**JavaScript String function**

জাভাস্ক্রিপ্টে স্ট্রিং (String) হচ্ছে কিছু অক্ষরের সমষ্টি। যেমন "Hello Webcoachbd" বা 'How many tutorials you have' এগুলি হচ্ছে স্ট্রিং। ডাবল বা সিঙ্গেল কোটেশনের ভিতরে কিছু রাখলেই সেটা স্ট্রিং হিসেবে গন্য হয়। এমনকি '15' এটাও একটা স্ট্রিং। স্ট্রিং এ সিঙ্গেল না ডাবল কোটেশন দিবেন এটা নিজের ইচ্ছা। দুটি স্ট্রিং যেমন

var x = 'Hello tanvir;

var y = 'How are you?';

\*\* এভাবে তৈরী না করে স্ট্রিং অবজেক্ট ব্যবহার করেও তৈরী করা যায়, new কিওয়ার্ড দিয়ে যেমন

var x = new String('Hello Webcoachbd');

তবে এভাবে কাজ করা হয়না। সাধারনত  উপরের উদাহরনের মত করেই স্ট্রিং তৈরী করা হয়।

স্ট্রিং তৈরীর সময় যদি ভিতরে কোন কারনে পূনরায় কোটেশন দিতে হয় তাহলে সেটা এসকেপ (escape) করতে হবে যেমন

**var z = 'Webcoachbd\'s tutorials are awesome such as "OOP" tutorials';**

**// if use double quotation**

**var z = "Webcoachbd's tutorials are awesome such as \"OOP\" tutorials";**

\*\* যদি পুরো স্ট্রিং সিঙ্গেল কোটেশনের ভিতর রাখেন তাহলে ভিতরে ডাবল কোটেশন রাখতে পারেন, কাজ হবে তবে সিঙ্গেল কোটেশনের ভিতর সিঙ্গেল কোটেশন দিলে এসকেপ করতে হবে। তদ্রুপ ডাবল কোটেশনের ভিতর সিঙ্গেল কোটেশন দিলে এসকেপ করতে হবেনা কিন্তু ডাবল কোটেশন দিলে এসকেপ করতে হবে যেমন আমি শেষ লাইনে করেছি।

1. .length : যাইহোক স্ট্রিং নিয়ে কাজ করার জন্য জাভাস্ক্রিপ্টে প্রচুর ফাংশন এবং প্রোপার্টিজ আছে, যেমন .length দিয়ে স্ট্রিং এর দৈর্ঘ্য বের করা যায়।

<script>

var x = 'Hello Webcoachbd';

document.write(x.length);

</script>

1. [**charAt(index)**](http://www.webcoachbd.com/javascript-tutorials/javascript-functions)**:** স্ট্রিং থেকে নির্দিষ্ট একটা অক্ষর নেয়া যায়। index প্যারামিটারে যে সংখ্যা দিবেন সেই সংখ্যা যে অক্ষরের ইনডেক্স সেটা রিটার্ন করবে। 0 থেকে প্রথম অক্ষরের ইনডেক্স শুরু।

var x = 'Hello Webcoachbd';

document.write(x.charAt(8));

1. [**indexOf(string, start)**](http://www.webcoachbd.com/javascript-tutorials/javascript-indexof)**:** স্ট্রিং খোজার জন্য। string প্যারামিটারে যে শব্দ বা স্ট্রিংটির ইনডেক্স দিতে চান সেটা দিলে পুরো স্ট্রিংয়ে ঐ শব্দটির অবস্থান কত নম্বর ইনডেক্সে সেটা বের করবে আর start প্যারামিটারটি ঐচ্ছিক। যদি এটা দেন (পূর্নসংখ্যা) তাহলে ঐ সংখ্যা যার ইনডেক্স হবে সেখান থেকে খোজা শুরু করবে। যেমন ৫ দিলে ৫ নম্বর ইনডেক্সধারী এলিমেন্ট থেকে খোজা শুরু হবে।

var str = "Hello world, welcome to the universe.";  
var n = str.indexOf("welcome");

//outpur 13

1. [**lastIndexOf(string, start)**](http://www.webcoachbd.com/javascript-tutorials/javascript-indexof)**:** পিছন দিক থেকে স্ট্রিং খোজার জন্য। ধরুন একটা স্ট্রিংয়ে ১০ টি "e" আছে এখন আপনি চান সর্বশেষ "e" টা কত নম্বর ইনডেক্সে আছে সেটা দেখবেন তখন এই মেথড ব্যবহার করতে পারেন। string প্যারামিটারে যেটা খুজবেন সেটা এবং start (এটা না দিলে ০ ইনডেক্সধারী এলিমেন্ট থেকে খোজা শুরু করবে) প্যারামিটারে কত নম্বর ইনডেক্স থেকে খোজা শুরু করবেন সেটা।

var country = Bangladesh is small country';

var pos = country.lastIndexOf('country'');

document.write('The position of "country" in country = ' + pos);

1. [**substr(start, length)**](http://www.webcoachbd.com/javascript-tutorials/javascript-functions)**:** স্ট্রিং থেকে একটা অংশ/সাবস্ট্রিং বের করার জন্য। start হল কত নম্বর স্ট্রিং পজিশন থেকে খোজা শুরু করবে আর length হচ্ছে কয়টি ক্যারেক্টার/অক্ষর নিবে।

var x = 'Hello Webcoachbd';

document.write(x.substr(3,5));

1. **parseInt(string):** স্ট্রিং থেকে পূর্নসংখ্যা বানাতে ব্যবহৃত হয়।

==> স্ট্রিং এর শুরুতে যে সংখ্যা থাকবে সেটাই রিটার্ন করবে।

==> যদি দশমিক নাম্বার দেন  যেমন 22.36 তাহলে পূর্নসংখ্যা বানিয়ে রিটার্ন করবে।

==> স্ট্রিংয়ের শুরুতে যদি কোন পূর্নসংখ্যা না থাকে তাহলে NaN(Not a Number) রিটার্ন করবে।

document.write(parseInt('52 Hello Webcoachbd'));

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

document.write(parseInt('52.230'));

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

document.write(parseInt('Hello'));

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

document.write(parseInt('52'));

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

document.write(parseInt('-52'));

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

document.write(parseInt('0Hello'));

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

document.write(parseInt('0xHello'));

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

document.write(parseInt('0x10'));

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

document.write(parseInt('10110001110',2));

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

document.write(parseInt('22',8));

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

document.write(parseInt('22',16));

1. **parseFloat(string):**স্ট্রিং থেকে নাম্বার করার জন্য।এখানে string এর জায়গায় যে স্ট্রিং দিবেন সেটার প্রথমে যদি কোন নাম্বার থাকে তাহলে সেটাকে নাম্বার করে রিটার্ন করবে। এখানে কোন radix প্যারামিটারে নেই। বাকি সব কাজ parseInt() এর মতই শুধু পার্থক্য হচ্ছে floating point number (দশমিক নাম্বার) দিলে parseFloat() সেটাকে সেভাবেই রিটার্ন করবে যেখানে parseInt() শুধু পূর্নসংখ্যার অংশটুকু রিটার্ন করবে। যেমন

document.write(parseFloat('52'));

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

document.write(parseFloat('52Hello'));

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

document.write(parseFloat('52.36'));

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

document.write(parseFloat('52.00'));

1. **floor():**এই মেথড দিয়ে দশমিক সংখ্যাকে কাছাকাছি কোন পূর্নসংখ্যাতে নিয়ে যেতে ব্যবহৃত হয় (নিচের দিকে যাবে)। যেমন ১.৮০ কে ১.০ বানিয়ে দিবে। এটা Math অবজেক্টের একটা মেথড তাই একসেস নিতে Math এরপর ডট (.) দিয়ে ব্যবহার করতে হবে। যেমন

document.write(Math.floor(1.80));

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

document.write(Math.floor(1.20));

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

document.write(Math.floor(-1.80));

1. **ceil() :**এটাও floor() এর মত শুধু পার্থক্য হচ্ছে ceil() উপরের দিকে যাবে। যেমন ১.৮০ কে সে ২ বানাবে যেখানে floor() মেথড ১ বানাবে।

document.write(Math.ceil(1.80));

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

document.write(Math.ceil(1.20));

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

document.write(Math.ceil(-1.80));

1. search() মেথড একটি প্যারামিটার সমর্থন করে। রেগুলার এক্সপ্রেশন। যেকোন স্ট্রিংয়ে রেগুলার এক্সপ্রেশন প্যারামিটার ব্যবহার search(regex) মেথড দিয়ে সব সাবস্ট্রিং/শব্দগুলি কোন পজিশনে আছে সেটা দেখতে পারেন।

search(regex) মেথড স্ট্রিংয়ের প্রথম থেকে খোজা শুরু করবে কোথায় মিল পাওয়া যায় (রেগুলার এক্সপ্রেশনের সাথে), যখনি প্রথম মিল পেয়ে যাবে তখন সেই সাবস্ট্রিং/শব্দের পজিশন রিটার্ন করে যদি মিল না পায় তাহলে -1 রিটার্ন করবে। যেমন

var regExp = /Webcoachbd/;

var sen = "What do you know about Webcoachbd. Webcoachbd is the largest Bengali tutorial site.";

var isMatch = sen.search(regExp);

if(isMatch != -1){

document.write("There was a match at position " + isMatch);

}else{

document.write("There was no match in the first string");

}

searh() এর মত আরকেটি মেথড আছে যার নাম **match()** এটা দিয়ে মিলে যাওয়া স্ট্রিং/শব্দগুলি বের করে আনা যায়। যেমন

var regExp = /Webcoachbd/g;

var sen = "What do you know about Webcoachbd. Webcoachbd is the largest Bengali tutorial site.";

var isMatch = sen.match(regExp);

document.write("Strings those matches are : " + isMatch);

// and the array becomes isMatch = ['Webcoachbd', 'Webcoachbd'];

1. জাভাস্ক্রিপ্টে setTimeout() নামে একটা ফাংশন আছে যেটা দিয়ে একটা ফাংশন তথা কোডব্লক নির্দিষ্ট সময় পর এক্সিকিউট করাতে পারেন স্বয়ংক্রিয়ভাবে। এটা setInterval() ফাংশনের মতই একটা ফাংশন। শুধু ছোট্র কয়েকটি পাথর্ক্য আছে। গুরত্বপূর্ন পার্থক্যটি হল**setTimeout()**ফাংশনে যে সময় দেয়া হবে সেটা পার হবার পর একবার কোডব্লক এক্সিকিউট হয় অপরদিকে**setInterval()**ফাংশনে যে সময় দেয়া থাকে সেই সময় পর পর কোডব্লক অটোমেটিক অনির্দিষ্টকালের জন্য এক্সিকিউট হতে থাকবে, কখনও থামবেনা।

setTimeout() এর ভিতরে প্যারামিটার হিসেবে প্রথমে ফাংশনটির নাম বা পুরো কোডটুকুই লিখে দিতে পারেন এবং দ্বিতীয় প্যারামিটারে সময় দিতে হয়। মিলিসেকেন্ডে সময় দিতে হয়। যেমন ১ সেকেন্ড পর যদি কোডটি চালাতে চান তাহলে দিবে ১০০০।

<button onclick="myFunction()">Try it</button>

<script>

function myFunction() {

setTimeout(function(){ alert("Hello"); }, 3000);

}

</script>

**Or**

<button onclick="openWin()">Open "myWindow"</button>

<script>

function openWin() {

var myWindow = window.open("", "myWindow", "width=200, height=100");

myWindow.document.write("<p>This is 'myWindow'</p>");

setTimeout(function(){ myWindow.close() }, 3000);

}

</script>

**Create clock:**

<!DOCTYPE html>

<html>

<body onload="startTime()">

<div id="txt"></div>

<script>

function startTime() {

var today = new Date();

var h = today.getHours();

var m = today.getMinutes();

var s = today.getSeconds();

// add a zero in front of numbers<10

m = checkTime(m);

s = checkTime(s);

document.getElementById("txt").innerHTML = h + ":" + m + ":" + s;

var t = setTimeout(function(){ startTime() }, 500);

}

function checkTime(i) {

if (i < 10) {

i = "0" + i;

}

return i;

}

</script>

</body>

</html>