

Special /Escape Characters and Sequences in JAVA

In this lecture, we will discuss about the special characters and sequences that are presented in JAVA. Also we will discuss about their use in the JAVA programs.

Some of the special characters used in the JAVA are as;

//	double slash	Marks the beginning of a single line comment.
()	open and close parenthesis	Used in a method header to mark the <i>parameter list</i> .
{ }	open and close curly braces	Encloses a group of statements, such as the contents of a class or a method.
" "	quotation marks	Encloses a string of characters, such as a message that is to be printed on the screen
;	semi-colon	Marks the end of a complete programming statement

If a character is preceded by a **backslash** (\) then it is known as **Java special sequences**. It may include **letters, numerals, punctuations**, etc.

Here, it has to be remembered that escape characters to be used must be enclosed in **quotation marks** (""). These are the valid character literals that signifies its usages. Then while running the application, Java compiler compiles these characters as a single character that adds a specific meaning to the compiler and the required output is shown.

In Java, there are following escape sequences that are described as;

Escape Sequences	Description
\t	It is used to insert a tab in the text at this point.
\'	It is used to insert a single quote character in the text at this point.
\"	It is used to insert a double quote character in the text at this point.
\r	It is used to insert a carriage return in the text at this point.

<code>\\</code>	It is used to insert a backslash character in the text at this point.
<code>\n</code>	It is used to insert a new line in the text at this point.
<code>\f</code>	It is used to insert a form feed in the text at this point.
<code>\b</code>	It is used to insert a backspace in the text at this point.

Now let us implement these with suitable examples as;

At first create a project in IntelliJ and create a new Java file or Class with name as **SpecialSequencesExample**. Then add following lines of code to it.

```
public class SpecialSequencesExample {
    public static void main(String args[])
    {
        //it inserts a Tab Space
        String str = "Astinvone\tdeStarx";
        System.out.println(str);
        //it inserts a New Line
        String str1 = "Morning\nshows the \nday";
        System.out.println(str1);
        //it insert a backslash
        String str2 = "And\\Or";
        System.out.println(str2);
        //it insert a Carriage
        String str3 = "Carriage\rReturn";
        System.out.println(str3);
        //it prints a single quote
        String str4 = "Wall Street's";
        System.out.println(str4);
        //it prints double quote
        String str5 = "Ujjwal'Astinvone de Starx'Humagain";
        System.out.println(str5);
        String str6 = "Ujjwal";
        System.out.println(str6);
    }
}
```

The output will be as;

```
Astinvone    deStarx
Morning
shows the
day
And\Or
Return
Wall Street's
Ujjwal'Astinvone de Starx'Humagain
'Ujjwal'
```

Comments in JAVA

In computer programming, comments are a portion of the program that are completely ignored by compilers or interpreters. Here in JAVA also we can use comments to do so. They are mainly used to help programmers to understand the code. For example,

```
// declare and initialize two variables
int a =1;
int b = 3;

// print the output
System.out.println("This is output");
```

Here, we have used the following comments,

- declare and initialize two variables
- print the output

Types of Comments in Java:

In Java, there are two types of comments:

- single-line comment

- multi-line comment

Single-line Comment:

A single-line comment starts and ends in the same line. To write a single-line comment, we can use the `//` symbol. For example,

```
// "Hello, World!" program example

class Main {
    public static void main(String[] args) {
        {
            // prints "Hello, World!"
            System.out.println("Hello, World!");
        }
    }
}
```

Here, we have used two single-line comments:

- "Hello, World!" program example
- prints "Hello World!"

The Java compiler ignores everything from `//` to the end of line. Hence, it is also known as End of Line comment.

Multi-Line Comment:

When we want to write comments in multiple lines, we can use the multi-line comment. To write multi-line comments, we can use the `/*...*/` symbol. For example,

```
/* This is an example of multi-line comment.
 * The program prints "Hello, World!" to the standard output.
 */

class HelloWorld {
    public static void main(String[] args) {
        {
            System.out.println("Hello, World!");
        }
    }
}
```

Here, we have used the multi-line comment:

```
/* This is an example of multi-line comment.  
 * The program prints "Hello, World!" to the standard output.  
 */
```

This type of comment is also known as Traditional Comment. In this type of comment, the Java compiler ignores everything from `/*` to `*/`.

Use Comments the Right Way:

One thing you should always consider that comments shouldn't be the substitute for a way to explain poorly written code in English. You should always write well-structured and self-explaining code. And, then use comments.

Some believe that code should be self-describing and comments should be rarely used. However, in my personal opinion, there is nothing wrong with using comments. We can use comments to explain complex algorithms, regex or scenarios where we have to choose one technique among different technique to solve problems.

Note: In most cases, always use comments to explain 'why' rather than 'how' and you are good to go.

Task to Do:

Write three paragraphs about something where you should include all the escape sequences and characters as well as the types of comments studied.