**Javascript Strings**

<!DOCTYPE HTML>

<html>

<head></head>

<!-- onblur -->

<body>

<h2>**JavaScript Data Types**</h2>

<p>**JavaScript has dynamic types. This means that the same variable can be used to hold different data types:</**p>

<p id="demo"></p><p id="demo1"></p><p id="demo2"></p><p id="demo3"></p><p id="demo4"></p><p id="demo5"></p><p id="demo6"></p><p id="demo7"></p><p id="demo8"></p><p id="demo9"></p><p id="demo10"></p><p id="demo11"></p><p id="demo12"></p><p id="demo13"></p><p id="demo14"></p><p id="demo15"></p>

<p id="demo16"></p><p id="demo17"></p><p id="demo18"></p><p id="demo19"></p>

<p id="demo20"></p><p id="demo21"></p><p id="demo22"></p><p id="demo23"></p>

<p id="demo24"></p><p id="demo25"></p><p id="demo26"></p><p id="demo27"></p>

<p id="demo28"></p><p id="demo29"></p><p id="demo30"></p><p id="demo31"></p>

<p id="demo32"></p><p id="demo33"></p><p id="demo34"></p><p id="demo35"></p>

<script>

var x; // Now x is undefined

x = 5; // Now x is a Number

y = "John"; // Now y is a String

var arr=[1]; // Now arr is a object

document.getElementById("demo").innerHTML = typeof(x);

document.getElementById("demo1").innerHTML = typeof(y);

document.getElementById("demo2").innerHTML = typeof(arr);

/\* ------- OUTPUT:-----------

**JavaScript Data Types**

**JavaScript has dynamic types. This means that the same variable can be used to hold different data types:**

**number**

**string**

**object**

\*/

/\***JavaScript Objects**

**Objects are variables too. But objects can contain many values.**

**The values are written as name:value pairs (name and value separated by a colon).**

**JavaScript objects are containers for named values called properties or methods.**

**syntax for method:**

**fullName: function() {return this.fname + " " + this.lname;}**

**for access:**

**objectName.methodName()**

\*/

var employee={fname:"abc",

lname:"xyz",

age:20,

isManager:true,

fullname: function()

{

return this.fname+" "+this.lname;

}

};

document.getElementById("demo3").innerHTML="the age of "+employee.fname+" " +employee["lname"]+ " is "+employee.age+". and is manager "+employee.isManager;

document.getElementById("demo4").innerHTML=employee.fullname();

/\* **------- OUTPUT:-----------**

**the age of abc xyz is 20. and is manager true**

**abc xyz**

\*/

var str="the string has \"double quotes\"";

var str1='another way to display \"double quotes\"';

var str2="now both \'single\' as well as \"double\" quotes with backslash\"\\\"";

document.getElementById("demo5").innerHTML=str;

document.getElementById("demo6").innerHTML=str1;

document.getElementById("demo7").innerHTML=str2;

/\***JavaScript String**

**the string has "double quotes"**

**another way to display "double quotes"**

**now both 'single' as well as "double" quotes with backslash"\"**

\*/

**//JavaScript String Methods**

/\*

**String Length**

**The length property returns the length of a string:**\*/

document.getElementById("demo8").innerHTML=str.length;//30

//**Finding a String in a String**

**//The indexOf() method returns the index of (the position of) the first occurrence of a specified text in a string:**

**//The lastIndexOf() method returns the index of the last occurrence of a specified text in a string:**

**//Both indexOf(), and lastIndexOf() return -1 if the text is not found.**

var str2="now both \'single\' as well as \"double\" quotes with backslash\"\\\"";

var pos=str2.indexOf("as");

var pos1=str2.lastIndexOf("as");

document.getElementById("demo9").innerHTML=pos; //18

document.getElementById("demo10").innerHTML=pos1; //56

//**Both methods accept a second parameter as the starting position for the search:**

var pos2=str2.indexOf("as",20);

document.getElementById("demo11").innerHTML=pos2; //26

//**The lastIndexOf() methods searches backwards, meaning: if the second parameter is 15,**

**//the search starts at position 15, counting from the end,**

**//and searches to the beginning of the string.**

var pos3=str2.lastIndexOf("as",20);

document.getElementById("demo12").innerHTML=pos3; //18

**//Searching for a String in a String**

**//The search() method searches a string for a specified value and returns the position of the match:**

var pos=str2.search("as");

document.getElementById("demo13").innerHTML=pos;//18

**/\*Did You Notice?**

**The two methods, indexOf() and search(), are equal?**

**They accept the same arguments (parameters), and return the same value?**

**The two methods are NOT equal. These are the differences:**

**The search() method cannot take a second start position argument.**

**The indexOf() method cannot take powerful search values (regular expressions).\*/**

**/\***

**Extracting String Parts**

**There are 3 methods for extracting a part of a string:**

**slice(start, end)**

**substring(start, end)**

**substr(start, length)**

**The slice() Method**

**slice() extracts a part of a string and returns the extracted part in a new string.**

**The method takes 2 parameters: the start position, and the end position (end not included).**

**This example slices out a portion of a string from position 7 to position 12 (13-1):\*/**

var str="the string has \"double quotes\"";

var substring1=str.slice(4,13);

document.getElementById("demo14").innerHTML=substring1 ;//string ha

**/\*If a parameter is negative, the position is counted from the end of the string.**

**This example slices out a portion of a string from position -12 to position -6:\*/**

var str="the string has \"double quotes\"";

var substring1=str.slice(-13,-4);

document.getElementById("demo15").innerHTML=substring1 ; //ouble quo

//**If you omit the second parameter, the method will slice out the rest of the string:**

var str="the string has \"double quotes\"";

var substring1=str.slice(-13);

document.getElementById("demo16").innerHTML=substring1 ;//ouble quotes"

var substring1=str.slice(13);

document.getElementById("demo17").innerHTML=substring1 ;//s "double quotes"

**/\*The substring() Method**

**substring() is similar to slice().**

**The difference is that substring() cannot accept negative indexes.\*/**

var str="the string has \"double quotes\"";

var substring1=str.substring(10);

document.getElementById("demo18").innerHTML=substring1 ; //has "double quotes"

/\***The substr() Method**

**substr() is similar to slice().**

**The difference is that the second parameter specifies the length of the extracted part.**

\*/

var str="the string has \"double quotes\"";

var substring1=str.substr(13,4);

document.getElementById("demo19").innerHTML=substring1 ; //s "d

//**If you omit the second parameter, substr() will slice out the rest of the string**

var substring1=str.substr(13);

document.getElementById("demo20").innerHTML=substring1 ; //s "double quotes"

**//If the first parameter is negative, the position counts from the end of the string.**

var substring1=str.substr(-3);

document.getElementById("demo21").innerHTML=substring1 ;//es"

/\***Replacing String Content**

**The replace() method replaces a specified value with another value in a string:**

**By default, the replace() method replaces only the first match:**

**By default, the replace() method is case sensitive. Writing DOUBLE (with upper-case) will not work:**

**\*/**

var str="the string has \"double quotes\"double";

var substring1=str.replace("double","DOUBLE");

document.getElementById("demo22").innerHTML=substring1 ; //the string has "DOUBLE quotes"double

**//NOTE:The replace() method does not change the string it is called on. It returns a new string.**

document.getElementById("demo23").innerHTML=str ; //the string has "double quotes"double

**//To replace case insensitive, use a regular expression with an /i flag (insensitive):**

**//Note: that regular expressions are written without quotes.**

var str="the string has \"double quotes\"double";

var substring1=str.replace(/Double/i,"DOUBLE");

document.getElementById("demo24").innerHTML=substring1 ; //the string has "DOUBLE quotes"double

**//To replace all matches, use a regular expression with a /g flag (global match):**

var str="the string has \"double quotes\"double";

var substring1=str.replace(/double/g,"aa");

document.getElementById("demo25").innerHTML=substring1 ; //the string has "aa quotes"aa

**/\*Converting to Upper and Lower Case**

**A string is converted to upper case with toUpperCase():**

**A string is converted to lower case with toLowerCase():\*/**

var substring1=str.toUpperCase();

document.getElementById("demo26").innerHTML=substring1 ; //THE STRING HAS "DOUBLE QUOTES"DOUBLE

var substring1=str.toLowerCase();

document.getElementById("demo27").innerHTML=substring1 ;//the string has "double quotes"double

**/\*The concat() Method**

**concat() joins two or more strings:**

**The concat() method can be used instead of the plus operator. These two lines do the same:**

**All string methods return a new string. They don't modify the original string.**

**Formally said: Strings are immutable: Strings cannot be changed, only replaced.**

**\*/**

var txt1="hello";

var txt2="world";

var txt3=txt1.concat(" ",txt2);

document.getElementById("demo28").innerHTML=txt3 ;// hello world

**/\*String.trim()**

**The trim() method removes whitespace from both sides of a string:\*/**

var txt3=" Hello World!! ";

document.getElementById("demo29").innerHTML=txt3.trim() ;//Hello World!!

**/\*Extracting String Characters**

**There are 3 methods for extracting string characters:**

**1. charAt(position)**

**The charAt() method returns the character at a specified index (position) in a string:**

**2. charCodeAt(position)**

**The charCodeAt() method returns the unicode of the character at a specified index in a string:**

**The method returns a UTF-16 code (an integer between 0 and 65535).**

**3. Property access [ ]**

**ECMAScript 5 (2009) allows property access [ ] on strings:**

**Note: It makes strings look like arrays (but they are not)**

**(If you want to work with a string as an array, you can convert it to an array.)**

**If no character is found, [ ] returns undefined, while charAt() returns an empty string.**

**It is read only. txt[0] = "A" gives no error (but does not work!)**

**\*/**

var txt="Javascript Strings"

document.getElementById("demo30").innerHTML=txt.charAt(0) ;//J

document.getElementById("demo31").innerHTML=txt.charCodeAt(0) ;//74

document.getElementById("demo32").innerHTML=txt[0] ;//J

**/\*Converting a String to an Array**

**A string can be converted to an array with the split() method:**

**eg. var txt = "a,b,c,d,e"; // String**

**txt.split(","); // Split on commas**

**txt.split(" "); // Split on spaces**

**txt.split("|"); // Split on pipe**

**If the separator is "", the returned array will be an array of single characters:**

**var txt = "Hello"; // String**

**txt.split(""); // Split in characters**

**\*/**

var str = "1a,|b,ca,d,e,f";

var arr = str.split("a");

for(var i=0;i<arr.length;i++)

document.write(arr[i]+"<br>");

/\*1

,|b,c

,d,e,f\*/

</script>

</body>

</html>

***Output:***

**JavaScript Data Types**

JavaScript has dynamic types. This means that the same variable can be used to hold different data types:

number

string

object

the age of abc xyz is 20. and is manager true

abc xyz

the string has "double quotes"

another way to display "double quotes"

now both 'single' as well as "double" quotes with backslash"\"

30

18

56

26

18

18

string ha

ouble quo

ouble quotes"

s "double quotes"

has "double quotes"

s "d

s "double quotes"

es"

the string has "DOUBLE quotes"double

the string has "double quotes"double

the string has "DOUBLE quotes"double

the string has "aa quotes"aa

THE STRING HAS "DOUBLE QUOTES"DOUBLE

the string has "double quotes"double

hello world

Hello World!!

J

74

J

1  
,|b,c  
,d,e,f