

ActivitiesTerminalJan 22 23:48

Screenshot captured
You can paste the image from the clipboard.

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
$10 $11 $2 $4 $5 $6

Signatures:
void String.getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)

<press tab again to see documentation>
jshell> str.getChars(3)
|
| Error:
| Method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: int
|   reason: actual and formal argument lists differ in length
| str.getChars(3)
| ^^^^^^^^^^
|
jshell> str.getChars(6)
|
| Error:
| Method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: int
|   reason: actual and formal argument lists differ in length
| str.getChars(6)
| ^^^^^^^^^^
|
jshell> str.indexOf(a)
|
| Error:
| cannot find symbol
|   symbol:   variable a
| str.indexOf(a)
| ^
|
jshell> str.indexOf(
$10 $11 $12 $2 $4 $5 $6 s str

Signatures:
int String.indexOf(int ch)
int String.indexOf(int ch, int fromIndex)
int String.indexOf(String str)
int String.indexOf(String str, int fromIndex)

<press tab again to see documentation>
jshell> str.indexOf(3)
$16 ==> -1

jshell> str.intern()
Signatures:
String String.intern()

<press tab again to see documentation>
jshell> str.intern()
$17 ==> "shravani"

jshell>
```

ActivitiesTerminalJan 22 23:48

Screenshot captured
You can paste the image from the clipboard.

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
$10 $11 $2 $4 $5 $6

Signatures:
void String.getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)

<press tab again to see documentation>
jshell> str.getChars(3)
|
| Error:
| Method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: int
|   reason: actual and formal argument lists differ in length
| str.getChars(3)
| ^^^^^^^^^^
|
jshell> str.getChars(6)
|
| Error:
| Method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: int
|   reason: actual and formal argument lists differ in length
| str.getChars(6)
| ^^^^^^^^^^
|
jshell> str.indexOf(a)
|
| Error:
| cannot find symbol
|   symbol:   variable a
| str.indexOf(a)
| ^
|
jshell> str.indexOf(
$10 $11 $12 $2 $4 $5 $6 s str

Signatures:
int String.indexOf(int ch)
int String.indexOf(int ch, int fromIndex)
int String.indexOf(String str)
int String.indexOf(String str, int fromIndex)

<press tab again to see documentation>
jshell> str.indexOf(3)
$16 ==> -1

jshell> str.intern()
Signatures:
String String.intern()

<press tab again to see documentation>
jshell> str.intern()
$17 ==> "shravani"

jshell>
```

Screenshot captured
You can paste the image from the clipboard.

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
Signatures:
boolean String.endsWith(String suffix)

<press tab again to see documentation>
jshell> str.endsWith(l)
| Error:
| cannot find symbol
|   symbol:   variable l
| str.endsWith(l)
|           ^
jshell> str.endsWith(s)
$13 ==> false

jshell> str.equals(
$10  $11  $12  $13  $2  $3  $4  $5  $6  $8  $9  s  str

Signatures:
boolean String.equals(Object anObject)

<press tab again to see documentation>
jshell> str.equals(1'
<press tab again to see all possible completions; total possible completions: 552>
jshell> str.equals(5)
$14 ==> false

jshell> str.getBytes()
$15 ==> byte[8] { 115, 104, 114, 97, 118, 97, 110, 105 }

jshell> str.getChars()
| Error:
| method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: no arguments
|   reason: actual and formal argument lists differ in length
| str.getChars()
| ^^^^^^^^^^
jshell> str.getChars(
$10  $11  $2  $4  $5  $6

Signatures:
void String.getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)

<press tab again to see documentation>
jshell> str.getChars(3)
| Error:
| method getChars in class java.lang.String cannot be applied to given types;
|   required: int,int,char[],int
|   found: int
|   reason: actual and formal argument lists differ in length
| str.getChars(3)
| ^^^^^^^^^^
```



Screenshot captured

You can paste the image from the clipboard.



shravani@shravani-HP-Laptop-15s-eq2xxx: ~

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
jshell> str.codePointAt(3)
$4 ==> 97
```

```
jshell> str.codePointBefore(2)
$5 ==> 104
```

```
jshell> str.codePointCount(3)
| Error:
| method codePointCount in class java.lang.String cannot be applied to given types;
|   required: int,int
|   found:    int
|   reason:   actual and formal argument lists differ in length
| str.codePointCount(3)
| ^^^^^^^^^^^^^^
```

```
jshell> str.compareTo(str)
$6 ==> 0
```

```
jshell>
```

```
jshell> string s="sggs"
| Error:
| cannot find symbol
|   symbol:   class string
|   string s="sggs";
| ^^^^^
```

```
jshell> String s="sggs";
s ==> "sggs"
```

```
jshell> str.compareto(s);
| Error:
| cannot find symbol
|   symbol:   method compareto(java.lang.String)
|   str.compareto(s);
| ^^^^^^^^^
```

```
jshell> s.getClass
| Error:
| cannot find symbol
|   symbol:   variable getClass
|   s.getClass
| ^^^^^^^
```

```
jshell> str.getClass()
$8 ==> class java.lang.String
```

```
jshell> s.getClass()
$9 ==> class java.lang.String
```

```
jshell> str.compareTo(s)
$10 ==> 1
```

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



```
jshell> str.getClass()
$15 ==> class java.lang.String

jshell> exit
| Error:
| cannot find symbol
| symbol:   variable exit
| exit
| ^...^

jshell> /exit
| Goodbye
shravani@shravani-HP-Laptop-15s-eq2xxx:~$ jshell
| Welcome to JShell -- Version 11.0.21
| For an introduction type: /help intro

jshell> String str="shravani"
str ==> "shravani"

jshell> str.
charAt(      chars()      codePointAt(      codePointBefore(      codePointCount(      codePoints()      compareTo(      compareToIgnoreCase(
concat(      contains(      contentEquals(      endsWith(      equals(      equalsIgnoreCase(      getBytes(      getChars(
getClass()   hashCode()      indexOf(      intern()      isEmpty()      lastIndexOf(      length()
lines()      matches(      notify()      notifyAll()      offsetByCodePoints(      regionMatches(      repeat(      replace(
replaceAll(  replaceFirst(  split(      startWith(      strip()      stripLeading()      stripTrailing()      subSequence(
substring(   toCharArray()   toLowerCase(   toString()      toUpperCase(   trim()      wait(

jshell> str.
charAt(      chars()      codePointAt(      codePointBefore(      codePointCount(      codePoints()      compareTo(      compareToIgnoreCase(
concat(      contains(      contentEquals(      endsWith(      equals(      equalsIgnoreCase(      getBytes(      getChars(
getClass()   hashCode()      indexOf(      intern()      isEmpty()      lastIndexOf(      length()
lines()      matches(      notify()      notifyAll()      offsetByCodePoints(      regionMatches(      repeat(      replace(
replaceAll(  replaceFirst(  split(      startWith(      strip()      stripLeading()      stripTrailing()      subSequence(
substring(   toCharArray()   toLowerCase(   toString()      toUpperCase(   trim()      wait(
jshell> str.charAt(3)
$2 ==> 'a'

jshell> str.chars(

jshell> str.chars()
$3 ==> java.util.stream.IntPipeline$Head@5ef04b5

jshell> str.codePointAt()
| Error:
| method codePointAt in class java.lang.String cannot be applied to given types;
| required: int
| found: no arguments
| reason: actual and formal argument lists differ in length
| str.codePointAt()
| ^.....^

jshell> str.codePointAt(3)
$4 ==> 97
```

```
jshell> str.strip()
```

```
shravani@shravani-HP-Laptop-15s-eq2xxx:~$ javap
Usage: javap <options> <classes>
where possible options include:
-? -h --help -help      Print this help message
-version                Version information
-v -verbose             Print additional information
-l                     Print line number and local variable tables
-public                Show only public classes and members
-protected            Show protected/public classes and members
-package              Show package/protected/public classes
                     and members (default)
-p -private            Show all classes and members
-c                    Disassemble the code
-s                    Print internal type signatures
-sysinfo               Show system info (path, size, date, MD5 hash)
                     of class being processed
-constants             Show final constants
--module <module>, -m <module> Specify module containing classes to be disassembled
--module-path <path>      Specify where to find application modules
--system <jdk>           Specify where to find system modules
--class-path <path>       Specify where to find user class files
--classpath <path>        Specify where to find user class files
-cp <path>              Specify where to find user class files
-bootclasspath <path>     Override location of bootstrap class files

GNU-style options may use = instead of whitespace to separate the name of an option
from its value.

Each class to be shown may be specified by a filename, a URL, or by its fully
qualified class name. Examples:
  path/to/MyClass.class
  jar:file:///path/to/MyJar.jar!/mypkg/MyClass.class
  java.lang.Object

shravani@shravani-HP-Laptop-15s-eq2xxx:~$ javap java.lang.String
Compiled from "String.java"
public final class java.lang.String implements java.io.Serializable, java.lang.Comparable<java.lang.String>, java.lang.CharSequence {
    static final boolean COMPACT_STRINGS;
    public static final java.util.Comparator<java.lang.String> CASE_INSENSITIVE_ORDER;
    static final byte LATIN1;
    static final byte UTF16;
    public java.lang.String();
    public java.lang.String(java.lang.String);
    public java.lang.String(char[]);
    public java.lang.String(char[], int, int);
    public java.lang.String(int[], int, int);
    public java.lang.String(byte[], int, int, int);
    public java.lang.String(byte[], int);
    public java.lang.String(byte[], int, int, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.lang.String(byte[], int, int, java.nio.charset.Charset);
    public java.lang.String(byte[], java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.lang.String(byte[], java.nio.charset.Charset);
    public java.lang.String(byte[], int, int);
    public java.lang.String(byte[]);
```


shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
/edit -start
    Edit the startup snippets. Any changes are in this session, and do not
    affect the startup setting

/edit -all
    Edit all snippets including failed, overwritten, dropped, and startup

/edit
    Edit the currently active snippets of code that you typed or read with /open

jsshell> /ed v
| No such snippet: v
```

```
jshell> /open
.bash_history      .bash_logout      .bashrc
.cache/            .config/           .gitconfig
.gnupg/            .java/             .lessht
.local/            .profile           .ssh/
.sudo_as_admin_successful /                  Desktop/
Documents/         Downloads/         Music/
Pictures/          Public/           Templates/
Videos/            VirtualBox\ VMs/  file
myCommands.js      shrvani/
startup.js         workspace/
snap/
```

```
<press tab again to see synopsis>
jshell> /open d
open a file as source input

<press tab again to see full documentation>
jshell> /open d
| File 'd' for '/open' is not found.
```

```
jshell> /l -a

s1 : import java.io.*;
s2 : import java.math.*;
s3 : import java.net.*;
s4 : import java.nio.file.*;
s5 : import java.util.*;
s6 : import java.util.concurrent.*;
s7 : import java.util.function.*;
s8 : import java.util.prefs.*;
s9 : import java.util.regex.*;
s10 : import java.util.stream.*;
e1 : list command gives the entered data before we reset the terminal;
```

```
jshell> /se fe v
| Feedback mode: verbose
```

```
jshell>
```




```
/list <id>
    List the snippet with the specified snippet ID.
    One or more IDs or ID ranges may used, see '/help id'
jshell> /ed v
edit a source entry

<press tab again to see full documentation>
jshell> /ed v
Edit a snippet or snippets of source in an external editor.
The editor to use is set with /set editor. If no editor is set, then the
following environment variables are checked in order: JSHELLEDITOR, VISUAL,
and EDITOR. If no editor has been set and none of the editor environment
variables is set, a simple editor will be launched.

/edit <name>
    Edit the snippet or snippets with the specified name (preference for active snippets)

/edit <id>
    Edit the snippet with the specified snippet ID.
    One or more IDs or ID ranges may used, see '/help id'

/edit -start
    Edit the startup snippets. Any changes are in this session, and do not
    affect the startup setting

/edit -all
    Edit all snippets including failed, overwritten, dropped, and startup

/edit
    Edit the currently active snippets of code that you typed or read with /open
jshell> /ed v
| No such snippet: v

jshell> /open
.bash_history      .bash_logout      .bashrc
.cache/            .config/           .gitconfig
.gnupg/            .java/             .lessht
.local/            .profile           .ssh/
.sudo_as_admin_successful /                  Desktop/
Documents/         Downloads/         Music/
Pictures/          Public/           Templates/
Videos/            VirtualBox\ VMs/  file
myCommands.jsh     shravani/         snap/
startup.jsh        workspace/

<press tab again to see synopsis>
jshell> /open d
open a file as source input

<press tab again to see full documentation>
jshell> /open d
| File 'd' for '/open' is not found.

jshell> /
```

```

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> /help /vars

                /vars
                =====

List the type, name, and value of variables that were entered.

/vars
  List the type, name, and value of the current active variables

/vars <name>
  List variables with the specified name (preference for active variables)

/vars <id>
  List the variable with the specified snippet ID.
  One or more IDs or ID ranges may used, see '/help id'

/vars -start
  List the variables in the evaluated startup snippets

/vars -all
  List all variables including failed, overwritten, dropped, and startup

jshell> /list
-all  -history  -start  e1      s1      s10     s2      s3      s4
s5      s6      s7      s8      s9

<press tab again to see synopsis>
jshell> /list -
-all      -history  -start

<press tab again to see synopsis>
jshell> /list -
list the source you have typed

<press tab again to see full documentation>
jshell> /list -
-all      -history  -start

<press tab again to see synopsis>
jshell> /list -
list the source you have typed

<press tab again to see full documentation>
jshell> /list -
Show the snippets, prefaced with their snippet IDs.

/list

```



ActivitiesTerminalJan 22 16:20Screenshot captured
You can paste the image from the clipboard.

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

shravani@shravani-HP-Laptop-15s-eq2xxx: ~

```
/? [<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
rerun
  a description of ways to re-evaluate previously entered snippets

jshell> /help /vars

                /vars
                =====

List the type, name, and value of variables that were entered.

/vars
  List the type, name, and value of the current active variables

/vars <name>
  List variables with the specified name (preference for active variables)

/vars <id>
  List the variable with the specified snippet ID.
  One or more IDs or ID ranges may used, see '/help id'

/vars -start
  List the variables in the evaluated startup snippets

/vars -all
  List all variables including failed, overwritten, dropped, and startup

jshell>
```

Screenshot captured
You can paste the image from the clipboard.



```
shravani@shravani-HP-Laptop-15s-eq2xxx: ~
/? [<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
rerun
  a description of ways to re-evaluate previously entered snippets

jshell> /help /vars

                /vars
                =====

List the type, name, and value of variables that were entered.

/vars
  List the type, name, and value of the current active variables

/vars <name>
  List variables with the specified name (preference for active variables)

/vars <id>
  List the variable with the specified snippet ID.
  One or more IDs or ID ranges may used, see '/help id'

/vars -start
  List the variables in the evaluated startup snippets

/vars -all
  List all variables including failed, overwritten, dropped, and startup

jshell>
```



```
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]
    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
```

```
jshell> list command gives the entered data before we reset the terminal
Error:
';' expected
list command gives the entered data before we reset the terminal;
^

jshell> /list -all

s1 : import java.io.*;
s2 : import java.math.*;
s3 : import java.net.*;
s4 : import java.nio.file.*;
s5 : import java.util.*;
s6 : import java.util.concurrent.*;
s7 : import java.util.function.*;
s8 : import java.util.prefs.*;
s9 : import java.util.regex.*;
s10 : import java.util.stream.*;
e1 : list command gives the entered data before we reset the terminal;

jshell> /
/!      /?      /drop    /edit    /env      /exit
/help   /history /imports /list    /methods /open
/reload /reset   /save    /set     /types   /vars

<press tab again to see synopsis>
jshell> /open d
open a file as source input

<press tab again to see full documentation>
jshell> /open d /l
open a file as source input

<press tab again to see full documentation>
jshell> /open d /l/methods
File 'd /l/methods' for '/open' is not found.

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]
```

```
shravani@shravani-HP-Laptop-15s-eq2xxx: ~$ jshell
| Welcome to JShell -- Version 11.0.21
| For an introduction type: /help intro

jshell> /vars

jshell> /oprn myCommands.jsh
| Invalid command: /oprn
| Type /help for help.

jshell> /oprn myCommands.jsh
| Invalid command: /oprn
| Type /help for help.

jshell> /open myCommands.jsh

jshell> /vars
| String str = "hello"
| int i = 10

jshell> to create file
| Error:
| ';' expected
| to create file;
|      ^

jshell> /reset
| Resetting state.

jshell> int i=10
i ==> 10

jshell> /save
| '/save' requires a filename argument.

jshell> /save startup.jsh

jshell> saved as startup
| Error:
| ';' expected
| saved as startup;
|      ^

jshell> /vars
| int i = 10

jshell> /reset
| Resetting state.

jshell> /list

jshell> list command gives the entered data before we reset the terminal
| Error:
| ';' expected
```