



```
knownperson@Known-Person:~$ jshell
| Welcome to JShell -- Version 23.0.1
| For an introduction type: /help intro
```

```
jshell> String s = "Mohit Khandale"
s ==> "Mohit Khandale"
```

```
jshell> String t = "2023bit017"
t ==> "2023bit017"
```

```
jshell> s.
charAt(          chars()          codePointAt(          codePointBefore(          codePointCount(          codePoints()
compareTo(      compareToIgnoreCase( concat(          contains(          contentEquals(          describeConstable()
endsWith(        equals(          equalsIgnoreCase(   formatted(          getBytes(          getChars(
getClass()       hashCode()       indent(          indexOf(          intern()          isBlank()
isEmpty()        lastIndexOf(      length()        lines()          matches(          notify()
notifyAll()      offsetByCodePoints( resolveConstantDesc( split(          replace(          replaceAll(
replaceFirst(    stripLeading()     stripTrailing()   subSequence(      startsWith(      strip()
stripIndent()    toString()        toUpperCase()     transform(        substring(        toCharArray()
toLowerCase()    wait(             toUpperCase()     transform(        translateEscapes() trim()

jshell> s.charAt(0)
$3 ==> 'M'
```

```
jshell> s.codePoin
codePointAt(      codePointBefore(  codePointCount(  codePoints()
jshell> s.codePointAt(0)
$4 ==> 77
```

```
jshell> s.codePointBefore(1)
$5 ==> 77
```

```
jshell> s.codePointCount(0,4)
```



```
jshell> s.codePointBefore(1)
```

```
$5 ==> 77
```

```
jshell> s.codePointCount(0,4)
```

```
$6 ==> 4
```

```
jshell> s.compareTo
```

```
compareTo(           compareToIgnoreCase(
```

```
jshell> s.compareTo(t)
```

```
$7 ==> 27
```

```
jshell> t.compareTo(s)
```

```
$8 ==> -27
```

```
jshell> s.compareTo(s)
```

```
$9 ==> 0
```

```
jshell> s.compareToIgnoreCase(t)
```

```
$10 ==> 59
```

```
jshell> s.compareToIgnoreCase(s)
```

```
$11 ==> 0
```

```
jshell> t.compareToIgnoreCase(s)
```

```
$12 ==> -59
```

```
jshell> s.concat(t)
```

```
$13 ==> "Mohit Khandale2023bit017"
```

```
jshell> s.contains("oh")
```

```
$14 ==> true
```

```
jshell> s.contains("ho")
```



```
jshell> s.concat(t)
$13 ==> "Mohit Khandale2023bit017"
```

```
jshell> s.contains("oh")
$14 ==> true
```

```
jshell> s.contains("ho")
$15 ==> false
```

```
jshell> s.contentEquals("Mohit")
$16 ==> false
```

```
jshell> s.contentEquals("Mohit Khandale")
$17 ==> true
```

```
jshell> s.endsWith("dale")
$18 ==> true
```

```
jshell> s.endsWith("Mohit")
$19 ==> false
```

```
jshell> s.equals
equals(           equalsIgnoreCase(
jshell> s.equals
equals(           equalsIgnoreCase(
jshell> s.equals(s)
$20 ==> true
```

```
jshell> s.equals(t)
$21 ==> false
```

```
jshell> String m = "2023BIT017"
m ==> "2023BIT017"
```



```
jshell> String m = "2023BIT017"
```

```
m ==> "2023BIT017"
```

```
jshell> m.equalsIgnoreCase(t)
```

```
$23 ==> true
```

```
jshell> String m = "Mohit, %s"
```

```
m ==> "Mohit, %s"
```

```
jshell> String n = m.formatted("007")
```

```
n ==> "Mohit, 007"
```

```
jshell> s.getBytes()
```

```
$26 ==> byte[14] { 77, 111, 104, 105, 116, 32, 75, 104, 97, 110, 100, 97, 108, 101 }
```

```
jshell> char[] a = {'a','b','c','d','e','f','g'}
```

```
a ==> char[7] { 'a', 'b', 'c', 'd', 'e', 'f', 'g' }
```

```
jshell> s.getC
```

```
getChars(    getClass())
```

```
jshell> s.getChars(0,5,a,0)
```

```
jshell> a
```

```
a ==> char[7] { 'M', 'o', 'h', 'i', 't', 'f', 'g' }
```

```
jshell> s.hashCode()
```

```
$30 ==> 1586916559
```

```
jshell> s.indexOf("Khandale")
```

```
$31 ==> 6
```

```
jshell> String m = "Mohit"
```

```
m ==> "Mohit"
```



```
jshell> String m = "Mohit"
```

```
m ==> "Mohit"
```

```
jshell> String n = new String("Mohit")
```

```
n ==> "Mohit"
```

```
jshell> s==k
```

```
| Error:  
| cannot find symbol  
|   symbol:   variable k  
|   s==k  
|   ^
```

```
jshell> m == n
```

```
$34 ==> false
```

```
jshell> n = n.intern()
```

```
n ==> "Mohit"
```

```
jshell> m == n
```

```
$36 ==> true
```

```
jshell> s.isEmpty()
```

```
$37 ==> false
```

```
jshell> String k = ""
```

```
k ==> ""
```

```
jshell> k.isEmpty()
```

```
$39 ==> true
```

```
jshell> k = "Yo YO"
```

```
k ==> "Yo YO"
```



```
jshell> k = "Yo YO"  
k ==> "Yo YO"
```

```
jshell> k.lastIndexOf("Yo")  
$41 ==> 0
```

```
jshell> k = "Yo Yo Yo"  
k ==> "Yo Yo Yo"
```

```
jshell> k.lastIndexOf("Yo")  
$43 ==> 6
```

```
jshell> s.length()  
$44 ==> 14
```

```
jshell> k = "Mohit|Hello|You"  
k ==> "Mohit|Hello|You"
```

```
jshell> "Mohit".matches(k)  
$46 ==> true
```

```
jshell> s.offsetByCodePoints(3,7)  
$47 ==> 10
```

```
jshell> s.re  
regionMatches(      repeat(      replace(      replaceAll(      replaceFirst(      resolveConstantDesc(  
jshell> s.regionMatches(0,k,5,0)  
$48 ==> true
```

```
jshell> t.replac  
replace(      replaceAll(      replaceFirst(  
jshell> t.replace('2','9')  
$49 ==> "9a93hi+017"
```



```
jshell> t.replac
replace(      replaceAll(      replaceFirst(
jshell> t.replace('2','9')
$49 ==> "9093bit017"
```

```
jshell> s.replaceAll("Khandale", "0007")
$50 ==> "Mohit 0007"
```

```
jshell> m = "abc abc 123"
m ==> "abc abc 123"
```

```
jshell> m.replaceFirst("abc", "xyz")
$52 ==> "xyz abc 123"
```

```
jshell> s.split(" ")
$53 ==> String[2] { "Mohit", "Khandale" }
```

```
jshell> s.split("a")
$54 ==> String[3] { "Mohit Kh", "nd", "le" }
```

```
jshell> t.startsWith("2023")
$55 ==> true
```

```
jshell> t.subSequence(2,8)
$56 ==> "23bit0"
```

```
jshell> t.substring(4)
$57 ==> "bit017"
```

```
jshell> t.substring(4,6)
$58 ==> "bi"
```

```
jshell> t.toCharArray()
```




```
jshell> t.substring(4)
```

```
$57 ==> "bit017"
```

```
jshell> t.substring(4,6)
```

```
$58 ==> "bi"
```

```
jshell> t.toCharArray()
```

```
$59 ==> char[10] { '2', '0', '2', '3', 'b', 'i', 't', '0', '1', '7' }
```

```
jshell> k = "MOHIT"
```

```
k ==> "MOHIT"
```

```
jshell> k.toLowerCase()
```

```
$61 ==> "mohit"
```

```
jshell> t.toString()
```

```
$62 ==> "2023bit017"
```

```
jshell> t
```

```
t ==> "2023bit017"
```

```
jshell> t.toUpperCase()
```

```
$64 ==> "2023BIT017"
```

```
jshell> k = "    abcd    abce    "
```

```
k ==> "    abcd    abce    "
```

```
jshell> k.tr
```

```
transform(          translateEscapes()  trim())
```

```
jshell> k.trim()
```

```
$66 ==> "abcd    abce"
```

```
jshell>
```




```
39 : k.isEmpty()
40 : k = "Yo Yo"
41 : k.lastIndexOf("Yo")
42 : k = "Yo Yo Yo"
43 : k.lastIndexOf("Yo")
44 : s.length()
45 : k = "Mohit|Hello|You"
46 : "Mohit".matches(k)
47 : s.offsetByCodePoints(3,7)
48 : s.regionMatches(0,k,5,0)
49 : t.replace('2','9')
50 : s.replaceAll("Khandale", "0007")
51 : m = "abc abc 123"
52 : m.replaceFirst("abc", "xyz")
53 : s.split(" ")
54 : s.split("a")
55 : t.startsWith("2023")
56 : t.subSequence(2,8)
57 : t.substring(4)
58 : t.substring(4,6)
59 : t.toCharArray()
60 : k = "MOHIT"
61 : k.toLowerCase()
62 : t.toString()
63 : t
64 : t.toUpperCase()
65 : k = "    abcd    abce    "
66 : k.trim()
```

```
jshell> /edit t
t ==> "2023BIT017"
```

```
jshell>
```



```
jshell> k.trim()  
$66 ==> "abcd    abce"
```

```
jshell> /help
```

```
| Type a Java language expression, statement, or declaration.  
| Or type one of the following commands:  
| /list [<name or id>|-all|-start]  
|     list the source you have typed  
| /edit <name or id>  
|     edit a source entry  
| /drop <name or id>  
|     delete a source entry  
| /save [<-all>|-history>|-start] <file>  
|     Save snippet source to a file  
| /open <file>  
|     open a file as source input  
| /vars [<name or id>|-all>|-start]  
|     list the declared variables and their values  
| /methods [<name or id>|-all>|-start]  
|     list the declared methods and their signatures  
| /types [<name or id>|-all>|-start]  
|     list the type declarations  
| /imports  
|     list the imported items  
| /exit [<integer-expression-snippet>]  
|     exit the jshell tool  
| /env [-class-path <path>] [-module-path <path>] [-add-modules <modules>] ...  
|     view or change the evaluation context  
| /reset [-class-path <path>] [-module-path <path>] [-add-modules <modules>]...  
|     reset the jshell tool  
| /reload [-restore] [-quiet] [-class-path <path>] [-module-path <path>]...  
|     reset and replay relevant history -- current or previous (-restore)  
| /history [-all]
```



```
reset and replay relevant history -- current or previous (-restore)
```

```
/history [-all]
```

```
history of what you have typed
```

```
/help [<command>|<subject>]
```

```
get information about using the jshell tool
```

```
/set editor|start|feedback|mode|prompt|truncation|format ...
```

```
set configuration information
```

```
/? [<command>|<subject>]
```

```
get information about using the jshell tool
```

```
#!
```

```
rerun last snippet -- see /help rerun
```

```
/<id>
```

```
rerun snippets by ID or ID range -- see /help rerun
```

```
/-<n>
```

```
rerun n-th previous snippet -- see /help rerun
```

For more information type '/help' followed by the name of a command or a subject.

For example '/help /list' or '/help intro'.

Subjects:

intro

an introduction to the jshell tool

keys

a description of readline-like input editing

id

a description of snippet IDs and how use them

shortcuts

a description of keystrokes for snippet and command completion, information access, and automatic code generation

context

a description of the evaluation context options for /env /reload and /reset



```
context
  a description of the evaluation context options for /env /reload and /reset
rerun
  a description of ways to re-evaluate previously entered snippets
```

```
jshell> /vars
```

```
String s = "Mohit Khandale"
String t = "2023bit017"
char $3 = 'M'
int $4 = 77
int $5 = 77
int $6 = 4
int $7 = 27
int $8 = -27
int $9 = 0
int $10 = 59
int $11 = 0
int $12 = -59
String $13 = "Mohit Khandale2023bit017"
boolean $14 = true
boolean $15 = false
boolean $16 = false
boolean $17 = true
boolean $18 = true
boolean $19 = false
boolean $20 = true
boolean $21 = false
boolean $23 = true
byte[] $26 = byte[14] { 77, 111, 104, 105, 116, 32, 75, 104, 97, 110, 100, 97, 108, 101 }
char[] a = char[7] { 'M', 'o', 'h', 'i', 't', 'f', 'g' }
int $30 = 1586916559
int $31 = 6
String m = "abc abc 123"
String n = "Mohit"
```



```
String m = "abc abc 123"  
String n = "Mohit"  
boolean $34 = false  
boolean $36 = true  
boolean $37 = false  
String k = "    abcd    abce    "  
boolean $39 = true  
int $41 = 0  
int $43 = 6  
int $44 = 14  
boolean $46 = true  
int $47 = 10  
boolean $48 = true  
String $49 = "9093bit017"  
String $50 = "Mohit 0007"  
String $52 = "xyz abc 123"  
String[] $53 = String[2] { "Mohit", "Khandale" }  
String[] $54 = String[3] { "Mohit Kh", "nd", "le" }  
boolean $55 = true  
CharSequence $56 = "23bit0"  
String $57 = "bit017"  
String $58 = "bi"  
char[] $59 = char[10] { '2', '0', '2', '3', 'b', 'i', 't', '0', '1', '7' }  
String $61 = "mohit"  
String $62 = "2023bit017"  
String $64 = "2023BIT017"  
String $66 = "abcd    abce"
```

jshell> /methods

jshell> /list

```
1 : String s = "Mohit Khandale";
```




```
1 : String s = "Mohit Khandale";
2 : String t = "2023bit017";
3 : s.charAt(0)
4 : s.codePointAt(0)
5 : s.codePointBefore(1)
6 : s.codePointCount(0,4)
7 : s.compareTo(t)
8 : t.compareTo(s)
9 : s.compareTo(s)
10 : s.compareToIgnoreCase(t)
11 : s.compareToIgnoreCase(s)
12 : t.compareToIgnoreCase(s)
13 : s.concat(t)
14 : s.contains("oh")
15 : s.contains("ho")
16 : s.contentEquals("Mohit")
17 : s.contentEquals("Mohit Khandale")
18 : s.endsWith("dale")
19 : s.endsWith("Mohit")
20 : s.equals(s)
21 : s.equals(t)
22 : m.equalsIgnoreCase(t)
26 : s.getBytes()
27 : char[] a = {'a','b','c','d','e','f','g'};
28 : s.getChars(0,5,a,0)
29 : a
30 : s.hashCode()
31 : s.indexOf("Khandale")
32 : String m = "Mohit";
33 : String n = new String("Mohit");
34 : m == n
35 : n = n.intern()
36 : m == n
```



```
35 : n = n.intern()
36 : m == n
37 : s.isEmpty()
38 : String k = "";
39 : k.isEmpty()
40 : k = "Yo Yo"
41 : k.lastIndexOf("Yo")
42 : k = "Yo Yo Yo"
43 : k.lastIndexOf("Yo")
44 : s.length()
45 : k = "Mohit|Hello|You"
46 : "Mohit".matches(k)
47 : s.offsetByCodePoints(3,7)
48 : s.regionMatches(0,k,5,0)
49 : t.replace('2','9')
50 : s.replaceAll("Khandale", "0007")
51 : m = "abc abc 123"
52 : m.replaceFirst("abc", "xyz")
53 : s.split(" ")
54 : s.split("a")
55 : t.startsWith("2023")
56 : t.subSequence(2,8)
57 : t.substring(4)
58 : t.substring(4,6)
59 : t.toCharArray()
60 : k = "MOHIT"
61 : k.toLowerCase()
62 : t.toString()
63 : t
64 : t.toUpperCase()
65 : k = "    abcd    abce    "
66 : k.trim()
```




```
39 : k.isEmpty()
40 : k = "Yo Yo"
41 : k.lastIndexOf("Yo")
42 : k = "Yo Yo Yo"
43 : k.lastIndexOf("Yo")
44 : s.length()
45 : k = "Mohit|Hello|You"
46 : "Mohit".matches(k)
47 : s.offsetByCodePoints(3,7)
48 : s.regionMatches(0,k,5,0)
49 : t.replace('2','9')
50 : s.replaceAll("Khandale", "0007")
51 : m = "abc abc 123"
52 : m.replaceFirst("abc", "xyz")
53 : s.split(" ")
54 : s.split("a")
55 : t.startsWith("2023")
56 : t.subSequence(2,8)
57 : t.substring(4)
58 : t.substring(4,6)
59 : t.toCharArray()
60 : k = "MOHIT"
61 : k.toLowerCase()
62 : t.toString()
63 : t
64 : t.toUpperCase()
65 : k = "    abcd    abce    "
66 : k.trim()
```

```
jshell> /edit t
t ==> "2023BIT017"
```

```
jshell> 
```

```
37 : s.isEmpty()
38 : String k = "";
39 : k.isEmpty()
40 : k = "Yo YO"
41 : k.lastIndexOf("Yo")
42 : k = "Yo Yo Yo"
43 : k.lastIndexOf("Yo")
44 : s.length()
45 : k = "Mohit|Hello|You"
46 : "Mohit".matches(k)
47 : s.offsetByCodePoints(3,7)
48 : s.regionMatches(0,k,5,0)
49 : t.replace('2','9')
50 : s.replaceAll("Khandale", "0007")
51 : m = "abc abc 123"
52 : m.replaceFirst("abc", "xyz")
53 : s.split(" ")
54 : s.split("a")
55 : t.startsWith("2023")
56 : t.subSequence(2,8)
57 : t.substring(4)
58 : t.substring(4,6)
59 : t.toCharArray()
60 : k = "MOHIT"
61 : k.toLowerCase()
62 : t.toString()
63 : t
64 : t.toUpperCase()
65 : k = "   abcd   abce   "
66 : k.trim()
```

jshell> /edit t

```
String t = "2023bit017";
```

```
jshell> /edit t
```

Cancel

Accept

Exit