Question 4

**substr()**

Syntax

string.substr(start[, length]);

Argument

start − Location at which to start extracting characters (an integer between 0 and one less than the length of the string).

length − The number of characters to extract.

Note − If start is negative, substr uses it as a character index from the end of the string.

Return Value

The substr() method returns the new sub-string based on given parameters.

Example

Try the following example.

<html>

<head>

<title>JavaScript String substr() Method</title>

</head>

<body>

<script type="text/javascript">

var str = "Apples are round, and apples are juicy.";

document.write("(1,2): " + str.substr(1,2));

document.write("<br />(-2,2): " + str.substr(-2,2));

document.write("<br />(1): " + str.substr(1));

document.write("<br />(-20, 2): " + str.substr(-20,2));

document.write("<br />(20, 2): " + str.substr(20,2));

</script>

</body>

</html>

Output

(1,2): pp

(-2,2): Ap

(1): pples are round, and apples are juicy.

(-20, 2): Ap

(20, 2): d

**charAt()**

Syntax

string.charAt(index);

Argument Details

index − An integer between 0 and 1 less than the length of the string.

Return Value

Returns the character from the specified index.

Example

Try the following example.

<html>

<head>

<title>JavaScript String charAt() Method</title>

</head>

<body>

<script type="text/javascript">

var str = new String( "This is string" );

document.writeln("str.charAt(0) is:" + str.charAt(0));

document.writeln("<br />str.charAt(1) is:" + str.charAt(1));

document.writeln("<br />str.charAt(2) is:" + str.charAt(2));

document.writeln("<br />str.charAt(3) is:" + str.charAt(3));

document.writeln("<br />str.charAt(4) is:" + str.charAt(4));

document.writeln("<br />str.charAt(5) is:" + str.charAt(5));

</script>

</body>

</html>

Output

str.charAt(0) is:T

str.charAt(1) is:h

str.charAt(2) is:i

str.charAt(3) is:s

str.charAt(4) is:

str.charAt(5) is:i

**concat()**

Syntax

string.concat(string2, string3[, ..., stringN]);

Argument Details

string2...stringN − These are the strings to be concatenated.

Return Value

Returns a single concatenated string.

Example

Try the following example.

<html>

<head>

<title>JavaScript String concat() Method</title>

</head>

<body>

<script type="text/javascript">

var str1 = new String( "This is string one" );

var str2 = new String( "This is string two" );

var str3 = str1.concat( str2 );

document.write("Concatenated String :" + str3);

</script>

</body>

</html>

Output

Concatenated String :This is string oneThis is string two.

**indexOf()**

Syntax

string.indexOf(searchValue[, fromIndex])

Argument Details

searchValue − A string representing the value to search for.

fromIndex − The location within the calling string to start the search from. It can be any integer between 0 and the length of the string. The default value is 0.

Return Value

Returns the index of the found occurrence, otherwise -1 if not found.

Example

Try the following example.

<html>

<head>

<title>JavaScript String indexOf() Method</title>

</head>

<body>

<script type="text/javascript">

var str1 = new String( "This is string one" );

var index = str1.indexOf( "string" );

document.write("indexOf found String :" + index );

document.write("<br />");

var index = str1.indexOf( "one" );

document.write("indexOf found String :" + index );

</script>

</body>

</html>

Output

indexOf found String :8

indexOf found String :15

**match()**

Syntax

string.match( param )

Argument Details

param − A regular expression object.

Return Value

If the regular expression does not include the g flag, it returns the same result as regexp.exec(string).

If the regular expression includes the g flag, the method returns an Array containing all the matches.

Example

Try the following example.

<html>

<head>

<title>JavaScript String match() Method</title>

</head>

<body>

<script type="text/javascript">

var str = "For more information, see Chapter 3.4.5.1";

var re = /(chapter \d+(\.\d)\*)/i;

var found = str.match( re );

document.write(found );

</script>

</body>

</html>

Output

Chapter 3.4.5.1,Chapter 3.4.5.1,.1

**slice()**

Syntax

string.slice( beginslice [, endSlice] );

Argument Details

beginSlice − The zero-based index at which to begin extraction.

endSlice − The zero-based index at which to end extraction. If omitted, slice extracts to the end of the string.p>

Return Value

If successful, slice returns the index of the regular expression inside the string. Otherwise, it returns -1.

Example

Try the following example.

<html>

<head>

<title>JavaScript String slice() Method</title>

</head>

<body>

<script type="text/javascript">

var str = "Apples are round, and apples are juicy.";

var sliced = str.slice(3, -2);

document.write( sliced );

</script>

</body>

</html>

Output

les are round, and apples are juic