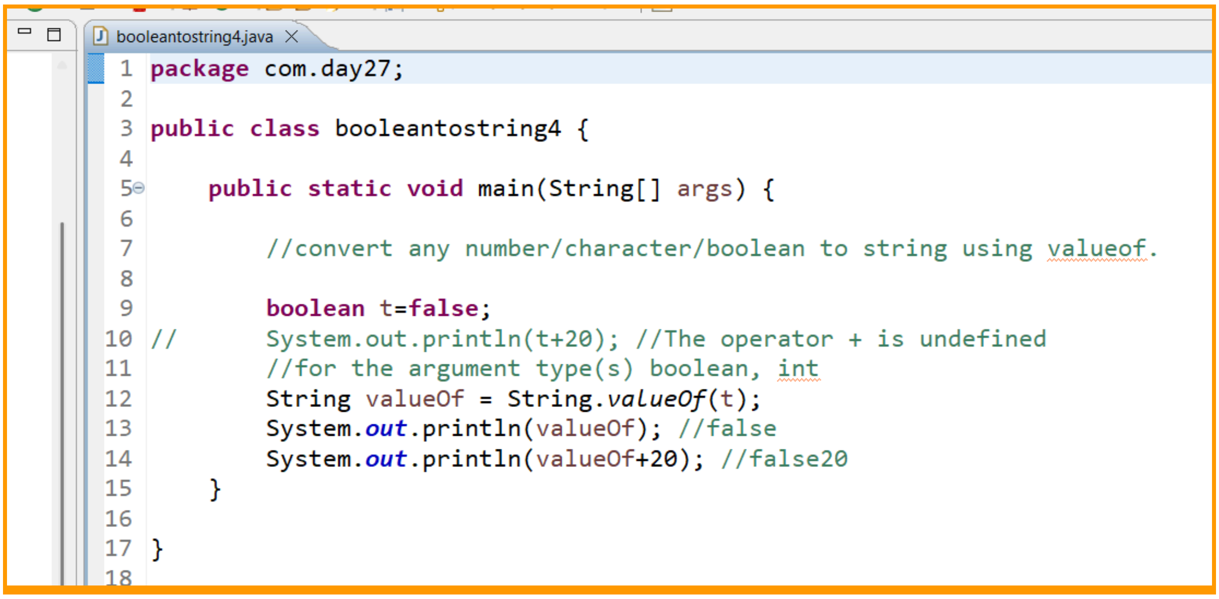
200.3320



```
        char ch = 'A';
        String cs = String.valueOf(ch);//"A"
        System.out.println(cs+20);
```

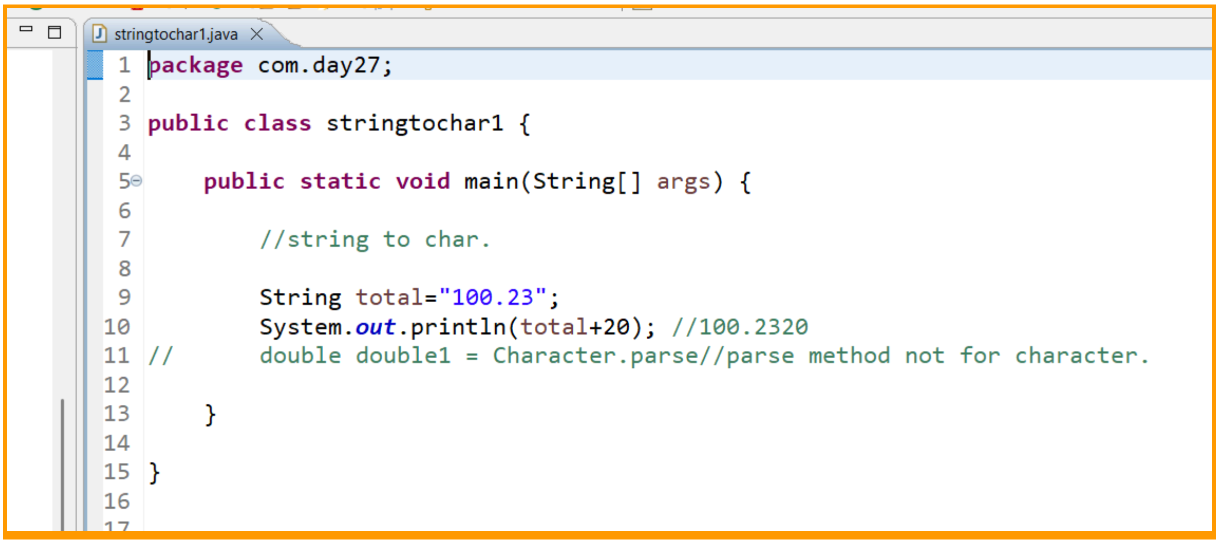
A20

paste booleantostring4-



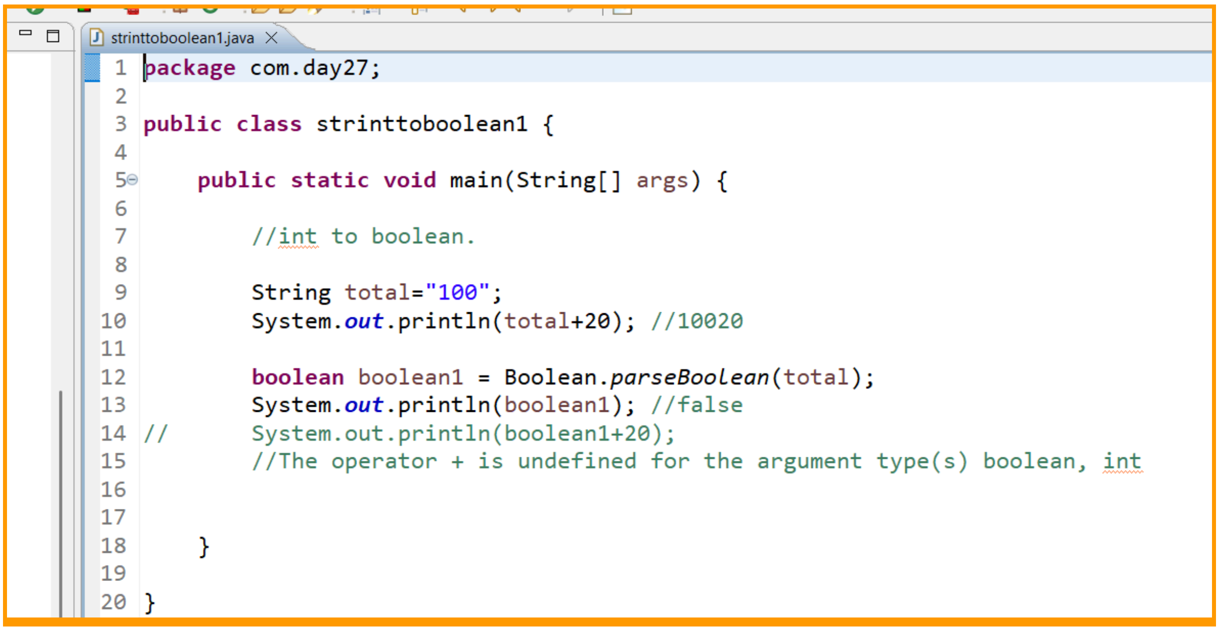
```
package com.day27;

public class booleantostring4 {

    public static void main(String[] args) {

        //convert any number/character/boolean to string using valueof.

        boolean t=false;
//      System.out.println(t+20); //The operator + is undefined
        //for the argument type(s) boolean, int
        String valueOf = String.valueOf(t);
        System.out.println(valueOf); //false
        System.out.println(valueOf+20); //false20
    }

}
```

paste stringtochar1-



```
package com.day27;

public class stringtochar1 {

    public static void main(String[] args) {

        //string to char.

        String total="100.23";
        System.out.println(total+20); //100.2320
//      double double1 = Character.parse//parse method not for character.

    }

}
```

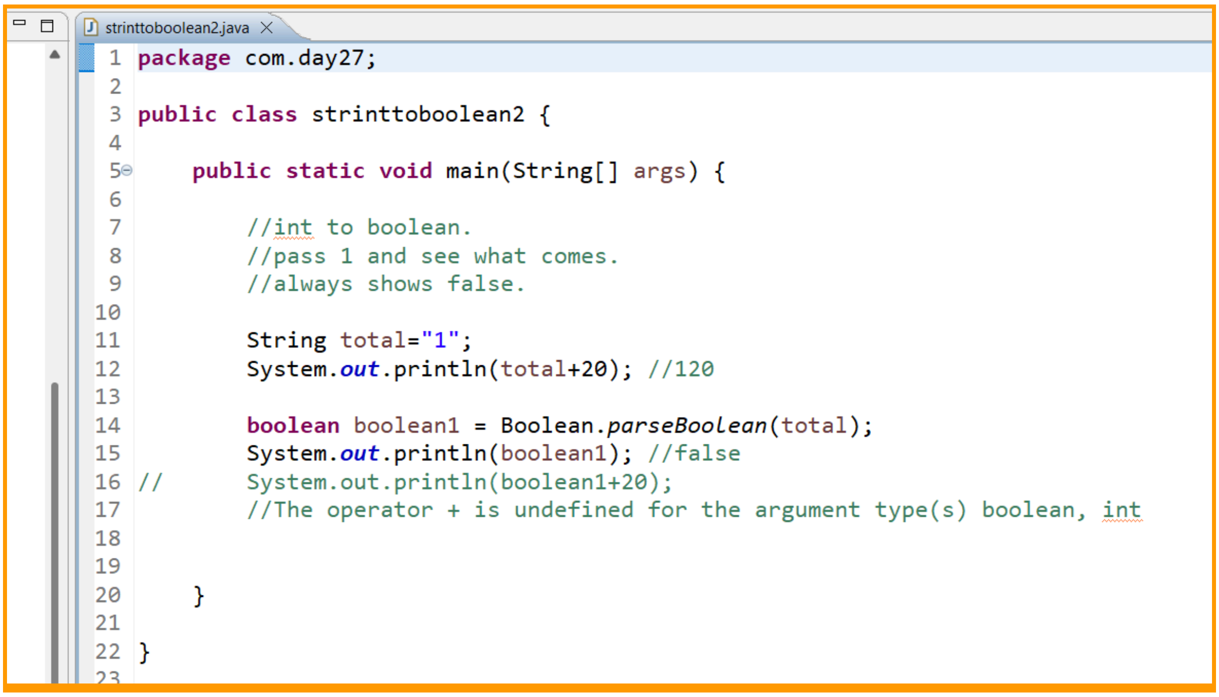
paste stringtoboolean1



```java
package com.day27;

public class strinttoboolean1 {

    public static void main(String[] args) {

        //int to boolean.

        String total="100";
        System.out.println(total+20); //10020

        boolean boolean1 = Boolean.parseBoolean(total);
        System.out.println(boolean1); //false
//      System.out.println(boolean1+20);
        //The operator + is undefined for the argument type(s) boolean, int


    }

}
```

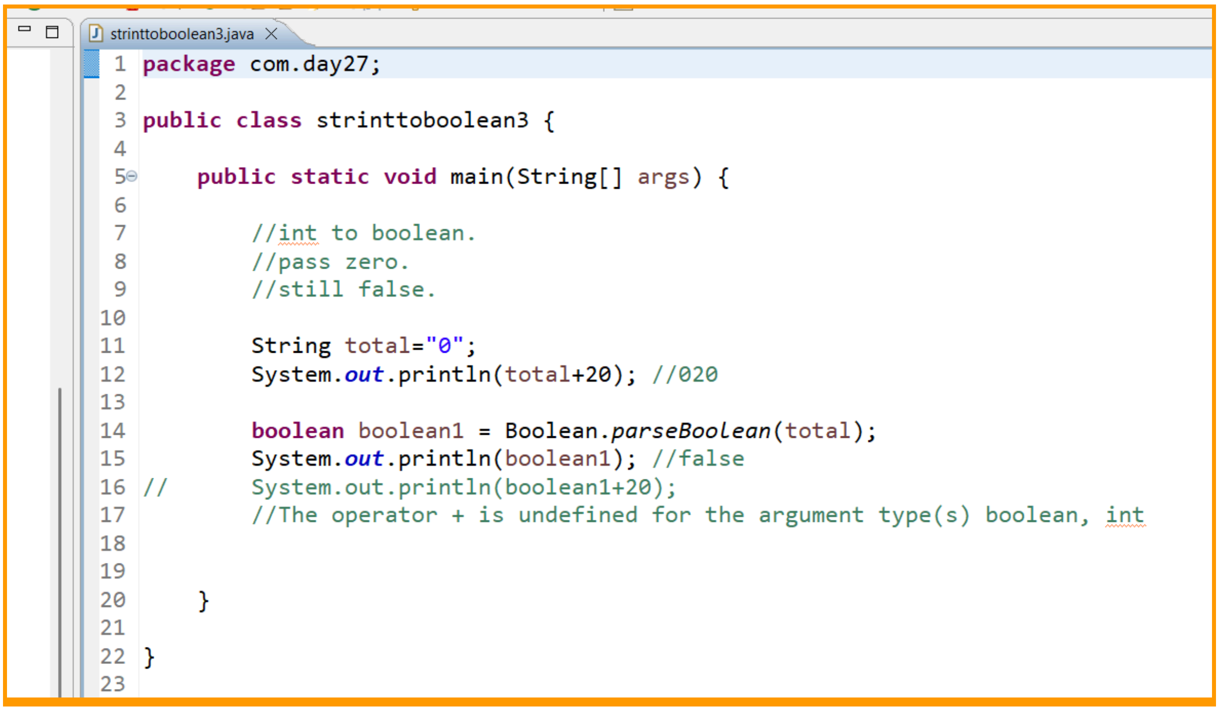
paste strinttoboolean1 done above.

paste strinttoboolean2



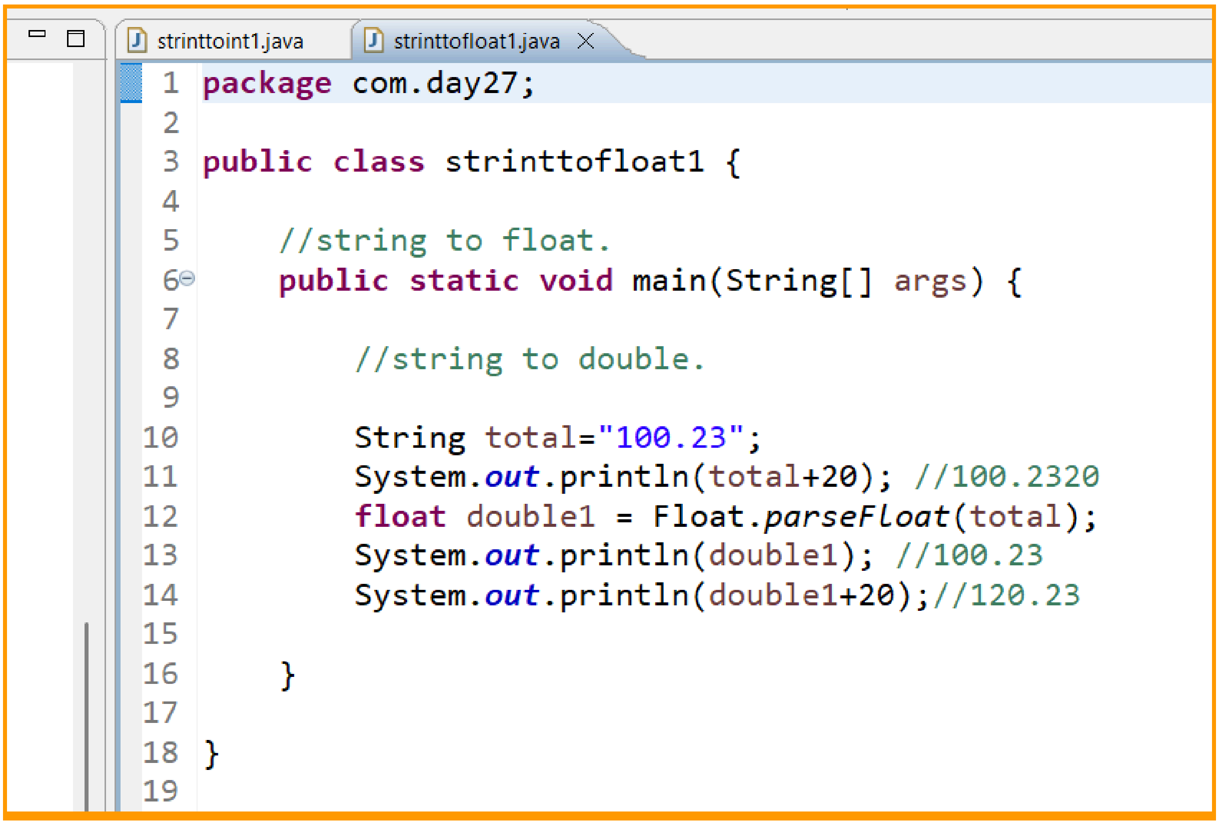
```java
package com.day27;

public class strinttoboolean2 {

    public static void main(String[] args) {

        //int to boolean.
        //pass 1 and see what comes.
        //always shows false.

        String total="1";
        System.out.println(total+20); //120

        boolean boolean1 = Boolean.parseBoolean(total);
        System.out.println(boolean1); //false
//      System.out.println(boolean1+20);
        //The operator + is undefined for the argument type(s) boolean, int


    }

}
```

paste strinttoboolean3

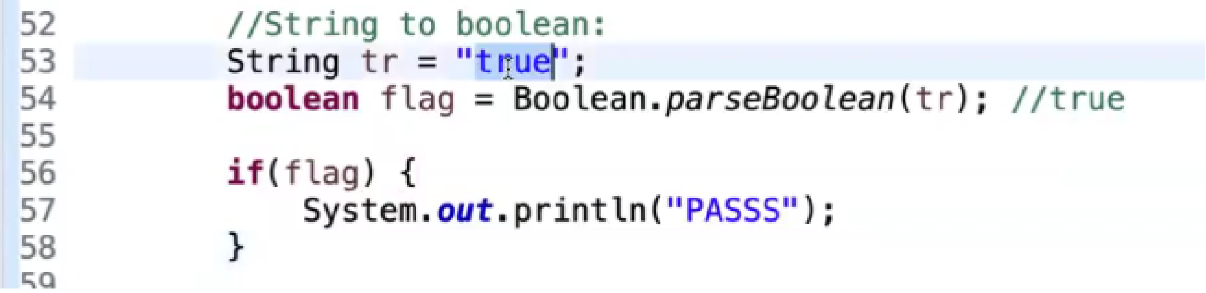


```java
package com.day27;

public class strinttoboolean3 {

    public static void main(String[] args) {

        //int to boolean.
        //pass zero.
        //still false.

        String total="0";
        System.out.println(total+20); //020

        boolean boolean1 = Boolean.parseBoolean(total);
        System.out.println(boolean1); //false
//      System.out.println(boolean1+20);
        //The operator + is undefined for the argument type(s) boolean, int


    }

}

```

paste strinttofloat1



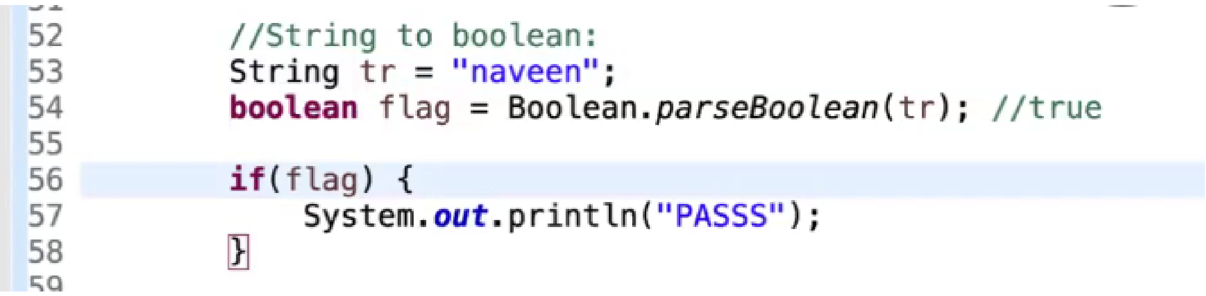
```java
package com.day27;

public class strinttofloat1 {

    //string to float.
    public static void main(String[] args) {

        //string to double.

        String total="100.23";
        System.out.println(total+20); //100.2320
        float double1 = Float.parseFloat(total);
        System.out.println(double1); //100.23
        System.out.println(double1+20);//120.23

    }

}

```



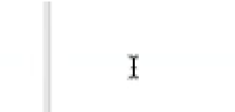
```
//String to boolean:
String tr = "true";
boolean flag = Boolean.parseBoolean(tr); //true

if(flag) {
    System.out.println("PASSS");
}
```

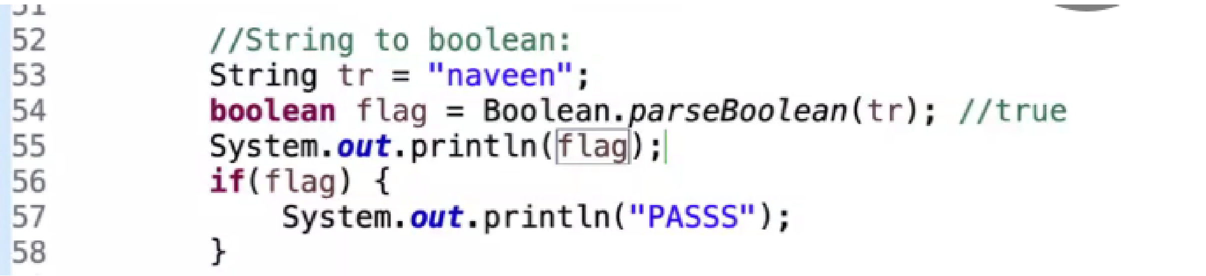
Pass

Give invalid Boolean value-

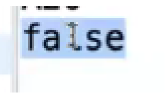


```
//String to boolean:
String tr = "naveen";
boolean flag = Boolean.parseBoolean(tr); //true

if(flag) {
    System.out.println("PASSS");
}
```

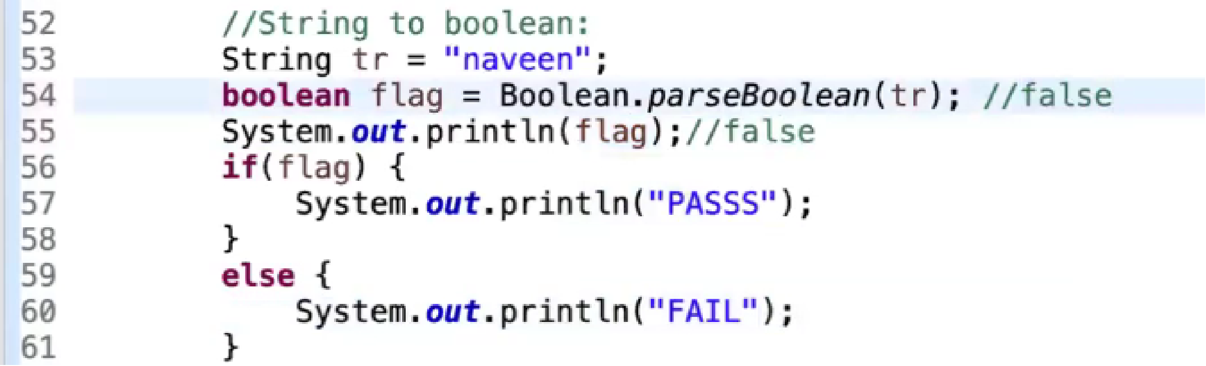




```
//String to boolean:
String tr = "naveen";
boolean flag = Boolean.parseBoolean(tr); //true
System.out.println(flag);
if(flag) {
    System.out.println("PASSS");
}
```



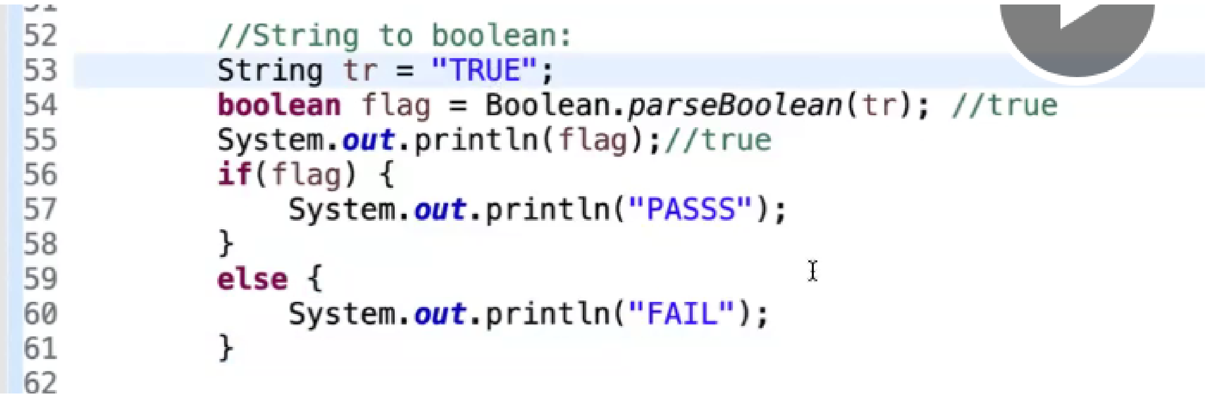
```
false
```



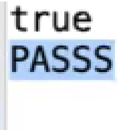
```
//String to boolean:
String tr = "naveen";
boolean flag = Boolean.parseBoolean(tr); //false
System.out.println(flag);//false
if(flag) {
    System.out.println("PASSS");
}
else {
    System.out.println("FAIL");
}
```

False

Fail

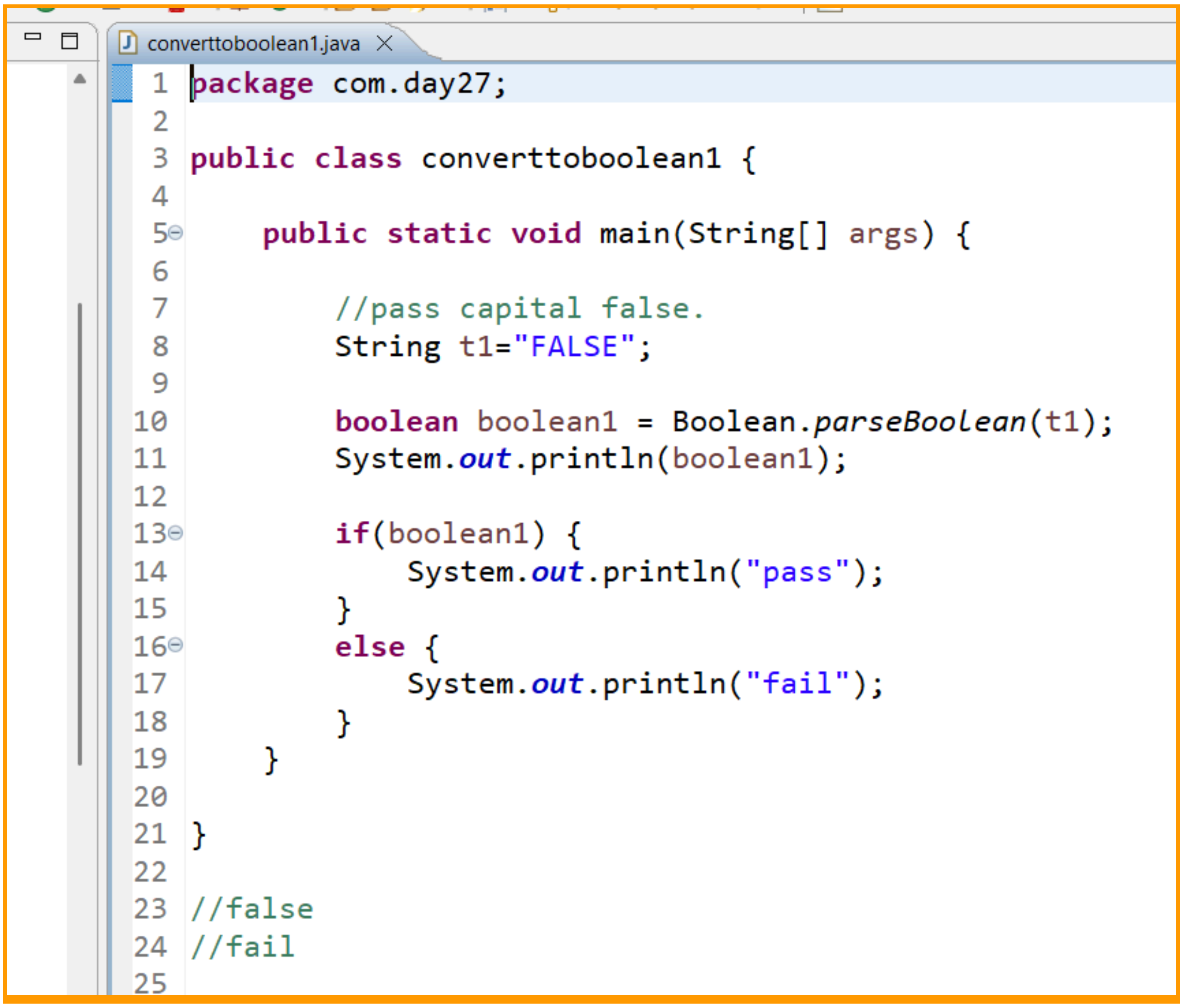


```
        //String to boolean:
        String tr = "TRUE";
        boolean flag = Boolean.parseBoolean(tr); //true
        System.out.println(flag);//true
        if(flag) {
            System.out.println("PASSS");
        }
        else {
            System.out.println("FAIL");
        }
```

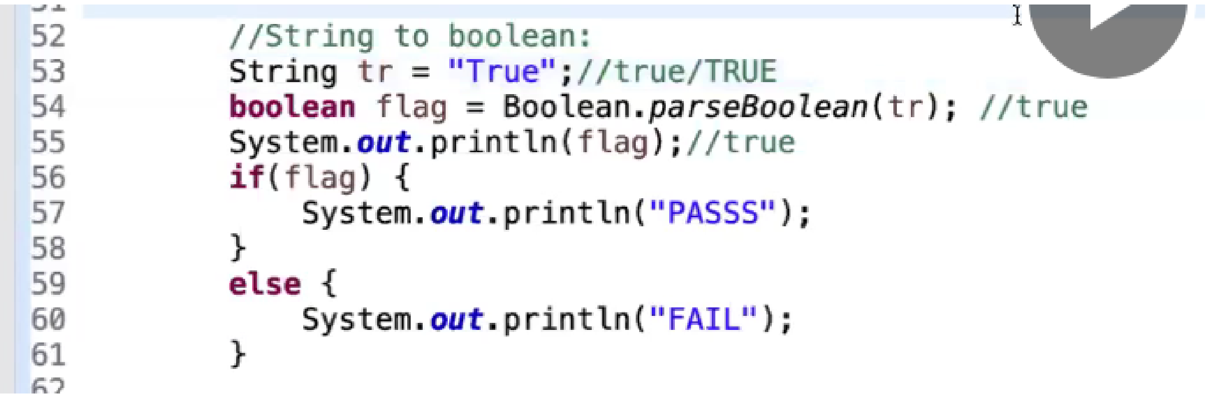


```
true
PASSS
```

paste converttoboolean1



```
package com.day27;

public class converttoboolean1 {

    public static void main(String[] args) {

        //pass capital false.
        String t1="FALSE";

        boolean boolean1 = Boolean.parseBoolean(t1);
        System.out.println(boolean1);

        if(boolean1) {
            System.out.println("pass");
        }
        else {
            System.out.println("fail");
        }
    }

}

//false
//fail
```

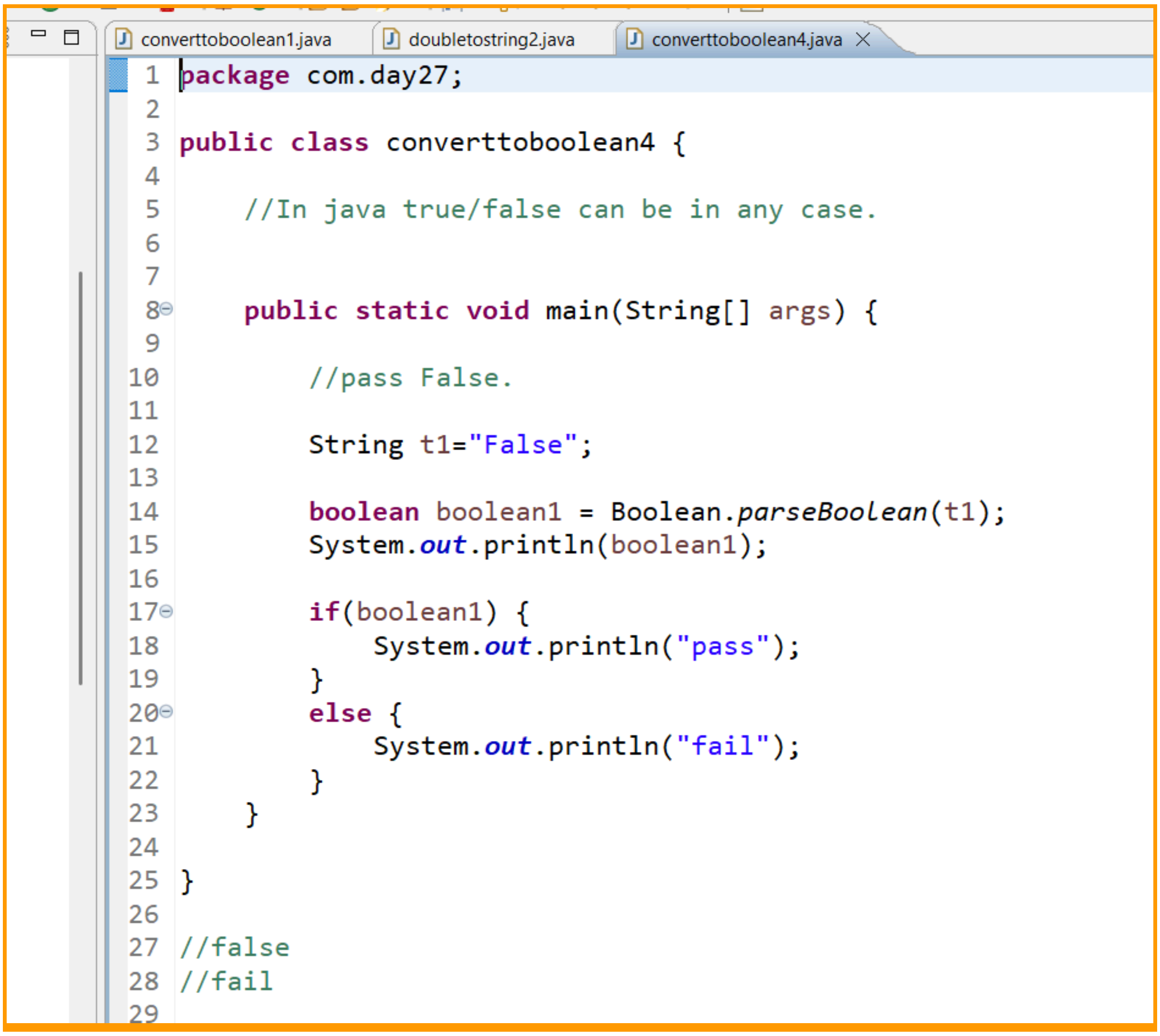


```
        //String to boolean:
        String tr = "True";//true/TRUE
        boolean flag = Boolean.parseBoolean(tr); //true
        System.out.println(flag);//true
        if(flag) {
            System.out.println("PASSS");
        }
        else {
            System.out.println("FAIL");
        }
```

True

Pass

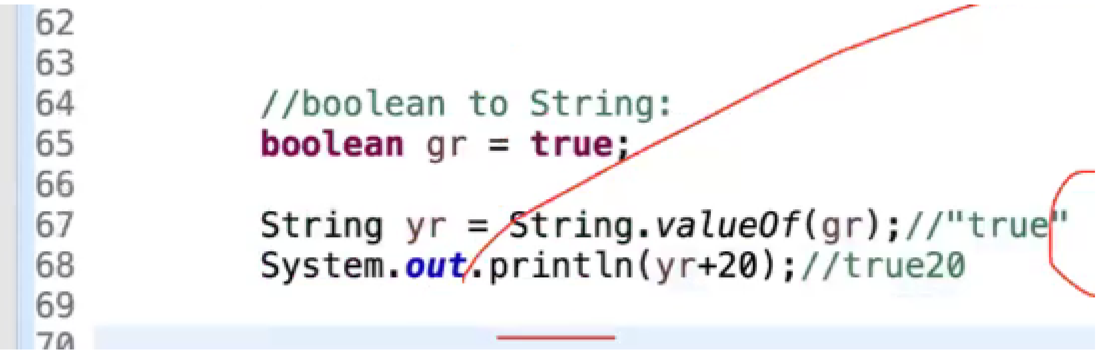
paste converttoboolean4



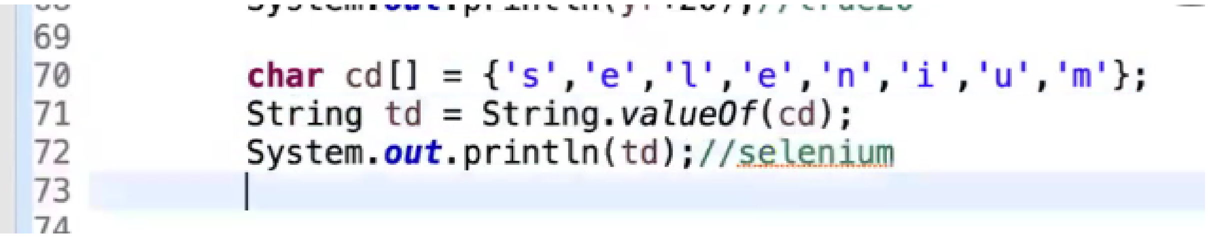
```java
package com.day27;

public class converttoboolean4 {

    //In java true/false can be in any case.


    public static void main(String[] args) {

        //pass False.

        String t1="False";

        boolean boolean1 = Boolean.parseBoolean(t1);
        System.out.println(boolean1);

        if(boolean1) {
            System.out.println("pass");
        }
        else {
            System.out.println("fail");
        }
    }

}

//false
//fail
```

In java true/false can be in any case.

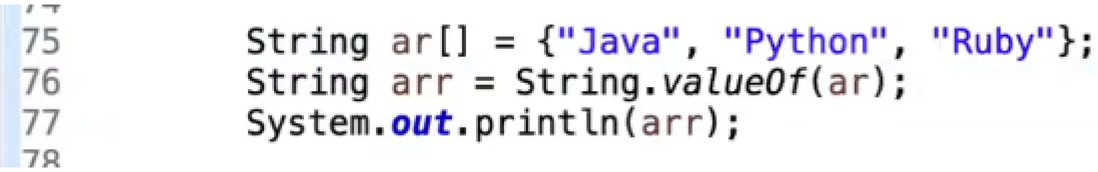
~~44,00~~



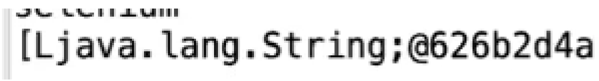
```java

        //boolean to String:
        boolean gr = true;

        String yr = String.valueOf(gr);//"true"
        System.out.println(yr+20);//true20

```



```
        System.out.println(j+20);//true20

        char cd[] = {'s','e','l','e','n','i','u','m'};
        String td = String.valueOf(cd);
        System.out.println(td);//selenium
```

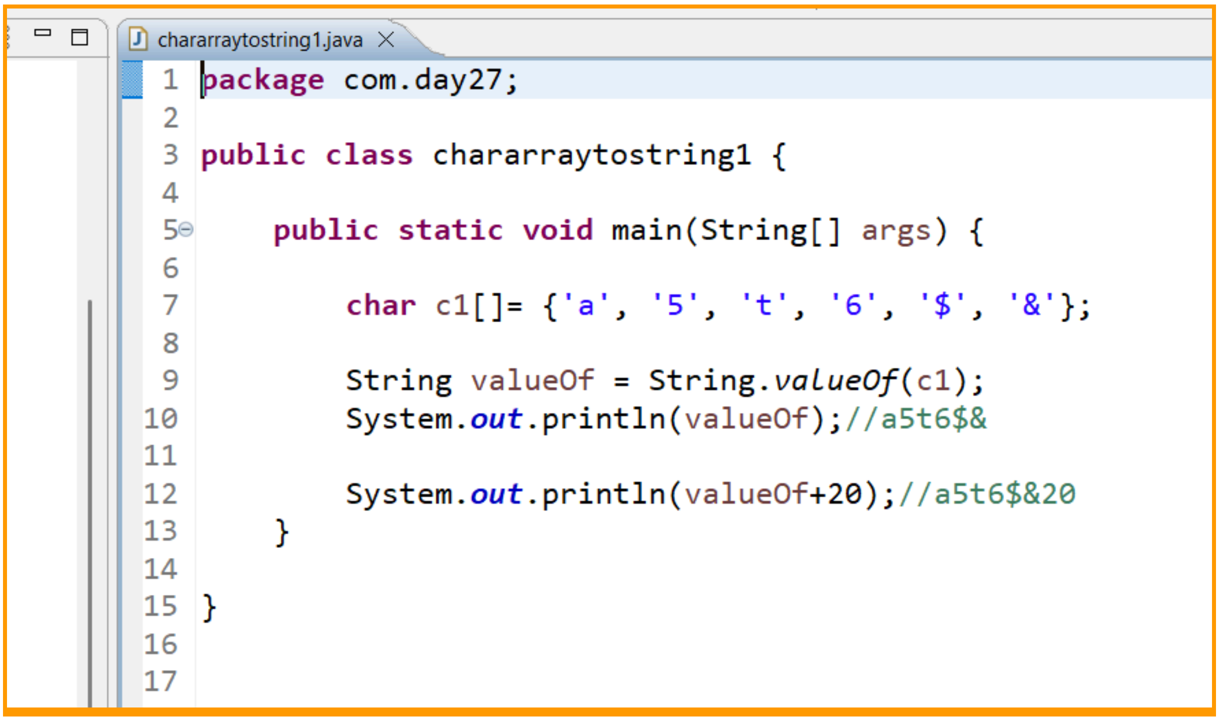


```
        String ar[] = {"Java", "Python", "Ruby"};
        String arr = String.valueOf(ar);
        System.out.println(arr);
```



```
selenium
[Ljava.lang.String;@626b2d4a
```

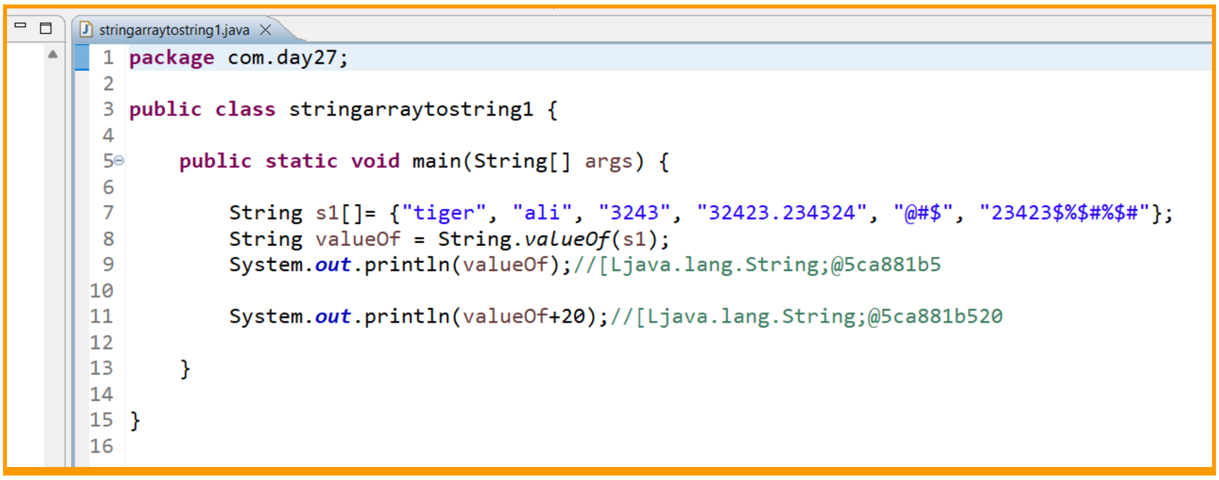
paste chararraytostring1



chararraytostring1.java

```
package com.day27;

public class chararraytostring1 {

    public static void main(String[] args) {

        char c1[]= {'a', '5', 't', '6', '$', '&'};

        String valueOf = String.valueOf(c1);
        System.out.println(valueOf);//a5t6$&

        System.out.println(valueOf+20);//a5t6$&20
    }

}
```

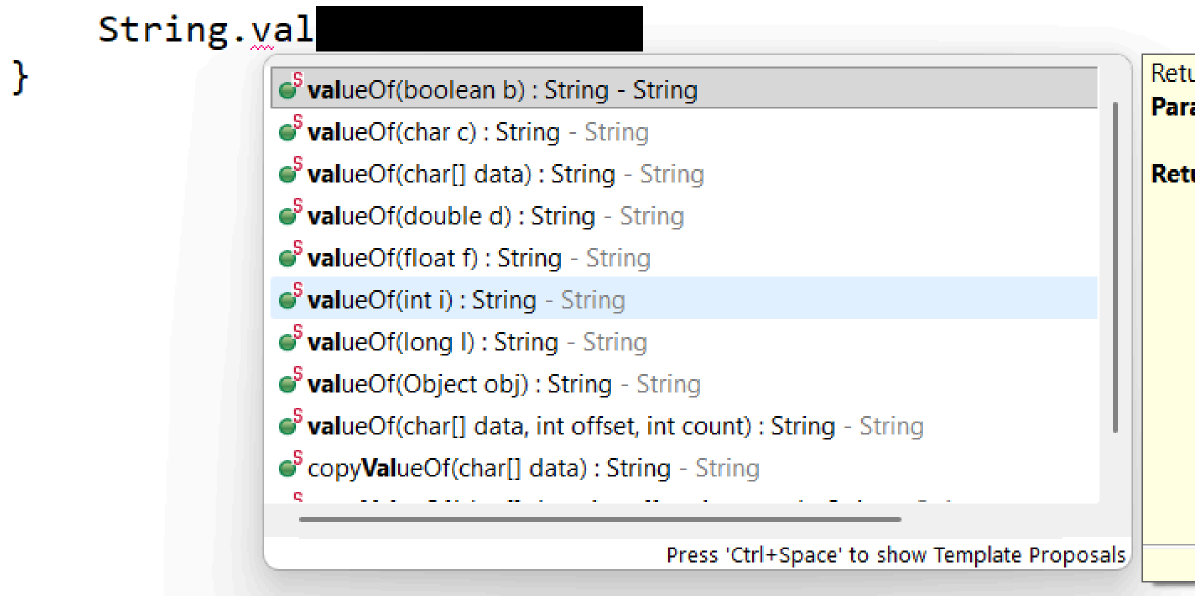
paste stringarraytostring1



```
package com.day27;

public class stringarraytostring1 {

    public static void main(String[] args) {

        String s1[]= {"tiger", "ali", "3243", "32423.234324", "@#$", "23423$%$#%$#"};
        String valueOf = String.valueOf(s1);
        System.out.println(valueOf);//[Ljava.lang.String;@5ca881b5

        System.out.println(valueOf+20);//[Ljava.lang.String;@5ca881b520

    }

}
```

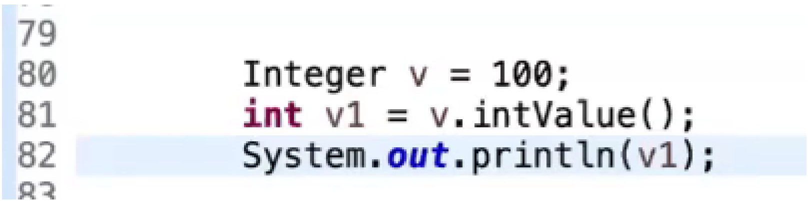
Value of not applicable for any arrays except character. (48)-->

we can see only char array here.



```
    String.val[redacted]
}
```

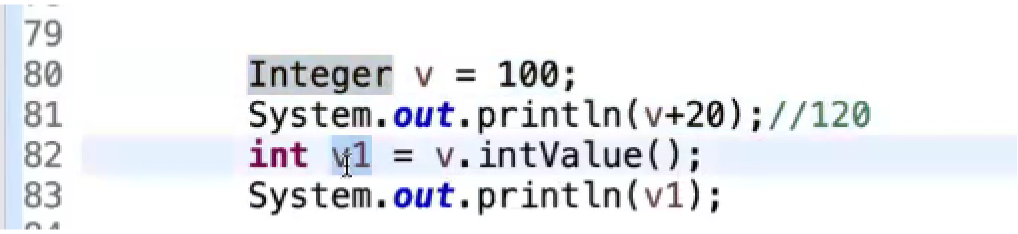
valueOf(boolean b) : String - String
valueOf(char c) : String - String
valueOf(char[] data) : String - String
valueOf(double d) : String - String
valueOf(float f) : String - String
valueOf(int i) : String - String
valueOf(long l) : String - String
valueOf(Object obj) : String - String
valueOf(char[] data, int offset, int count) : String - String
copyValueOf(char[] data) : String - String
Press 'Ctrl+Space' to show Template Proposals
Retu
Para
Retu



```
        Integer v = 100;
        int v1 = v.intValue();
        System.out.println(v1);
```

100

Converted from wrapper to primitive.

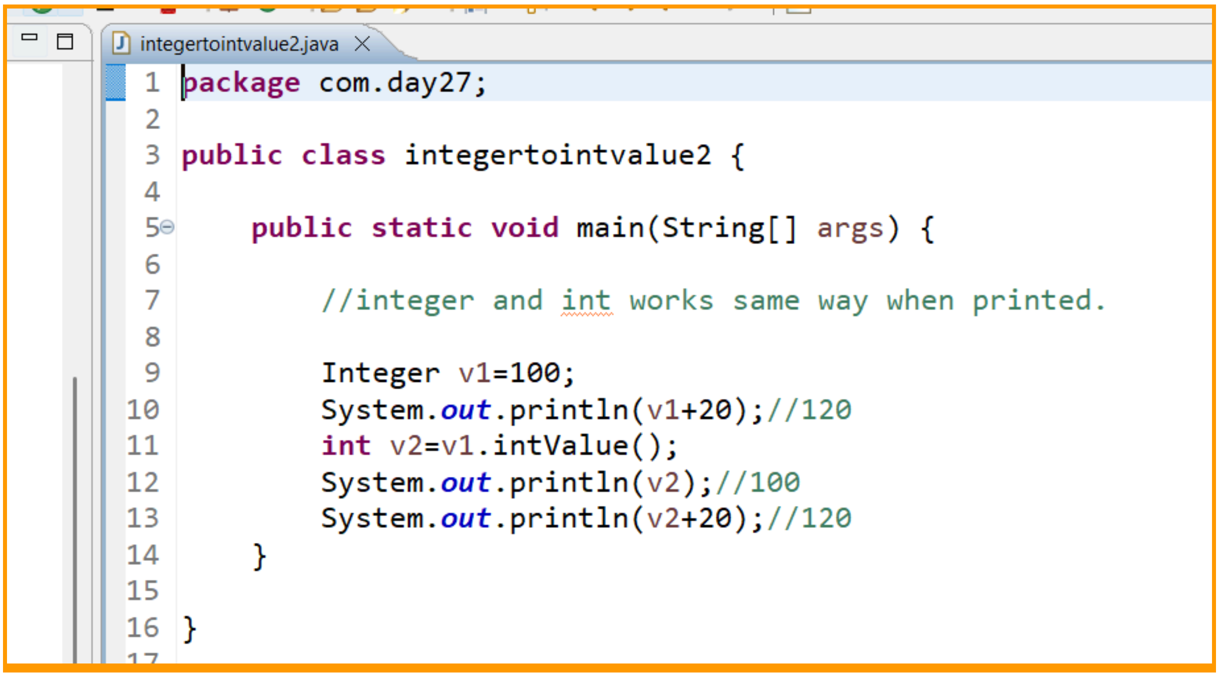


```
        Integer v = 100;
        System.out.println(v+20);//120
        int v1 = v.intValue();
        System.out.println(v1);
```

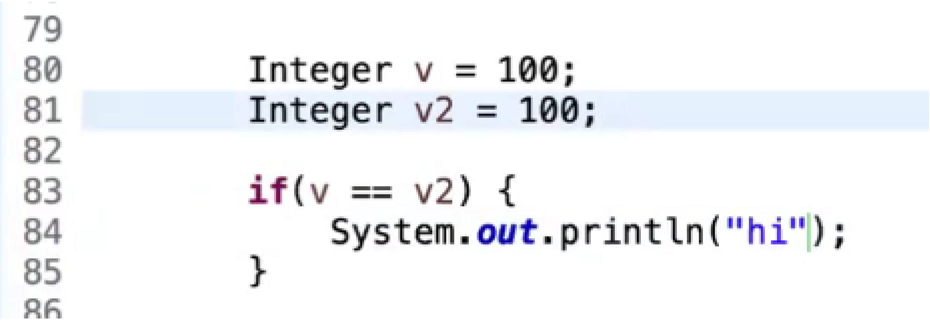
120

100

paste integertointvalue2

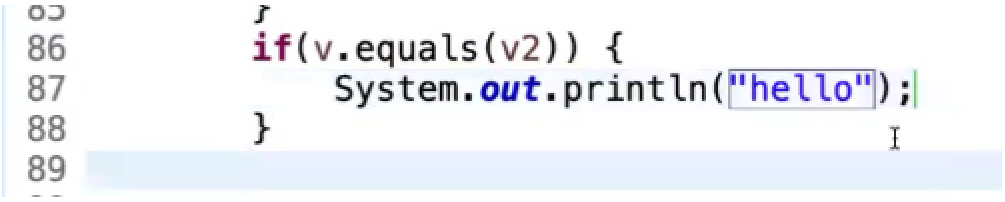


```
package com.day27;

public class integertointvalue2 {

    public static void main(String[] args) {

        //integer and int works same way when printed.

        Integer v1=100;
        System.out.println(v1+20);//120
        int v2=v1.intValue();
        System.out.println(v2);//100
        System.out.println(v2+20);//120
    }

}
```



```
        Integer v = 100;
        Integer v2 = 100;

        if(v == v2) {
            System.out.println("hi");
        }
```

Hi



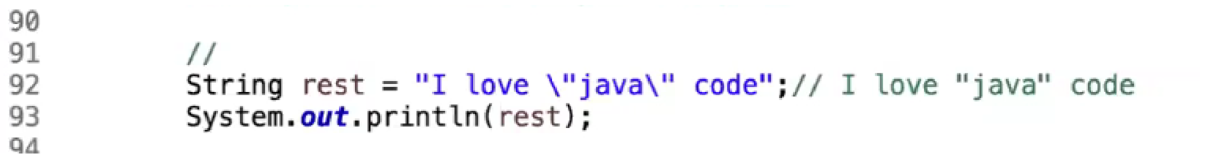
```
        }
        if(v.equals(v2)) {
            System.out.println("hello");
        }
```

Hello

Non primitive compare using **.equals. all wrapper can be compared using ==.**

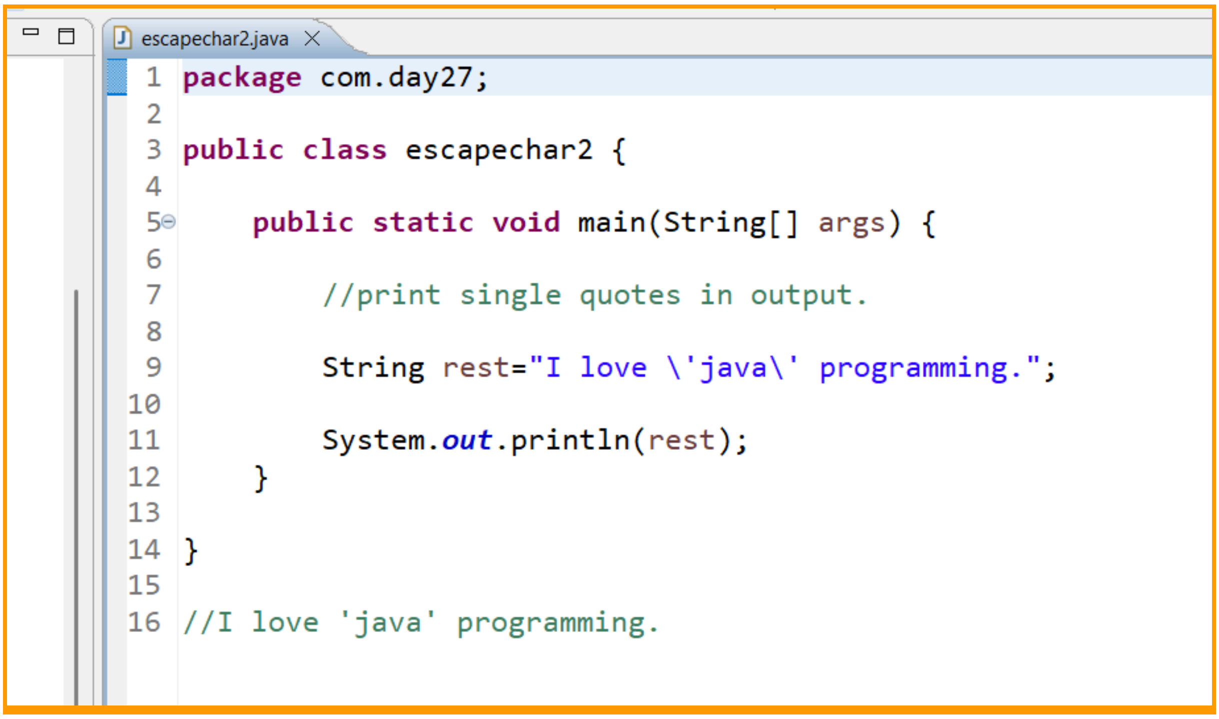
Primitive compare using **==**. 54.00

Escape chars-



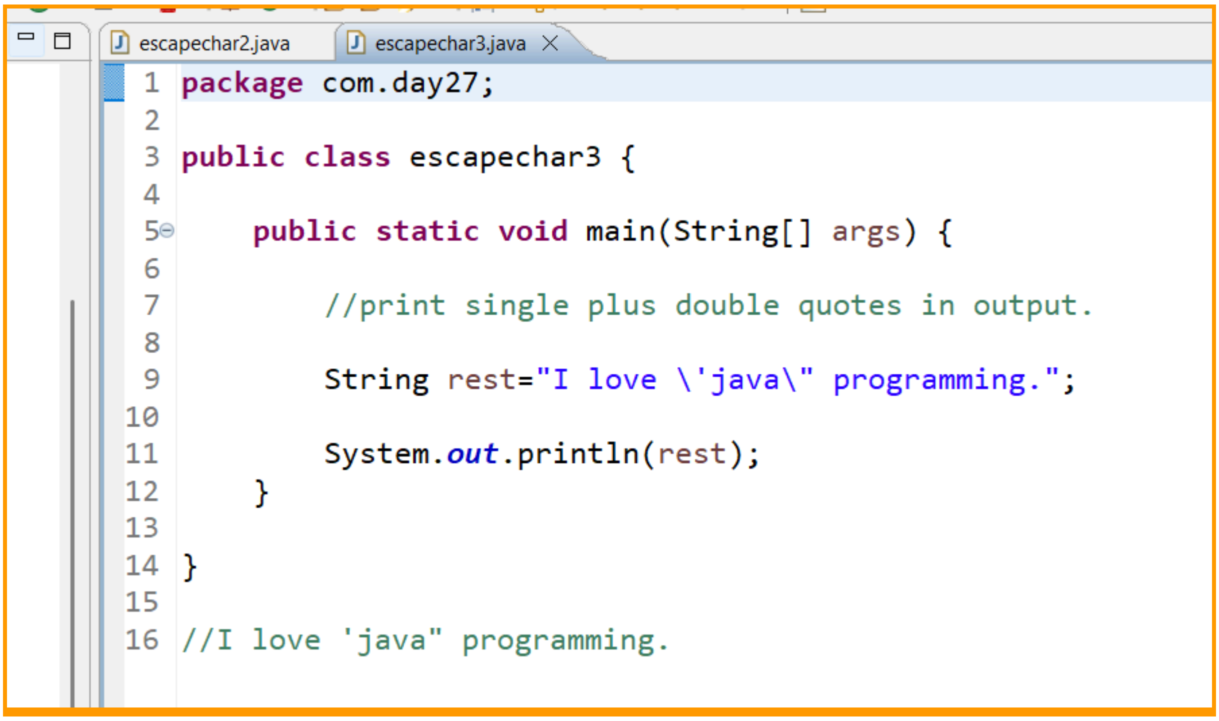
```
        //
        String rest = "I love \"java\" code";// I love "java" code
        System.out.println(rest);
```

paste escapechar2

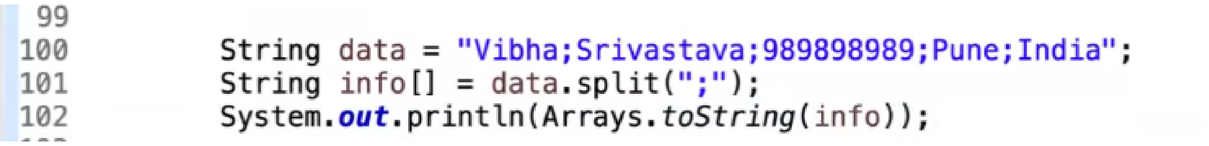


```
package com.day27;

public class escapechar2 {

    public static void main(String[] args) {

        //print single quotes in output.

        String rest="I love \'java\' programming.";

        System.out.println(rest);
    }

}

//I love 'java' programming.
```

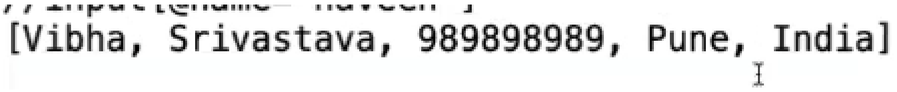
paste escapechar3



```
package com.day27;

public class escapechar3 {

    public static void main(String[] args) {

        //print single plus double quotes in output.

        String rest="I love \'java\" programming.";

        System.out.println(rest);
    }

}

//I love 'java" programming.
```



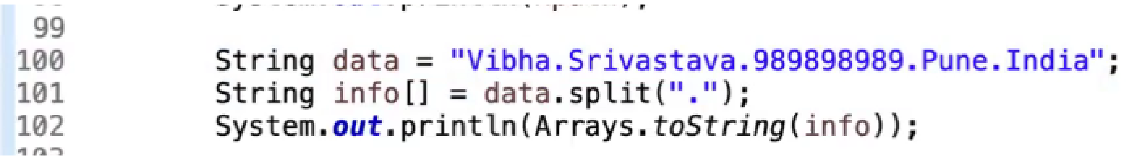
```
        String data = "Vibha;Srivastava;989898989;Pune;India";
        String info[] = data.split(";");
        System.out.println(Arrays.toString(info));
```



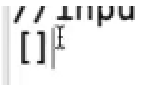
```
[Vibha, Srivastava, 989898989, Pune, India]
```

Split with dot-

no output.

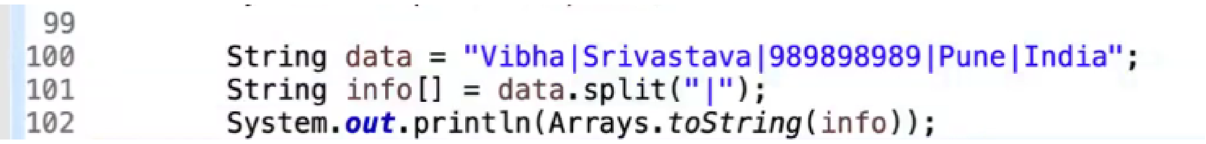


```
        String data = "Vibha.Srivastava.989898989.Pune.India";
        String info[] = data.split(".");
        System.out.println(Arrays.toString(info));
```

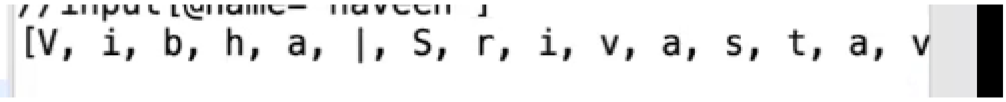


```
//inpu
[]
```

Split with pipe-

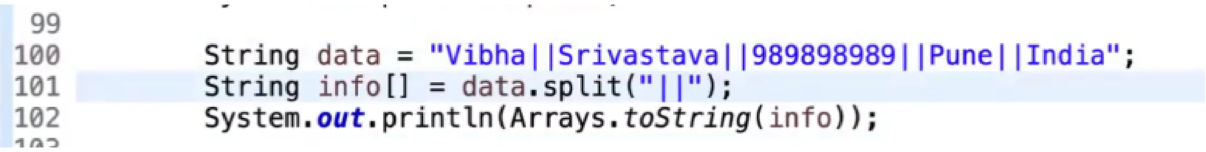


```
        String data = "Vibha|Srivastava|989898989|Pune|India";
        String info[] = data.split("|");
        System.out.println(Arrays.toString(info));
```

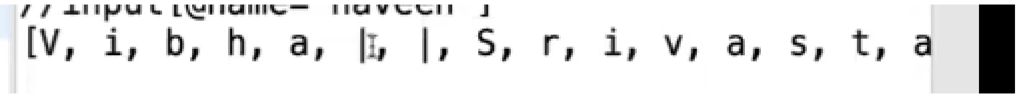


```
[V, i, b, h, a, |, S, r, i, v, a, s, t, a, v
```

[redacted]



```
        String data = "Vibha||Srivastava||989898989||Pune||India";
        String info[] = data.split("||");
        System.out.println(Arrays.toString(info));
```



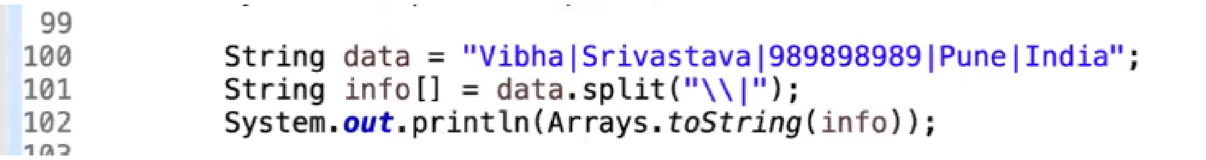
```
[V, i, b, h, a, |, |, S, r, i, v, a, s, t, a
```

[redacted]

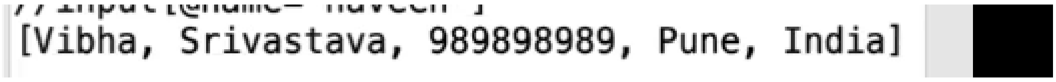
Special char in java not read properly in split.

1.12 - doubt solving time given.

Double escape sequence-

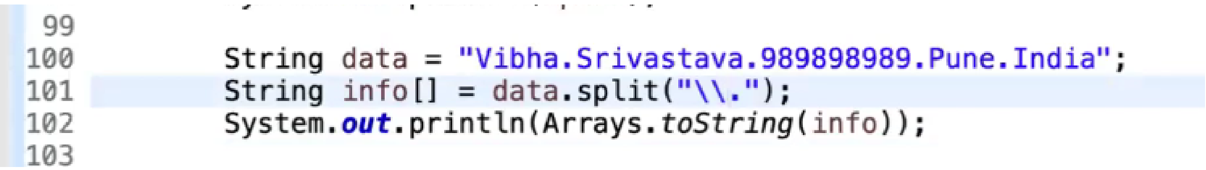


```
        String data = "Vibha|Srivastava|989898989|Pune|India";
        String info[] = data.split("\\|");
        System.out.println(Arrays.toString(info));
```

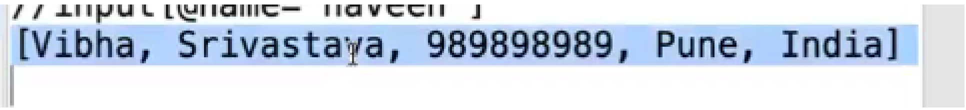


```
[Vibha, Srivastava, 989898989, Pune, India]
```

[redacted]

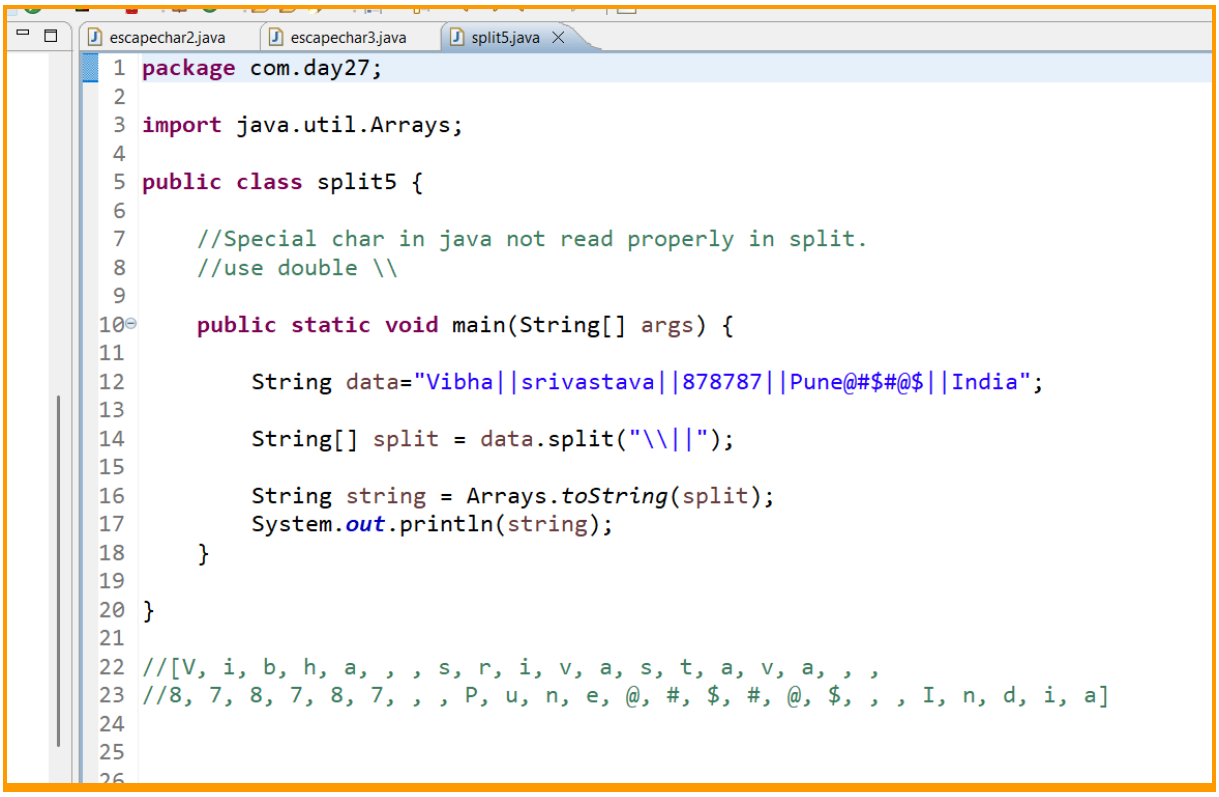


```
        String data = "Vibha.Srivastava.989898989.Pune.India";
        String info[] = data.split("\\.");
        System.out.println(Arrays.toString(info));
```



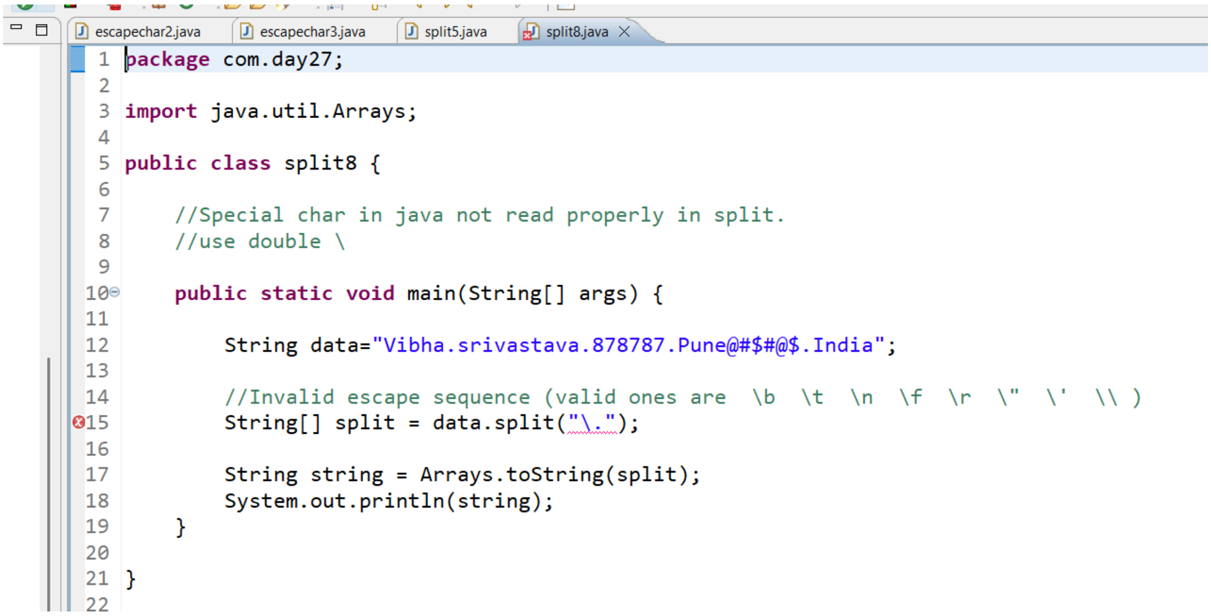
```
[Vibha, Srivastava, 989898989, Pune, India]
```

paste split5



```java
package com.day27;

import java.util.Arrays;

public class split5 {

    //Special char in java not read properly in split.
    //use double \\

    public static void main(String[] args) {

        String data="Vibha||srivastava||878787||Pune@#$#@$||India";

        String[] split = data.split("\\||");

        String string = Arrays.toString(split);
        System.out.println(string);
    }

}

//[V, i, b, h, a, , , s, r, i, v, a, s, t, a, v, a, , ,
//8, 7, 8, 7, 8, 7, , , P, u, n, e, @, #, $, #, @, $, , , I, n, d, i, a]


```

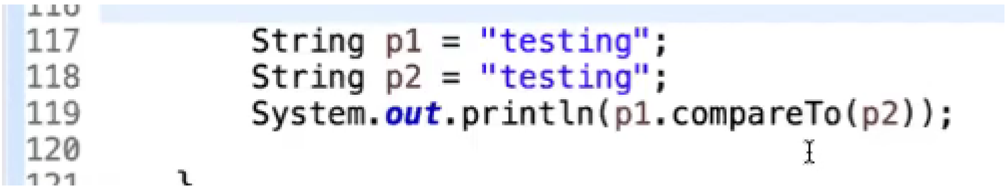
paste split8



```java
package com.day27;

import java.util.Arrays;

public class split8 {

    //Special char in java not read properly in split.
    //use double \

    public static void main(String[] args) {

        String data="Vibha.srivastava.878787.Pune@#$#@$.India";

        //Invalid escape sequence (valid ones are  \b  \t  \n  \f  \r  \"  \'  \\ )
        String[] split = data.split("\.");

        String string = Arrays.toString(split);
        System.out.println(string);
    }

}

```

For non primitive data types the memory is calculated at run time.

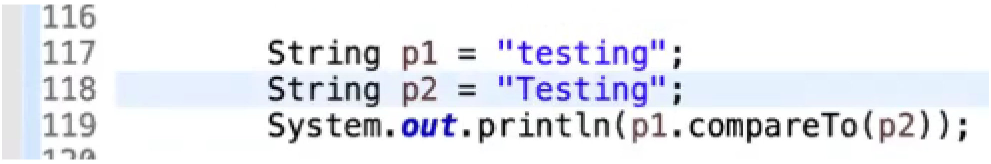
Compare to-



```
        String p1 = "testing";
        String p2 = "testing";
        System.out.println(p1.compareTo(p2));

    }
```

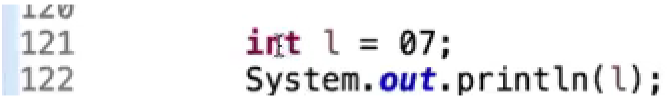
Returns int.

0



```
        String p1 = "testing";
        String p2 = "Testing";
        System.out.println(p1.compareTo(p2));
```

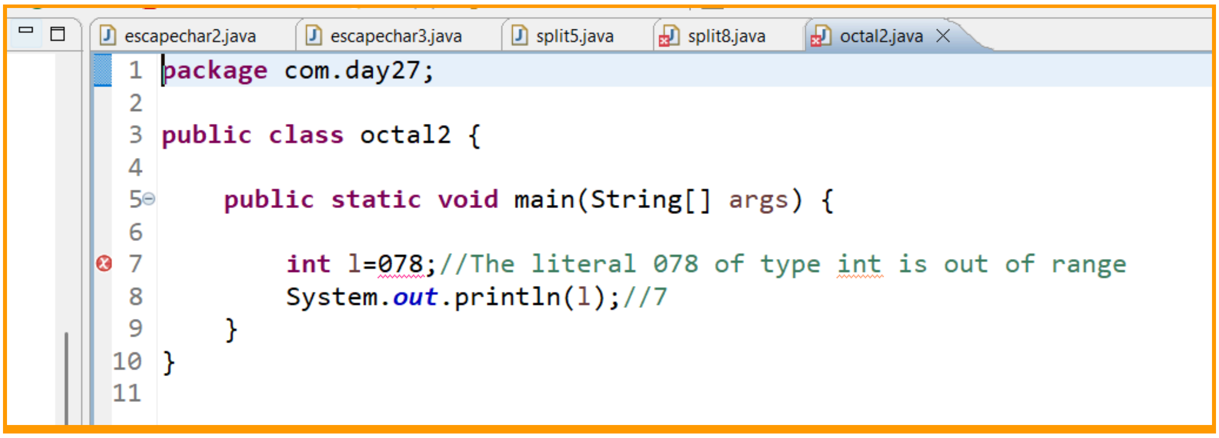
32



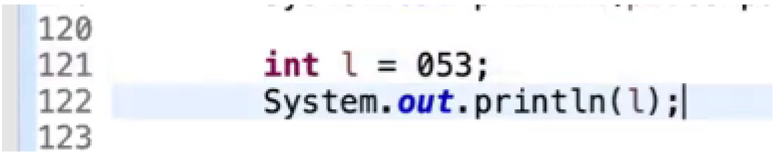
```
        int l = 07;
        System.out.println(l);
```

7

paste octal2



```
package com.day27;

public class octal2 {

    public static void main(String[] args) {

        int l=078;//The literal 078 of type int is out of range
        System.out.println(l);//7
    }
}

```



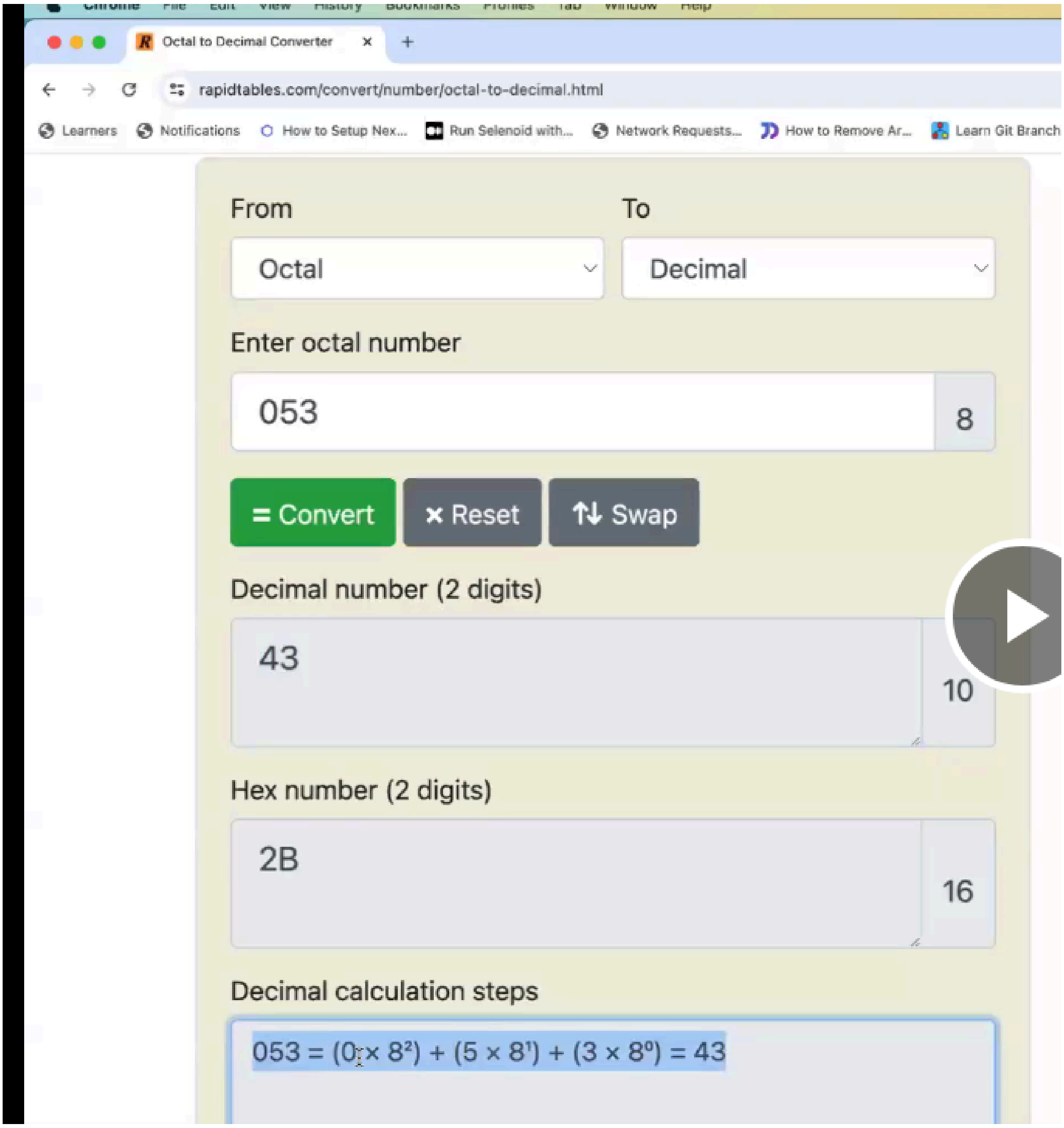
```

        int l = 053;
        System.out.println(l);

```

43

Octal number that is why this difference.

Octal is base 8.

Decimal is base 10.



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Octal to Decimal Converter
rapidtables.com/convert/number/octal-to-decimal.html
Learners
Notifications
How to Setup Nex...
Run Selenoid with...
Network Requests...
How to Remove Ar...
Learn Git Branch
= Convert
✖ Reset
⇅ Swap

From: Octal

To: Decimal

Enter octal number

053 8

Decimal number (2 digits)

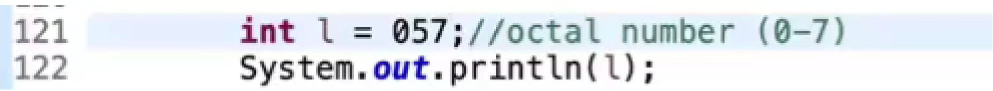
43 10

Hex number (2 digits)

2B 16

Decimal calculation steps

$053 = (0 \times 8^2) + (5 \times 8^1) + (3 \times 8^0) = 43$



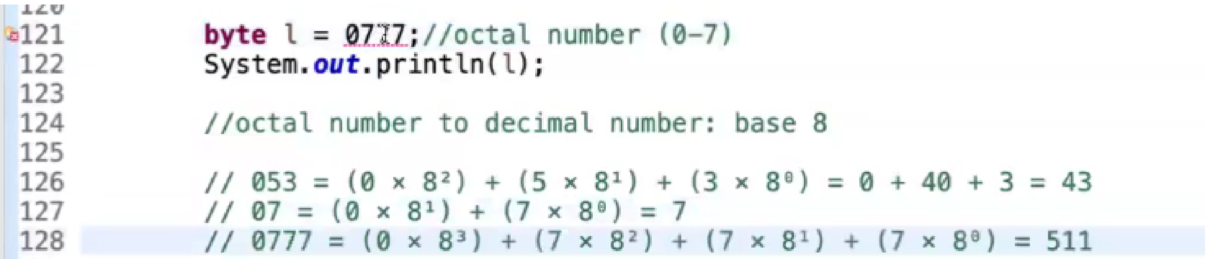
```
        int l = 057;//octal number (0-7)
        System.out.println(l);
```

47

Every digit of octal should be between 0 to 7.

If we write int l=058; we get error because last digit is greater than 7.

Return type should also be able to hold the value-

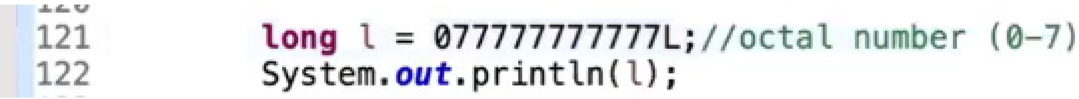


```
        byte l = 0777;//octal number (0-7)
        System.out.println(l);

        //octal number to decimal number: base 8

        // 053 = (0 × 8²) + (5 × 8¹) + (3 × 8⁰) = 0 + 40 + 3 = 43
        // 07 = (0 × 8¹) + (7 × 8⁰) = 7
        // 0777 = (0 × 8³) + (7 × 8²) + (7 × 8¹) + (7 × 8⁰) = 511
```

Byte cannot hold 511.

//Type mismatch: cannot convert from int to byte

Long-



```
        long l = 0777777777777L;//octal number (0-7)
        System.out.println(l);
```

Run and get the value starting with 8……